JAPAN EXCHANGE GROUP

JPX WORKING PAPER

Hypothetical Market Data Scenario Generation Using Generative AI

Syed Hashim SHAH[§,†]
Joel VIKLUND[§,†]
Minoru AOKI[§,‡]

[§]Japan Securities Clearing Corporation
[†]Aurora Solutions K.K.
[‡]Nomura Research Institute, Ltd.

August 4, 2025
Vol.48

Hypothetical Market Data Scenario Generation Using Generative AI

Syed Hashim SHAH[§,†] Joel VIKLUND[§,†] Minoru AOKI[§,‡]

[§]Japan Securities Clearing Corporation
[†]Aurora Solutions K.K. [‡]Nomura Research Institute, Ltd.

August 4, 2025

Abstract

Generative AI and AI have revolutionized a wide range of industries in recent years. However major use cases have been limited to Large Language Models (LLMs) such as chatbots, code assistants and text analysis. In this white paper we explore the use of generative AI for risk management in the context of central counterparties (CCPs). Since the world of finance is ruled by exact numbers, we want to look at generative models which can handle and generate numerical data. In the paper we will first introduce common AI and machine learning concepts and then explore how to use generative AI models to create synthetic yet realistic market data with a focus on 3-Month TONA Futures contracts. We compare variational autoencoder (VAE) and principal component analysis (PCA) to generate synthetic data and analyze the generated scenarios. Subsequently, we use the generated synthetic market data to estimate the risk profile of various portfolios by calculating Expected Shortfall Value-at-Risk (ES-VaR). The results showed that the VAE generated scenarios were more diverse and affected the ES-VaR more than the PCA generated scenarios.

Disclaimer

1. JSCC does not intend to use generative AI to create scenarios for initial margin calculation. This document is only for research purposes.
2. The English version is the authoritative version.

Table of Contents

1 Introduction

JSCC is the CCP (central counterparty) in Japan Exchange Group (JPX), and it clears several products such as cash equity, exchange traded derivatives, and OTC products such as IRS/CDS and JGB.

1.1 Role of CCP and Risk Management

A CCP is one of FMIs (Financial Market Infrastructures). FMIs are critical to the security and stability of the financial markets. A complex web of bilateral exposures is reduced to a simpler network via the CCP. A CCP can provide multilateral netting benefits [1]. Multilateral netting reduces settlement risk as shown in Fig. 1.1.

(a) Bilateral network. **(b)** Multilateral netting.

Fig. 1.1: Comparison of complex bilateral clearing network and a simplified multilateral clearing in presence of CCP.

A CCP also acts as the guarantor to every trade with each respective member or counterparty (the buyers and sellers). This role deters the default chain reaction. Multilateral netting structure also helps the CCP to do settlements on behalf of defaulters. CCPs mitigate financial risks.

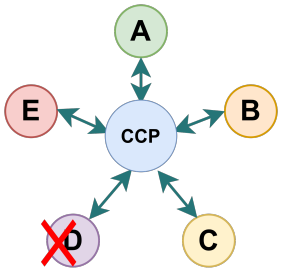


Fig. 1.2: CCP will pay and receive on behalf of defaulters based on assumption of obligation.

1.2 CCPs Absorb Risk

If any of the clearing members were to fail to meet their trade obligations to the CCP, then the default management process (DMP) would be triggered which includes the process by which financial resources are utilized from the defaulted member, the CCP and non-defaulting clearing members' resources where applicable.

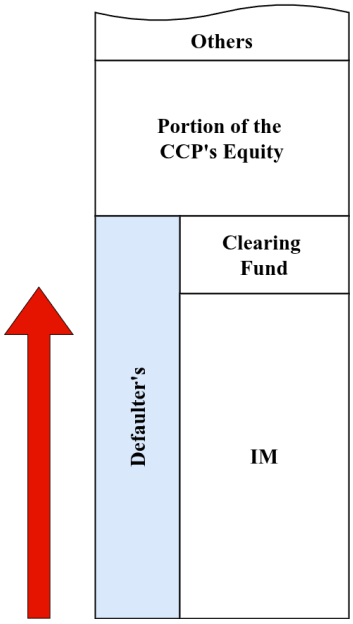


Fig. 1.3: Risk waterfall in DMP.

The main part of financial resources for a defaulted member is the initial margin (IM). Too low IM will cause lack of financial resources in DMP, and too large IM is also bad for a member's funding efficiency. So, balance is important. The more risk a portfolio has, the more IM is required.

1.3 Value at Risk

IM is an expected loss amount on the default event. Many CCPs have introduced historical simulation value at risk (HS-VaR) in their IM calculation methodology. Its loss distribution does not assume any specific mathematical distribution but an empirical one in many cases. IM is an expected loss amount by percentile on the distribution. Loss amount on percentile is called Value at Risk. In order to capture the tail loss, sometimes the average loss amount over a percentile is used as VaR, and this is called conditional VaR or expected shortfall (ES). In this paper, VaR means expected shortfall. A CCP needs to complete the DMP within a specific period, defined by the product. This is called Margin Period Of Risk or MPOR.

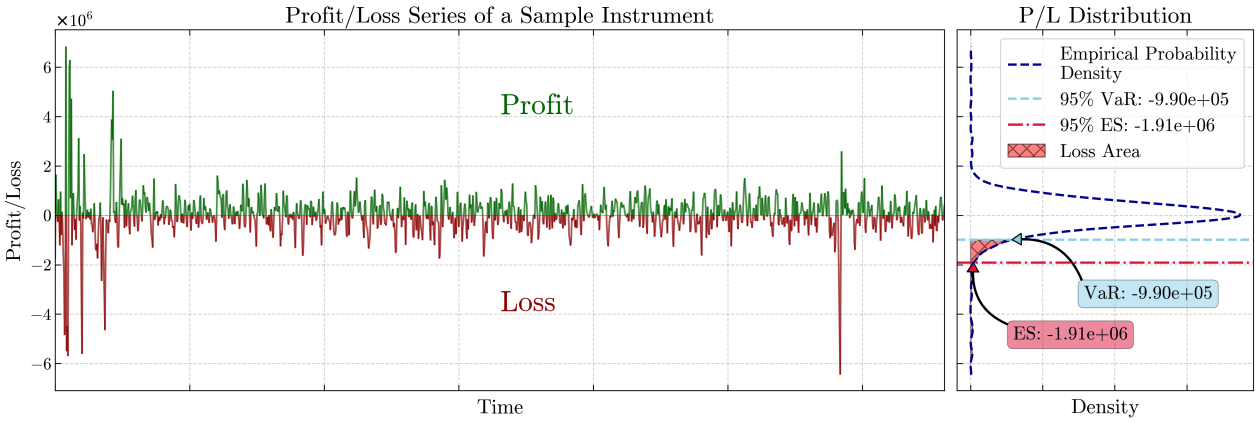


Fig. 1.4: Profit/Loss distribution over Margin period of Risk.

Each of the risk factor movements, called returns, are not independent and are sometimes correlated. Such correlated historical returns are used to determine the IM amount based on their probability. By applying each return to the current prices we can create instrument PLs for VaR calculation purposes. For futures, we

can create historical returns for any time to expiry by modeling the historical prices as future price curves that reflect the price level by time to expiry. The historical occurrence of these scenarios is a compelling argument for their plausibility as possible future scenarios. These possible scenarios are used in IM calculation and are called historical scenarios. Furthermore, historical scenarios which occurred during periods of stress are called stress scenarios.

In contrast, hypothetical scenarios are those which have not occurred in the past but are created artificially under certain assumptions. They should be extreme but plausible to cover for previously unseen stress events. The IM is calculated based on all the three scenario types. However, imagining scenarios that have not happened yet is difficult and requires a considerable amount of expertise. With AI, it may be possible to generate new probable scenarios which we will refer to as plausible synthetic scenarios. A key factor for sophisticated risk management is the quality of scenarios employed in the IM calculation.

1.4 Generative AI for Market Scenarios

Generative AI can learn from historical big data. It can capture the linear and non-linear correlations between the risk factors and can produce a compressed representation of data points with reduced dimensionality. This representation is called latent space. By taking a sample of new data points in this latent space and restoring it back to the original data space, we can create synthetic data samples. By using generative AI, it may be possible to create plausible synthetic scenarios based on the probability of occurrence of other risk factor movements.

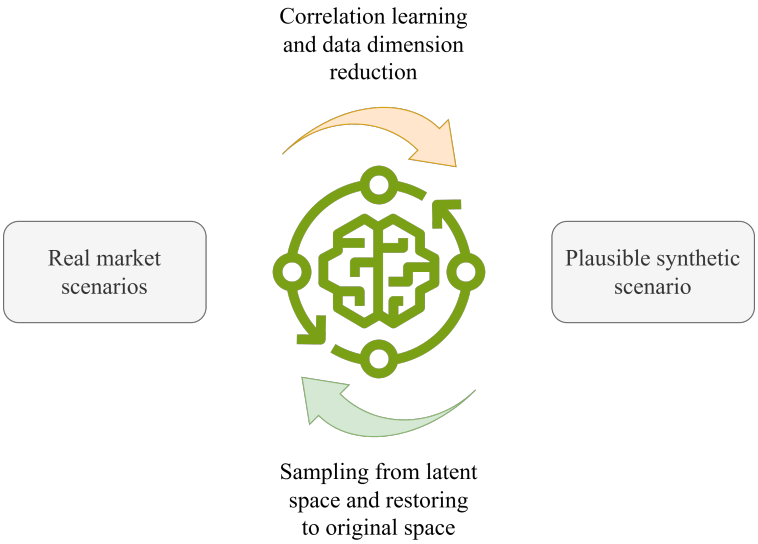


Fig. 1.5: Generative AI use case for market scenarios.

This is the initial motivation to try to use generative AI in this white paper. In the following chapters, we will describe the general aspects of AI/ML/generative AI and then use some the introduced methodologies to create synthetic financial data for 3-Month TONA Futures. 3-Month TONA Futures are interest rate futures based on the 3-month compounded Tokyo Over-Night Average (TONA) rate and were listed in 2023 by the Osaka Exchange of the JPX Group [2]. The MPOR of this product is 2 days. In the final chapters of the white paper, we will integrate the synthetically generated scenarios in a few sample VaR calculations to explore how they can expand our risk coverage.

2 Life, Machine Learning, AI and Generative AI

Artificial intelligence, machine learning and generative AI have been applied to a wide variety of both consumer and corporate domains and have revolutionized how we approach and solve complex problems. Before we jump into discussing artificial intelligence, we need to at least have a working definition of intelligence. Defining intelligence has been difficult but if we take a minimal working definition of defining intelligence as "ability to solve problems", then we can see that history of intelligence aligns well with evolution of biological life and subsequent rise of humanity. We humans have been able to solve extremely complex problems, comprehend abstract concepts and languages, create art and marvelous feats of engineering thanks to our extremely complex brains.

2.1 Definitions

Artificial Intelligence is a broad umbrella term that includes many methods, heuristics, and algorithms. It has a wide range of subfields and methods. A brief visual summary of the field is shown in Fig. 2.1.

Artificial Intelligence (AI):

AI can be defined as a simulation of biological intelligence, especially human-like intelligence, by machines and particularly by computer systems.

- Examples include areas and methodologies such as ant colony optimization and genetic algorithms.

Machine Learning (ML):

Simulation of intelligent behavior without explicitly programming such behaviors into the computers. Computer programs automatically learn relationships directly from the data.

- Examples include algorithms such as decision trees, random forests and deep neural networks.

Generative AI (GenAI):

Using AI/ML to generate synthetic yet realistic data.

- Examples include text engines such as ChatGPT and others, realistic image models like Stable Diffusion and Dall-E [3], [4], [5].

The data on computers, such as text, pictures, and audio, must be represented as numbers as computers only understand binary 0s and 1s. Financial data is also represented as numbers and hence it is a prime target for machine learning and generative AI.

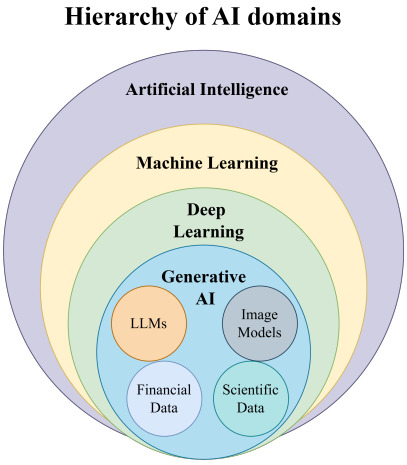


Fig. 2.1: Landscape of Artificial Intelligence Domains.

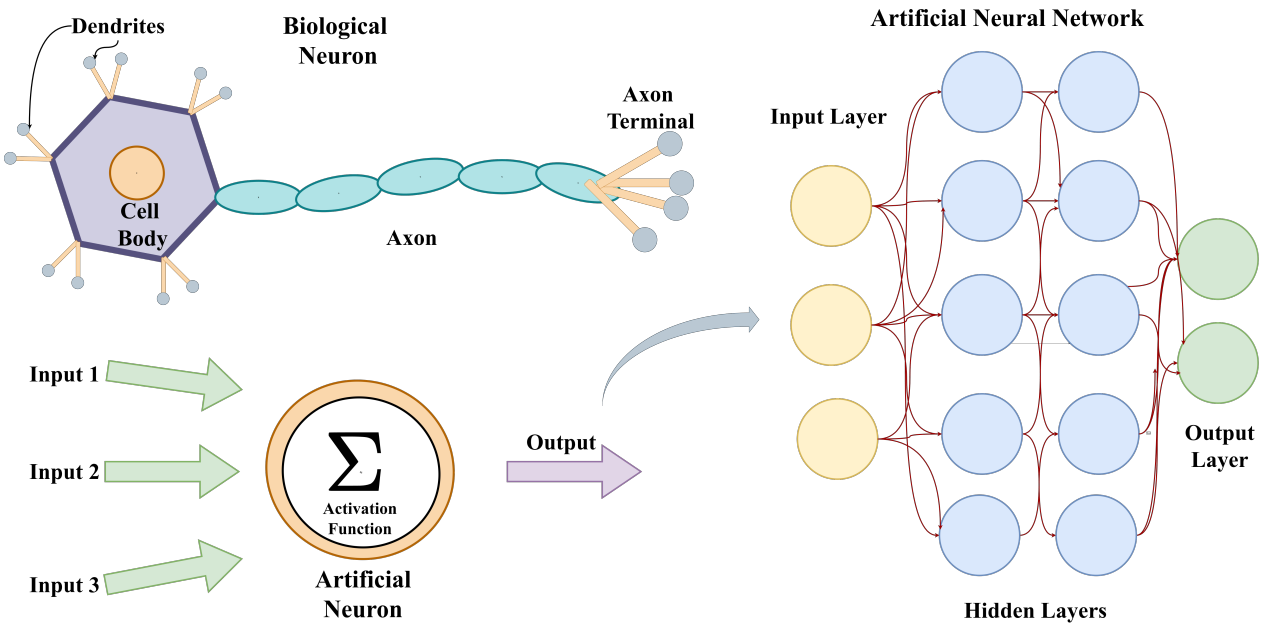


Fig. 2.2: A schematic of biological neuron. Dendrites receive signals from other neurons around them which are then processed in cell body and an electric signal is sent down through axons. The Axon terminal connects with other neurons downstream. A mathematical approximation of the biological neuron called "Artificial Neuron" is also shown. It receives multiple numerical inputs and combines into a single numerical output value by activation function and passes it to other neurons downstream. When multiple such neurons are connected in a network, we get an "Artificial Neural Network", which can fit arbitrarily complex patterns. We can add multiple hidden layers in between input and output layers and obtain "Deep Neural Networks".

We can program a computer with complex rules to simulate intelligent behavior, but those algorithms cannot learn or adapt. This is how the initial chess and board game type algorithms worked which defeated world champions. Powerful AI algorithms called Artificial Neural Networks (ANNs) were inspired by neurons in human brains and could learn automatically from data and do complex tasks [6],[7]. Neural networks are mathematical analogs of biological neuronal networks in our brains, as shown in Fig. 2.2. While our brains can have billions of neurons interconnected in a massively complicated network flowing with electrochemical signals, the mathematical neural networks which can be coded into computer software, take numerical inputs and do complex mathematical transformations to output calculation result.

The reason behind the success of ANNs is a mathematical theorem named "Universal Approximation Theorem" [8]. Briefly, universal approximation theorem states that a sufficiently wide neural network can approximate any continuous function to arbitrary accuracy. Intuitively, since inside the neural network there are so many degrees of freedom (parameters) which can be independently tweaked, we can approximate any continuous function or pattern with it.

2.2 AI/Machine Learning is Curve Fitting

To many risk professionals, the idea of curve fitting or fitting a probability distribution are familiar tasks. Traditionally, splines are used for fitting curves and copulas are used for fitting data distributions. We will soon see that the core of machine learning is curve/surface fitting or capturing underlying data distributions by using novel architectures not traditionally seen in long-standing classical statistical methods. When we have complicated high-dimensional data sets, we need a generic architecture to learn decision boundaries and probability distributions. In classical statistics, we make certain assumptions on the data and try to fit it via gaussian, a mixture of gaussians or some combination of fat tailed distributions. This approach is limited because of:

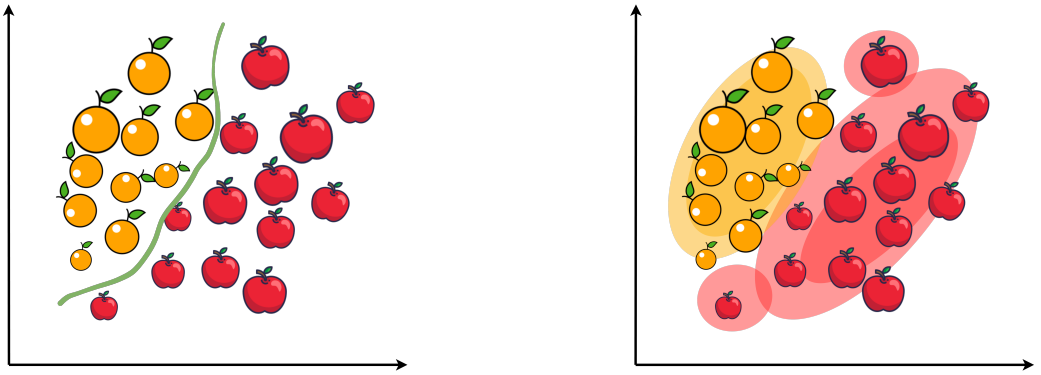
1. Imposition of assumptions on data (independence of features, ignoring non-linear correlations).

2. The curse of dimensionality. In high dimensions, the possible number of combinations of data features grows exponentially, which classical models fail to model. Our intuition can easily fail us in higher dimensions.

Therefore, using neural networks allows us to fit complicated models to our data without making any simplifying assumptions.

There are two paradigms that are used in machine learning to build AI models, depending on the task:

1. Discriminative.
2. Generative.



(a) Discriminative model learns decision boundary.

(b) Generative model learns feature distribution.

Fig. 2.3: Comparison of two ML paradigms: Discriminative (Fig. 2.3a) where we learn the decision boundary vs. Generative Modelling (Fig. 2.3b) where we learn the probability density.

Discriminative AI/ML is about classifying or predicting a label of the data given a data sample. When classifying data into different classes, we want to find a discriminating boundary in feature space which efficiently segregates the data classes as shown in Fig. 2.3a. This boundary can be a complex curve or surface in case of high-dimensional data. Therefore, simple linear boundaries or simple functional forms are usually not enough. A flexible functional form is typically needed to form the boundary. Discriminative models are trained in a supervised manner.

In generative modeling of the data, we try to learn the joint probability distribution of the data features and their labels, as illustrated in Fig. 2.3b. We learn different combinations of data features and their ranges, and learn a probability distribution over them. For example, we can learn the size, color and shape parameter combinations that define apples and oranges. The generative aspect is then just sampling efficiently from this latent distribution and converting it into data space. Since different features can be distributed in a complex, usually non-gaussian manner and can have non-linear dependencies on each other, a flexible non-parametric function is needed to model them.

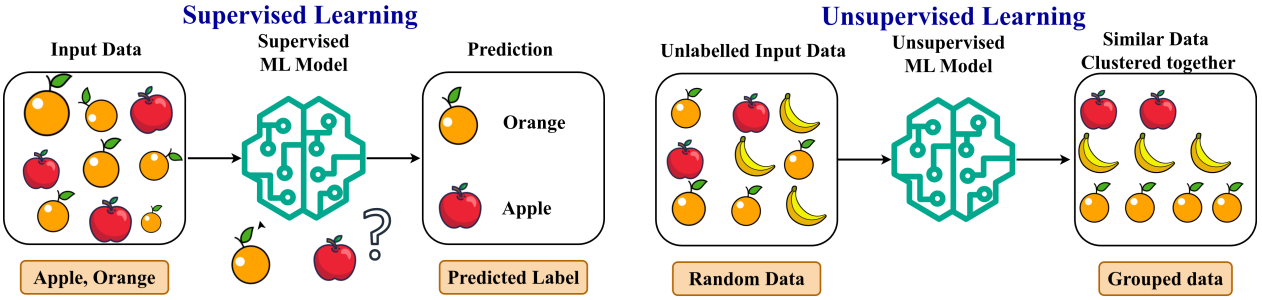
2.3 Supervised, Unsupervised, and Reinforcement Learning

There are three major ways we can train or teach the AI and ML models to do the tasks we want:

1. Supervised Learning.
2. Unsupervised Learning.
3. Reinforcement Learning.

Supervised learning is when, along with the questions or data, we also provide the answer or label for that data. For example, if we show enough pictures of oranges and apples to an AI model and tell them that these are apples and these are oranges, it can learn to distinguish between them as illustrated in Fig. 2.4a.

Unsupervised learning is when we do not have data with labels. We train the algorithm to learn the complex patterns in the data. It can learn to find similarities between different data samples and cluster them together, or make predictions, as shown in Fig. 2.4b.



(a) Distinguishing apples and oranges.

(b) Grouping similar fruits together.

Fig. 2.4: A schematic comparison of supervised learning (a) and unsupervised learning (b) methodologies.

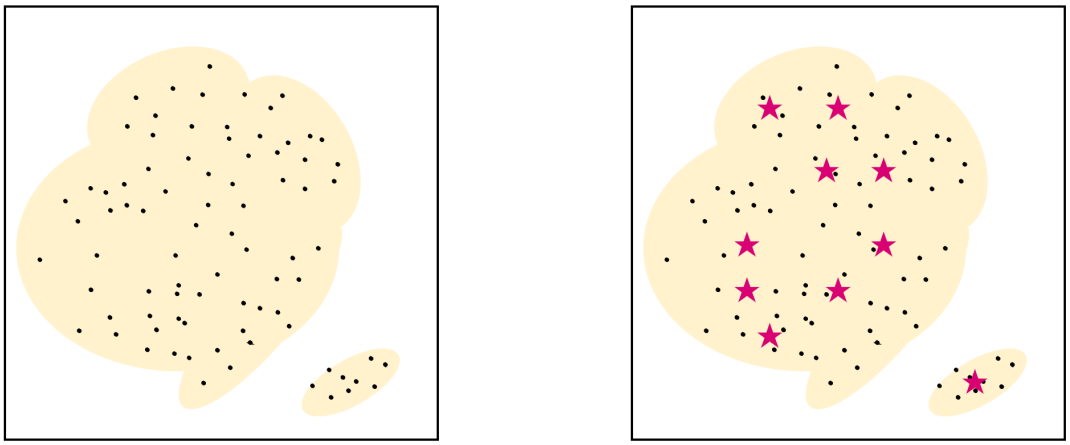
We will not discuss reinforcement learning in detail, but briefly it is when an ML model or an intelligent agent learns to take actions or decisions in a dynamic environment to maximize the reward. It is rewarded and penalized for right and wrong decisions. Reinforcement learning has been successfully applied in fields such as dynamic asset allocation, trading, portfolio management and dynamic hedging.

In practice, when building AI models, all three approaches are combined in the following order:

1. Unsupervised learning, since most of the available data is unlabeled.
2. Supervised learning, since accurately labeled data is limited. We can refine our unsupervised models with this.
3. Reinforcement learning, to fine tune the models and teach them extremely complex tasks.

2.4 Generative AI

If we have large data sets of images, multiple financial time series, or a corpus of text, we can learn the complex relationships between various aspects and features of the data. This can be done by using methods from the field of generative AI/modelling. The essence of generative AI is simple if we have some basic statistics and probability theory knowledge. Essentially, it is to learn a joint data feature distribution as accurately as possible and then sample from that distribution properly to generate novel synthetic data which follows the original data distribution as shown in Fig. 2.5.



(a) Data and its source probability distribution (yellow).

(b) Generating new data samples (★) from probability distribution.

Fig. 2.5: The black dots represent the data samples, while the yellow cloud approximates the probability distribution out of which data was generated. The stars (★) show new sampled data points that were sampled from this probability distribution. These new data points are from the same probability distribution, but they do not exist in our original data set.

In low-dimensional cases such as 2 or 3 dimensions, and when data joint distribution is simple such as a gaussian or mixture of gaussians, modelling and sampling from it is intuitive and easy. However, real data distributions are complex, high-dimensional and sparse. Sparse here means that in high-dimensional feature space, most of real-world data samples are concentrated in a small region of space. The available space of feature value combinations increases exponentially with each extra feature dimension, but we might not have enough data to fill a significant portion of that space. With any realistic data, we quickly end up with the curse of dimensionality. If we just randomly sample in this high-dimensional space, we will not be able to create any realistic data samples.

Secondly, in a complex real data distribution, we might have varying correlations in different regions of the data distribution. Some regions may have low probabilities or data may be sparse. Therefore, even in 2 dimensions we might not have a probability density function (PDF) and an associated sampling methodology.

We have two problems at hand:

1. Learn a continuous smooth probability distribution of the data.
2. Sample from that distribution efficiently.

Almost all generative AI research and methods are about solving these two problems.

2.4.1 Learning probability distributions and sampling

Real data distributions are complex, and in practice, we do not even know the ground truth distribution that generates our data. If we could transform our complex and sparse data samples into a continuous and simpler distribution, then it would be easier to handle and sample from. We can easily sample it and transform that sample back into real data by doing an inverse transformation. This process is illustrated in Fig. 2.6. We have an encoder function which encodes our data samples to regions in a simpler latent distribution, for example, a gaussian. By sampling randomly from this gaussian distribution and passing it through the decoder function, we generate a realistic looking data point. In general, both encoder and decoder functions are neural networks. For instance, we could sample from a multivariate normal distribution and then transform that sample into an actual data point, such as a realistic looking image or financial data.

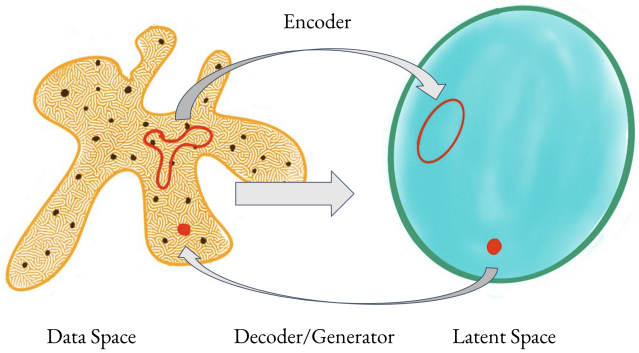


Fig. 2.6: Transforming complex real data distributions to a simpler easily workable distributions and back to data space. Encoder neural network transforms data samples into simpler latent distribution while decoder transforms samples of latent distribution back to the data distribution. The strangely shaped red region in data space is transformed into a simpler region in latent space.

How do we convert our complex high-dimensional data distribution into a simpler one? In this paper, we cover two major techniques: principal component analysis (PCA) which is a classical technique, and Variational autoencoders (VAE) which is one of the core generative AI methods.

2.4.2 Principal Component Analysis

Principal Component Analysis (PCA) of high-dimensional data finds a sequence of directions in data space called principal components (PCs) along which the data exhibits maximum variance or change [9]. The PC directions are ordered by the amount of variance they explain. Furthermore, they are orthogonal, linearly uncorrelated, and define a new coordinate system. The PCs are linear combinations of data features which have maximum covariance, and they are obtained as eigen vectors of the data covariance matrix. If we project our data onto the plane defined by the first two PCs which explain most of the data variance, we obtain a 2-dimensional representation of our high-dimensional data.

Mathematically, PCA is just a linear transformation, i.e., it is conducted by simple matrix multiplication of our data matrix with a transformation matrix. Intuitively, it gives us a viewpoint of the data that captures the most significant linear structure, as illustrated in Fig. 2.7.

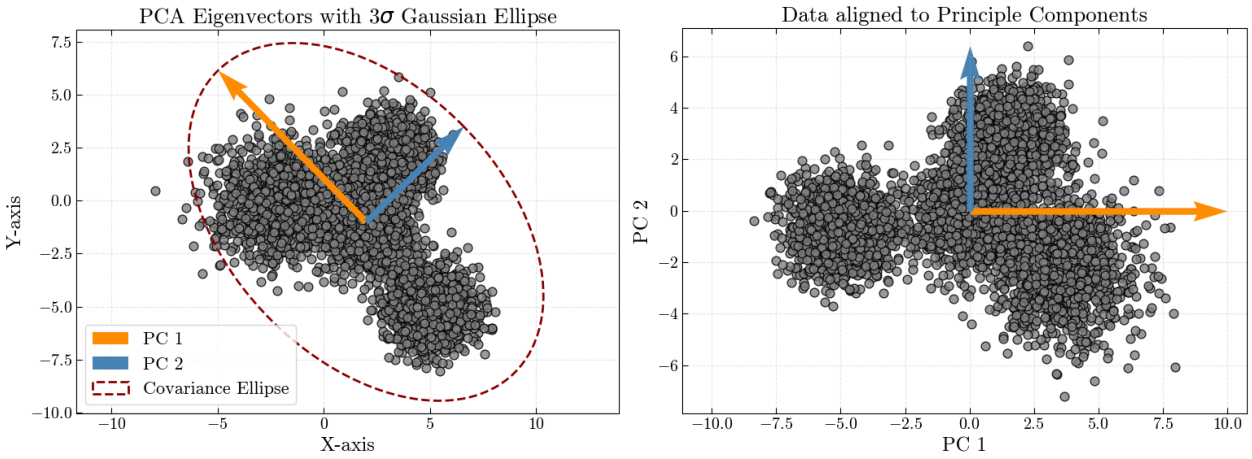


Fig. 2.7: Principal component analysis of dummy data generated from a mixture of gaussians. We can see that the PC1 eigen vector points in the direction of the largest linear change in data, while PC2 eigen vector points in the direction of the second-largest linear movement. We can also see the data in the new coordinates.

PCA just looks at linear structure of the data and completely ignores the non-linear relationships in the data.

Completely different data sets can have exactly the same PCs. For example, if we fit a gaussian distribution to the data in Fig. 2.7 and generate samples from that, it will have the exactly same eigen vectors but will be a completely different data distribution.

2.4.3 Variational Autoencoders (VAE)

Variational autoencoders are a type of generative model architecture that utilizes neural networks and ideas from probabilistic graphical models [10]. VAEs are a popular model for image generation, data compression, and denoising.

The step-by-step mechanism of VAE is as follows:

1. We pass our data into a deep neural network called an encoder with multiple hidden layers, as shown in Fig. 2.8.
2. The number of neurons in each layer gets smaller and smaller until we reach a bottleneck layer, which represents the latent space. This bottleneck forces the NN to learn the most essential features of the data, and the non-linear encoder NN captures the non-linear relationships between the data. This latent space Z is an efficient, compressed representation of our data.
3. The latent representation of our data is slightly perturbed with gaussian noise, and then passed to the decoder neural network.
4. The decoder tries to reconstruct the original input data as accurately as possible. In the beginning, the reconstructions will be poor, and the reconstruction errors will be extremely high. As we loop through these steps and train both the neural networks with backpropagation, the internal parameters of both the encoder and the decoder get updated until we get close to the original data-reconstructions. As a result, the latent space of VAE will also be a transformed representation of this lower data-carrying manifold.

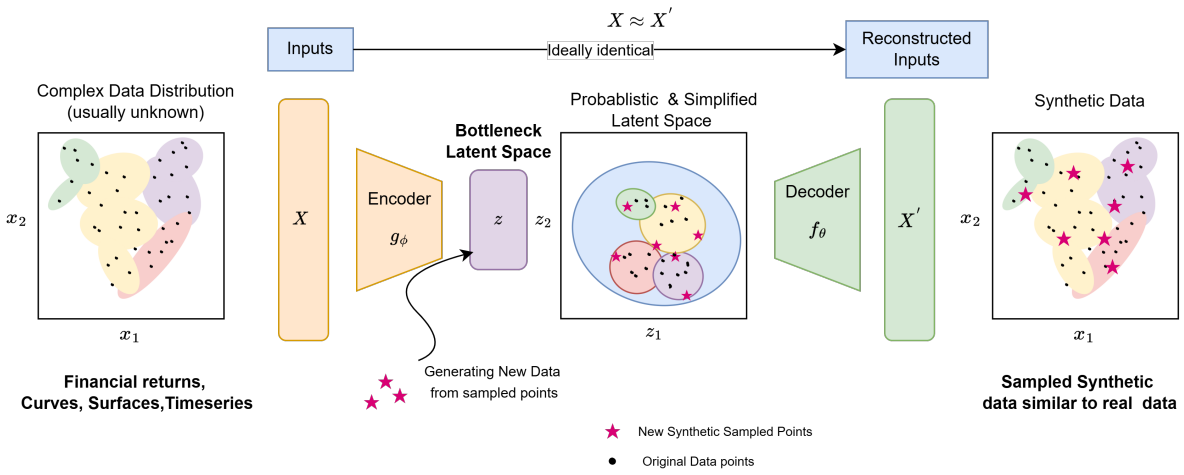


Fig. 2.8: Architecture of VAE.

VAE learns a continuous lower-dimensional latent space representation of data. The latent distribution is usually of lower dimensionality than data and directions in this latent space are meaningful. This reduction in dimension reduces redundancies, reduces computational complexity for doing operations on the probability space and is typically much easier to sample from. In the case of image data, we can reduce millions of dimensions (each colored pixel is three dimensions) into a few hundred dimensions. Sampling from this simpler distribution and decoding back to data space allows us to create synthetic yet realistic images.

Briefly, the variational autoencoders try to solve the following issues:

1. Efficiently compressing the high-dimensional sparse data into a structured, lower-dimensional simple probabilistic and continuous latent space.
2. Enable easy sampling of the latent space.

By using the architecture of variational autoencoders (Fig. 2.8), we can learn a compressed latent distribution of our data which has features of smoothness, structure, and continuity. For example, we might have several hundred dimensional data, yet most of that data lies in a simpler lower-dimensional subspace of a few dozen internal dimensions embedded in that space. Imagine a curved piece of paper embedded in 3-dimensional space. The curved paper has inherent dimensionality of two (approximately), yet its curvature forces it to be in 3 dimensions. By using a variational autoencoder we can learn this lower-dimensional data carrying subspace and effectively generate new synthetic data by sampling points from it.

Furthermore, the latent space of VAE is additionally regularized to make it smooth and continuous by using Kullback–Leibler divergence. This allows for any linear interpolation in the latent space of VAE to have meaningful reconstructions in data space, as illustrated in Fig. 2.9.

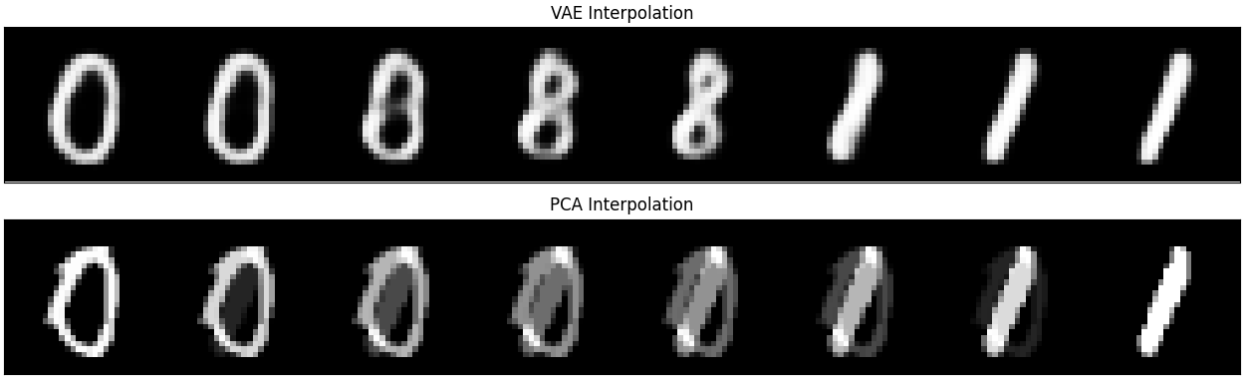


Fig. 2.9: Interpolating between an image of handwritten 0 and 1 in VAE vs. PCA latent space. All intermediate reconstructions of VAE interpolation look realistic and handwritten, while PCA space interpolation leads to apparitions where one digit slowly disappears and the other slowly appears.

VAEs are extremely extensible and powerful frameworks and have been used in multiple applications with remarkable success. We can compare PCA with VAE by compressing our data into 2 dimensions by both methods and reconstruction based on that. We will see that VAE preserves significantly more information in its latent space due to learning non-linear mappings. However, perfect reconstructions are not the goal of VAE as that can force the VAE to memorize data and decrease its ability as generative model. If we have a 3-dimensional point cloud which looks like a car, the first 2 PCs will give us a side view of the car while ignoring the depth information. VAEs on the other hand extend this idea of reducing dimensionality by using non-linear transformations. Instead of multiplying our data matrix with a transformation matrix as in PCA, we pass our data into a neural network which compresses into a lower-dimensional space non-linearly. For example, the 3-dimensional model of a car will be converted into a representation where the whole surface of the car has been peeled and laid flat on a 2-dimensional surface.

When building a generative AI model, particularly in a financial context, it's desirable to have a model with a probabilistic and explicit latent space, such as VAE, to facilitate control over the generative process.

2.5 Other Generative Models

VAEs are not the only technique for learning data distribution effectively and creating synthetic data. There are multiple other techniques which have emerged over the last decade which allow us to do generative modelling effectively, each with their own strength and weaknesses:

1. Generative Adversarial networks (GANs).
2. Flow based models.

3. Diffusion models.

Generative adversarial networks [11] are trained via an adversarial optimization technique where two neural networks play a min-max game against each other. A generator network tries to convert random gaussian noise vectors to a realistic looking data sample, for instance, a picture of a human face while a discriminator network tries to discriminate against a fake generated picture or data from a real data point. The goal of the generator is to produce such samples that the discriminator cannot tell them apart from real data. The goal of the discriminator is to be able to tell real and fake data apart. Both are punished if they fail to meet their goal. Equilibrium reaches when the generator produces samples which the discriminator can assign only 50% probability of being real or fake i.e., random guessing. At this stage, the generator has learned the true data distribution. Once trained, GANs take a standard multivariate gaussian noise vector and produce a realistic sample from it. Hence, it is a map from multivariate gaussian distribution to data distribution. The multivariate gaussian is usually of lower dimension than actual data dimensions. However, the disadvantage of GANs is that our latent space is just standard normal gaussian, and we have no control over this latent space. We can generate data, but we cannot embed our existing data into a latent space or manipulate it.

Similarly, flow based models [12] learn invertible mappings that transform data distribution step by step into a gaussian distribution of the same dimensionality. By reversing this transformation, we can convert any gaussian sample into realistic data. This allows for exact likelihood calculation. However, the need for invertible mappings restricts the allowed neural network architectures. Furthermore, the high dimensionality of the latent space makes training difficult and slow.

Diffusion models [13], [14], [15] take a very physics-based approach. They slowly convert the data samples into gaussian samples by slowly adding gaussian noise to them step by step. Then a neural network is trained to predict the denoised version of the sample. They are excellent at generating highly realistic data and are at the heart of models such as Stable Diffusion, Midjourney and Dall-E [4], [5], [16]. Theoretically they do not need to have a latent space as they can generate realistic looking images just from multidimensional gaussian noise. However, they are slow to sample from.

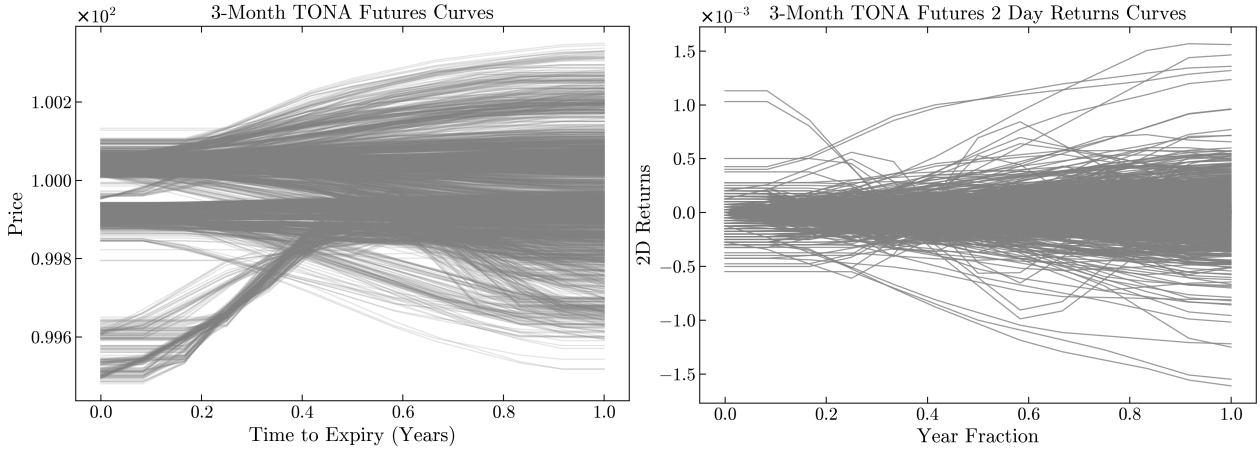
The architecture of variational autoencoders with a structured and continuous latent space is extremely useful from the perspective of extension. VAEs have been extended with adversarial loss similar to GANs, which gives us an explicit latent space along high generation quality [17]. VAEs also have been combined with diffusion models, which allows for massive speed-ups as well [18]. Similarly, ideas from flow-based models have been combined with the VAE latent space as well [19].

3 Generative AI For Synthetic Financial Data

In this section, we elaborate on the methodology and results for generating synthetic 3-Month TONA futures curves by using both VAE and PCA.

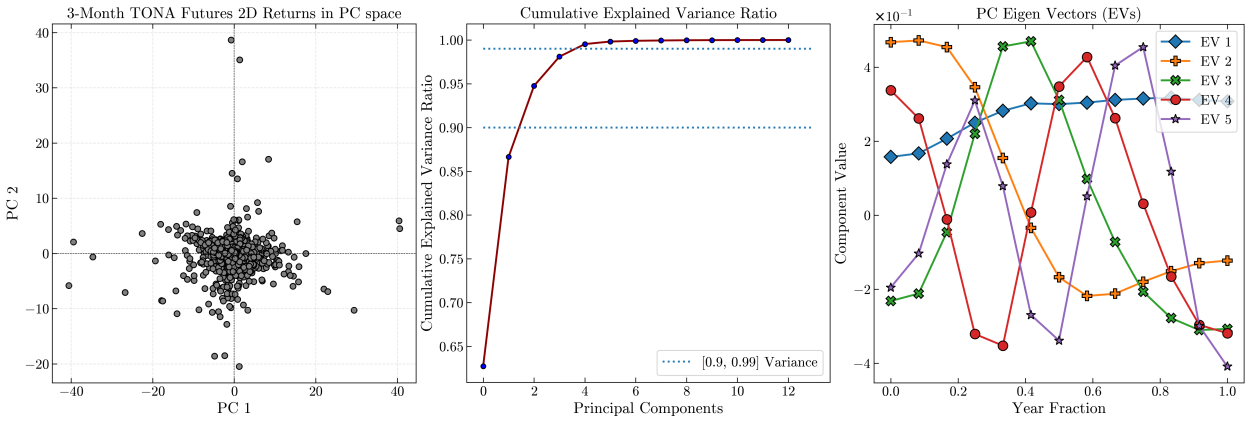
3.1 3-Month TONA Futures

To capture the market movements aligned with the MPOR introduced in Section 1.4, 2-day log returns (2D) were calculated using the raw 3-Month TONA futures curves as shown in Fig. 3.1. For the purposes of this paper, we consider each returns curve to be an independent sample, thus precluding the consideration of data autocorrelation. The resulting returns curves for maturities up to one year are presented below:



(a) Futures Curves. **(b)** 2D Log Returns.

Fig. 3.1: Historical 3-Month TONA futures curves are shown in Fig. 3.1a. Their corresponding 2D log returns curves, which capture the historical futures curve movements over the 2-day MPOR, are shown in Fig. 3.1b. These historical returns scenarios will be used as training data for our generative models.



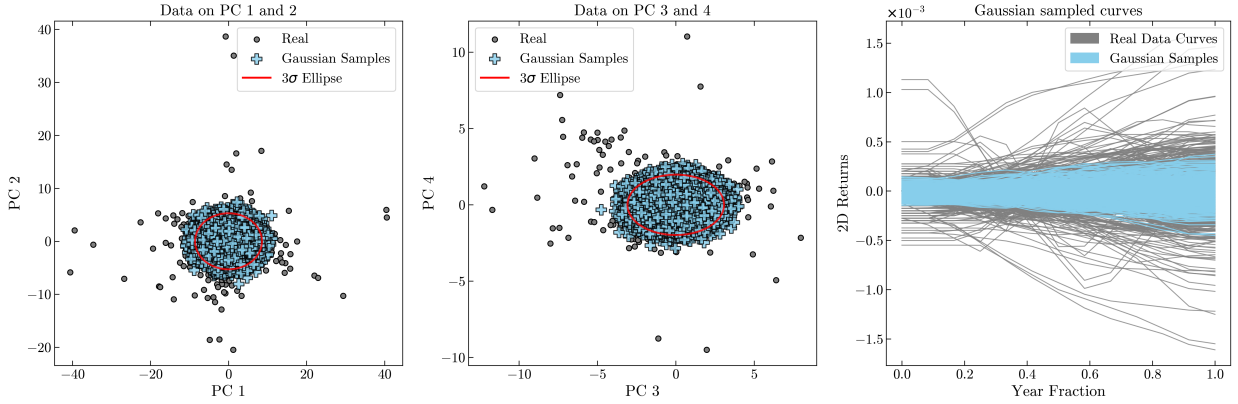
(a) Projection on PC1&2. **(b)** Explained Variance. **(c)** Eigen Vectors.

Fig. 3.2: Principal component projection of the returns data (Fig. 3.2a). Variance explained by eigen vectors (Fig. 3.2b), and the first few eigen vectors (Fig. 3.2c).

3.2 Using PCA as Generative Model

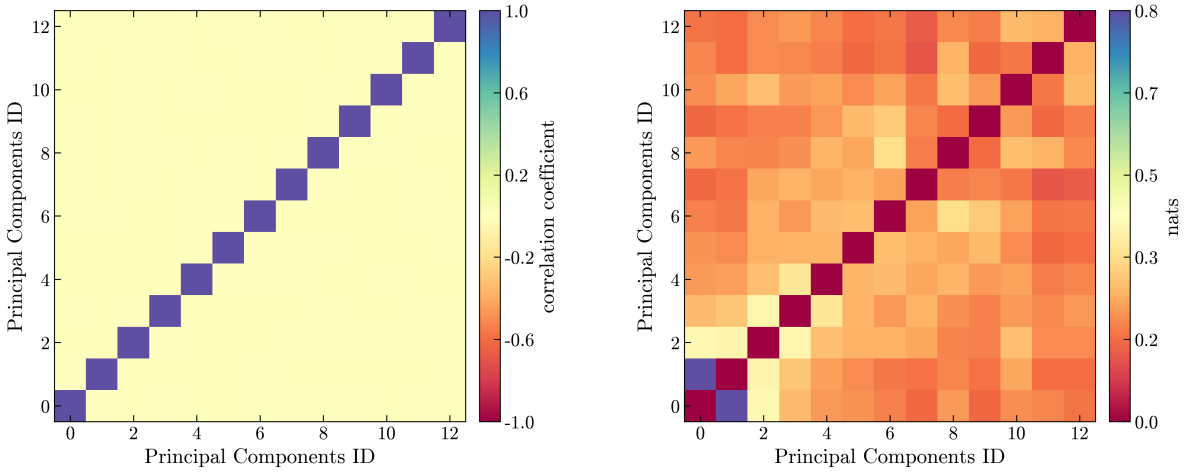
The simplest and naïve way to generate synthetic data is to first transform the data into principal component space (Fig. 3.2). Subsequently, we can fit a multivariate gaussian to this data in PC space, sample that multivariate gaussian distribution, and inverse transform the synthetic PCs back into data space as illustrated

in Fig. 3.3. We see that this sampling strategy is not able to take into account the tails of the distribution, as shown in Fig. 3.3c. Furthermore, even though the principal components are uncorrelated by construction, they might not necessarily be independent, as shown in Fig. 3.4.



(a) Gaussian on PC1&2. **(b)** Gaussian on PC3&4. **(c)** Gaussian samples as curves.

Fig. 3.3: Sampling in PCA latent space via jointly fitting a gaussian on all the data in PC space (Fig. 3.3a, Fig. 3.3b). The gaussian fails to capture the fat tails. The sampled points are converted back to curve space (Fig. 3.3c), and we can clearly see that gaussian samples only capture the core of the data distribution.



(a) Correlation matrix of PCs. **(b)** Mutual information matrix of PCs.

Fig. 3.4: Correlation between principal components is zero by design. Even after taking away the linear dependencies through PCA transformation, we observe that the mutual information between principal components is non-zero, which indicates the existence of non-linear dependencies in PCs and consequently in real data features as well.

The average mutual information in off-diagonal terms is 0.234 nats (Fig. 3.4b) and is equivalent to a correlation coefficient of 0.61 if we assume the two variables are gaussian. The formula connecting mutual information and the correlation coefficient of two gaussian random variables is:

$$|r| = \sqrt{1 - e^{-2I}} \tag{1}$$

Where r is the correlation coefficient, and I is mutual information. Even after linearly decorrelating the data via PCA transform (Fig. 3.4a), there still exists a non-linear dependency in the data with strength equivalent to a linear correlation of 0.61. This serves as an interpretive aid because mutual information is

positive and has no upper bound, while correlation coefficient is bounded. A perfect correlation will have unbounded mutual information. Since PCA cannot capture non-linear relationships in the data, we need non-linear methods to accurately capture the co-movements of various tenors and capture the fat tail effects.

3.3 Using VAE as Generative Model

To model the non-linear dependencies in the returns curves, we train a VAE to learn their joint distribution. Once we have the trained VAE model, we can sample and explore its latent space to generate new curves. The latent space of VAE, due to strong regularization, is much more well-behaved as shown in Fig. 3.5.

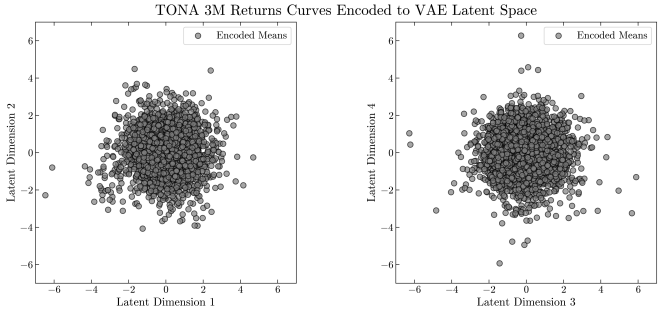


Fig. 3.5: First four latent space dimensions of VAE.

VAE latent space is smooth, i.e., nearby points in the latent space are transformed into similar data points by the decoder. Its smoothness and continuity allow us to sample data between any two actual curves. We can see that we can interpolate between two different curves both in data space and in VAE latent space. The data space interpolation trajectory is linear, as shown by the blue line in the principal component projection in the middle plots in Fig. 3.6. However, linear interpolation in VAE space, shown by the orange line, is curved when projected into PC space and follows contours of the data distribution. Therefore, the intermediate interpolated points are also much more realistic than those generated by simple interpolation.

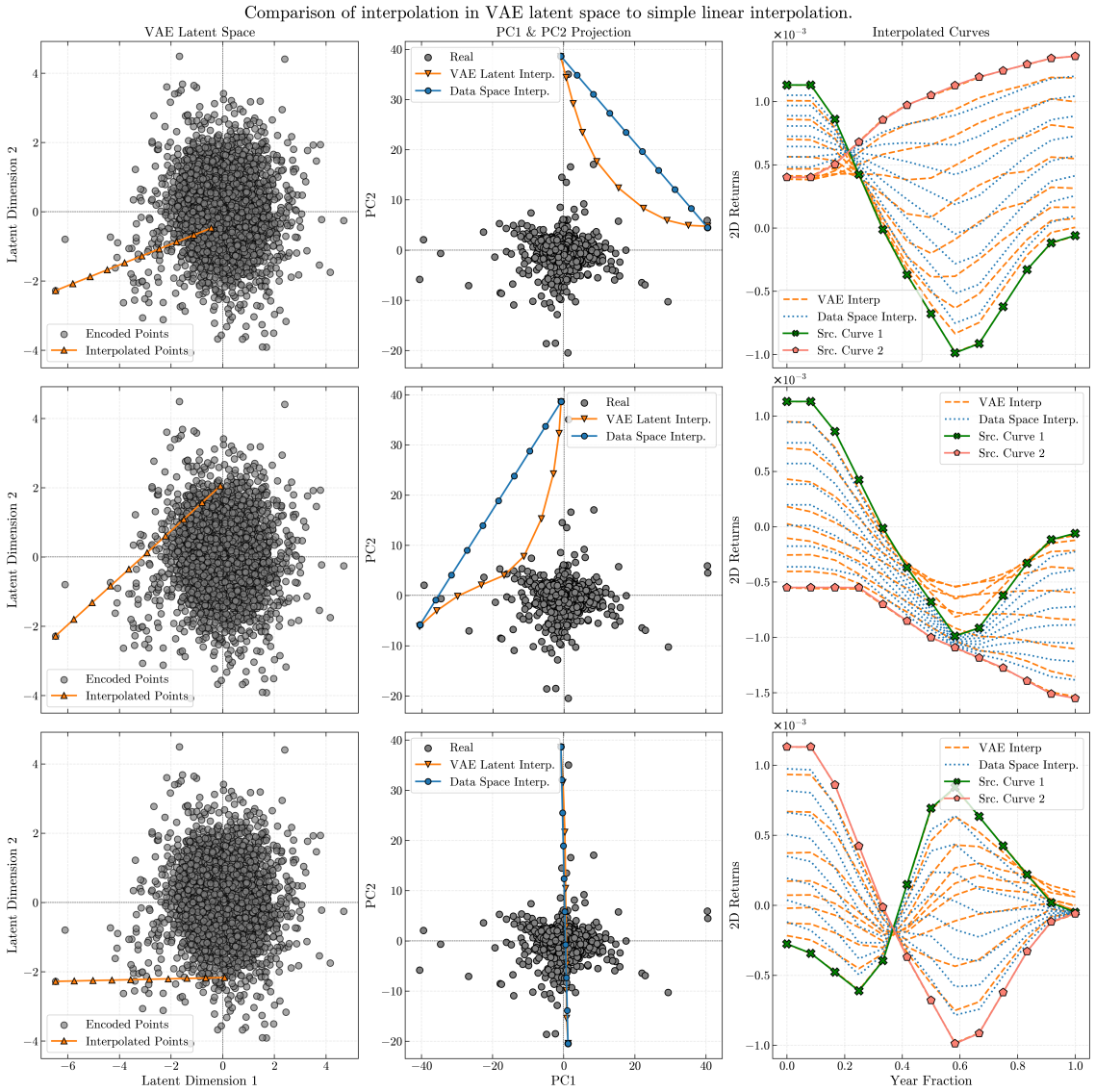


Fig. 3.6: Interpolation in VAE latent space vs data space. VAE produces more realistic curves and follows the contours of the data distribution.

3.4 Sampling the Data Distribution using Hierarchical VAE

To sample from the VAE's latent space, we can adopt multiple strategies. First, we discuss how to sample VAE's latent space in general and some fundamental issues and their resolutions. As a first step, we can fit a gaussian distribution to the latent space of the VAE and then randomly sample gaussian vectors and decode them into data space.

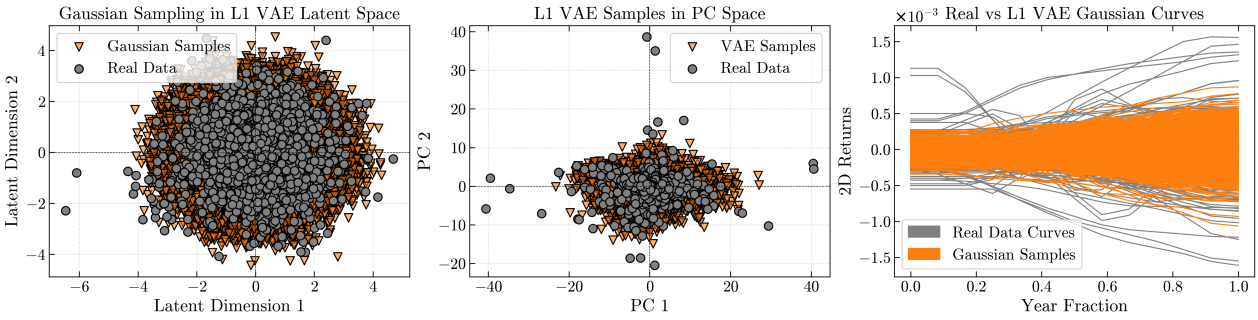


Fig. 3.7: The trained VAE latent space is non-gaussian. It is much more well-behaved than the data space but still non-gaussian.

However, as illustrated in the Fig. 3.7, despite the inherent VAE based regularization towards a standard normal, the aggregated latent space does not need to be close to a gaussian. Therefore, fitting a gaussian distribution to the latent space leads to ignoring the tails and poor sampling in extremes. As shown in the Mahalanobis distance Q-Q plot in Fig. 3.8, the latent space is still highly non-gaussian. Mahalanobis distance for n-dimensional normally distributed data should follow a Chi-Square distribution with n degrees of freedom. We can work around this issue by fitting another VAE on the latent space of the first VAE. We treat the latent space of the first VAE as data and input it into another VAE with a latent space of the same dimensionality as the original latent space. The second level VAE's latent space is better-behaved but still not gaussian enough to enable easy sampling. We fit a third VAE onto that, which leads to a very gaussian latent space, as shown in Fig. 3.8. Then we sample in level 3 VAE's latent space by fitting a gaussian distribution from which we obtain level 2 samples, which are then converted to level 1 samples, as shown in Fig. 3.9.

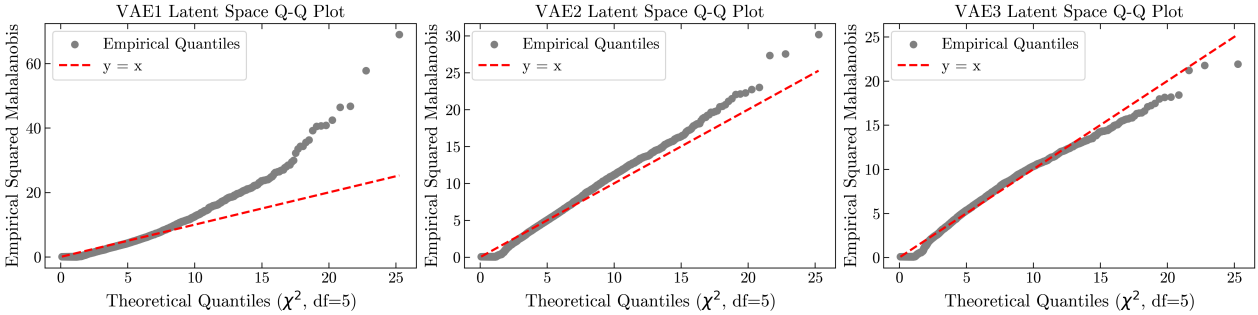


Fig. 3.8: As shown by the Q-Q plot of Mahalanobis distance level 2 VAE is still non-gaussian, while level 3 VAE is very close to gaussian, hence gaussian sampling can work.

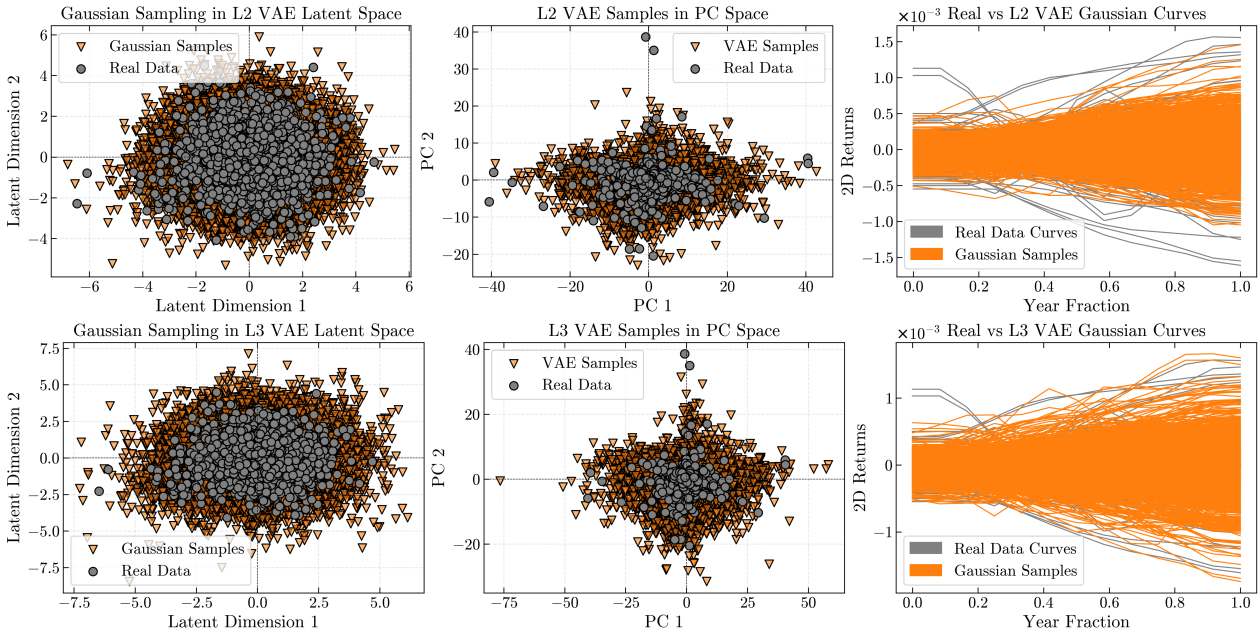


Fig. 3.9: Top panel shows samples in level 2 VAE while the bottom panel shows level 3 VAE latent space and corresponding samples in PC space. Hierarchical VAEs convert the latent space to match increasingly towards a gaussian, which can be sampled effectively by fitting a gaussian distribution. The decoded curves are shown on the right.

We can see that, at the level 3 latent space, the sampling encompasses the entire distribution nicely, including the outliers, as shown in the Fig. 3.9 middle column and the rightmost column.

3.5 Sampling around Individual Curves

We have two representations of 3-Month TONA futures returns curves. One is in principal component space; the other is VAE latent space. If we can find the extreme points and perturb them via gaussian noise in either principal component space or VAE latent space, we can generate different versions of those historical scenarios. In this case, we use level 1 VAE only. In the PCA latent space, we can select a scenario and sample around it with the same covariance matrix as the principal components.

As shown in Fig. 3.10, when sampling in the PC space around a particular scenario according to a gaussian with the same covariance as principal components, the first two eigen vectors which describe the mean value and slope have the strongest effect. This results in the sampled scenarios being largely similar to the original scenarios, but with parallel shifts and minor adjustments to their slopes. However, while sampling in VAE latent space, we sample according to the latent space covariance matrix. This allows us to sample in a nonlinear fashion and sample a wider range of curves around a given curve. Instead of using the global covariance matrix, VAE uses the local covariance for sampling data, giving more weight to different data features where needed rather than placing all the weight on the PC1&2, which only capture the global structure. We can see in the third column of Fig. 3.10 that when a data point is described mostly by the PC1 or PC2, its gaussian perturbations in the VAE latent space lead to larger perturbations along PC1 or PC2 respectively. While in PC space, even the samples described mostly by PC2, the PC1 can dominate under the gaussian perturbations with the PC covariance matrix.

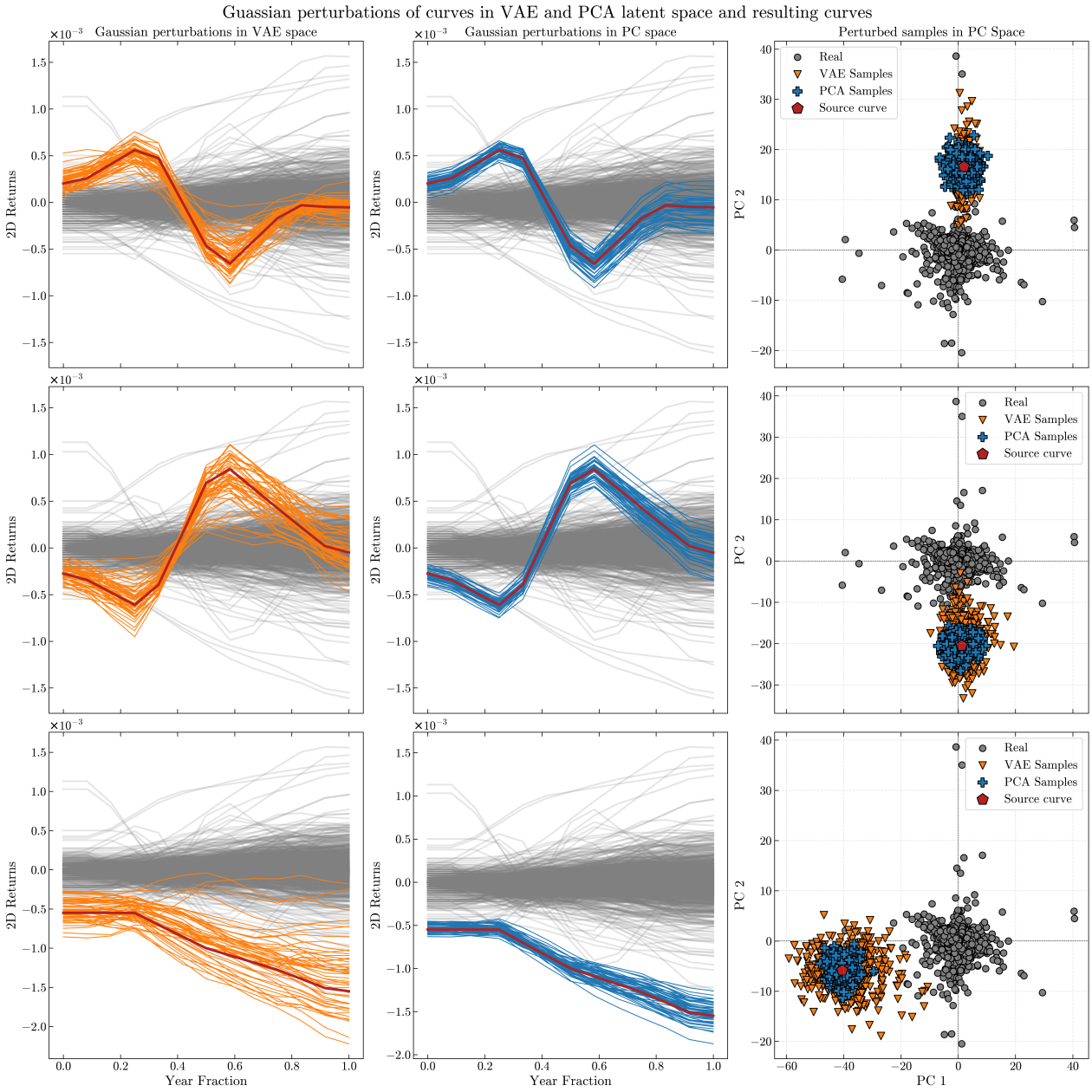


Fig. 3.10: Examples of gaussian perturbations in VAE space vs PCA space. All the perturbed samples are shown in the PC space for comparison.

3.6 Sampling Extremes

To sample extreme scenarios, we look at the tail of the distribution and sample around that region. The step-by-step procedure is as below:

1. Divide that data into its respective orthants in both VAE and PCA latent dimensions.
2. For each orthant, select the top 20 percentile of the samples by their norm in each orthants to select the outliers.
3. Mutate these outliers in various iterations with various levels of gaussian noise to generate new samples.
4. Find the nearest neighbors in real data for all the mutated points.
5. Select the mutated points that have the largest nearest neighbor distance to real data points for each iteration in 3.

This procedure can be conducted in VAE latent space and PCA latent space as well.

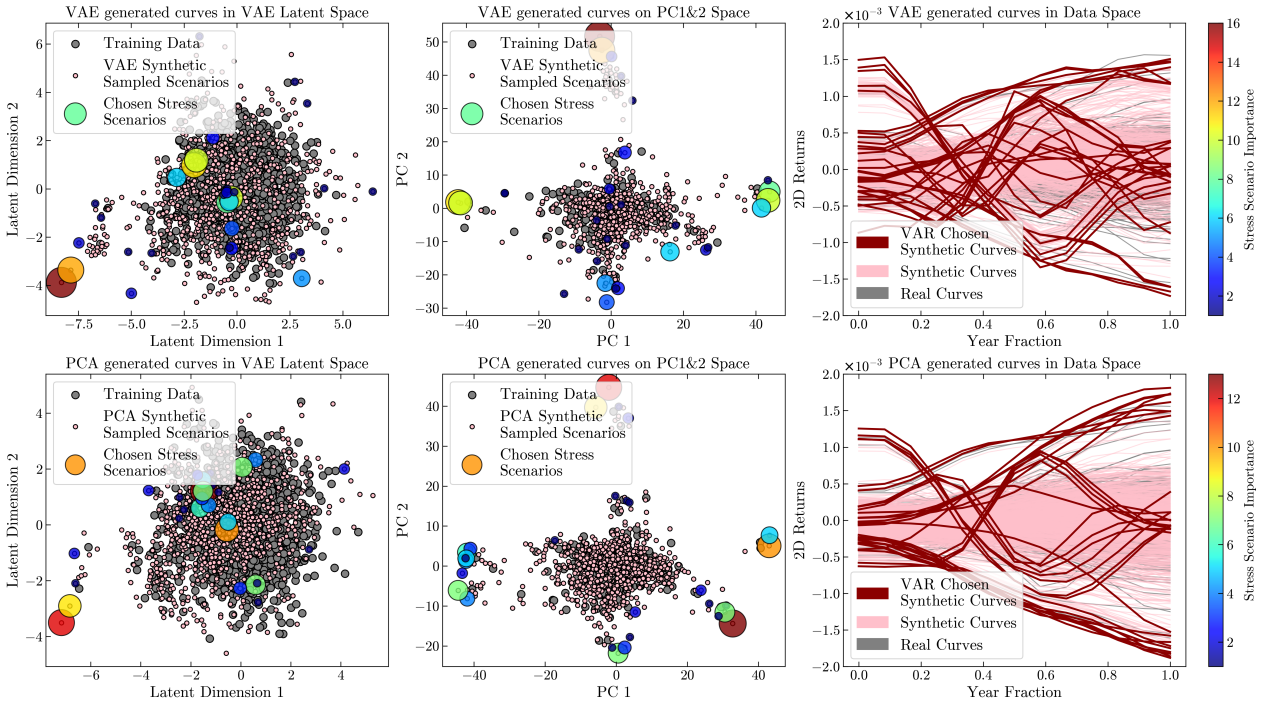


Fig. 3.11: Top panel shows the curves generated by farthest point sampling in VAE latent space. While the bottom panel shows the same data by doing farthest point sampling in PCA Space. The right most panel also shows curves chosen in VaR calculation for various portfolios described in Section 4. We can see that in case of PC space sampling, only extremes in PC1 and 2 are mostly chosen, while in the case of VAE generated curves, selected curves do not necessarily have to be in PC1, PC2 extremes as shown by the blue scenarios in the top mid figure. The color bar shows the "Stress Scenario Importance" which counts for how many portfolios a particular scenario was chosen as a stress scenario in ES-VaR calculation, as discussed in detail in the next chapter.

4 Hypothetical Scenarios in VaR Calculation

To show how the VAE and PCA synthetic scenarios can be used in practice and to further explore in what ways they can expand our risk coverage, we will now proceed with integrating them into our VaR calculations. Similar to Section 3.1, we will restrict the scope of our analysis to 3-Month TONA Futures, with a particular focus on the four closest-to-expiry instruments.

A simple test can be constructed by generating hypothetical portfolios with different combinations of 3-Month TONA Futures. Since the scope of the calculation is limited to four specific instruments, we can define a portfolio P_N as a vector of net quantities for each instrument's expiry respectively:

$$P_N = (I_{N1},\ I_{N2}, I_{N3}, I_{N4}) \tag{2}$$

where I_{N1}, I_{N2}, I_{N3}, I_{N4} are the net position quantities for the first, second, third, and fourth closest-to-expiry instrument, respectively. We further define the net position quantity as positive for net long positions and negative for net short positions. As an example, the four closest 3-Month TONA Futures expiries on 20250217 were in 202503, 202506, 202509 and 202512, respectively. A hypothetical sample portfolio P_X that has 2 long 202503 positions and one short 202509 position can then be defined as $P_X = (2, 0, 0, -1)$. For the purposes of this paper, we will create hypothetical portfolios for all -1, 0 and 1 net quantity combinations, resulting in $3^4 - 1 = 80$ portfolio combinations when excluding the empty $(0, 0, 0, 0)$ portfolio:

$$\begin{aligned} P_1 &= (-1, -1, -1, -1) \\ P_2 &= (-1, -1, -1,\ \ 0) \\ P_3 &= (-1, -1,\ \ 0, -1) \\ &\ldots \\ P_{80} &= (\ \ 1,\ \ 1,\ \ 1,\ \ 1) \end{aligned} \tag{3}$$

Additionally, we will use 20250217 as our sample calculation date and apply the same JGB calculation parameters as published on the JSCC website on the same date [20]. For comparison purposes, we will calculate VaR using three types of stress scenarios: (1) JSCC official stress scenarios, (2) JSCC official stress scenarios + VAE synthetic scenarios, and (3) JSCC official stress scenarios + PCA synthetic scenarios. This results in the following calculation input summary for our historical VaR calculation test:

Table 1: VaR calculation scope and parameters for analysis.

Parameter	Details
Calculation Date	20250217
Calculation Method	VaR (Expected Shortfall)
ES Stress Losses	2
ES Confidence Interval	97.5%
Historical Scenarios	Official scenarios published on the JSCC website (1250 scenarios).
Stress Scenarios	(1) Official JSCC scenarios. (2) Official JSCC scenarios + PCA synthetic scenarios. (3) Official JSCC scenarios + VAE synthetic scenarios.
Target Portfolios	All -1, 0, 1 net quantity combinations for 202503, 202506, 202509 and 202512 expiry 3-Month TONA Futures (80 combinations in total when excluding (0, 0, 0, 0)).

4.1 VaR Calculation Results

The relative impact, which we define as $\frac{\text{VaR}_{\text{Syntethic}}}{\text{VaR}_{\text{Base}}} - 1$, of adding hypothetical VAE and PCA scenarios to the set of stress scenarios is illustrated in Fig. 4.1a. As expected, adding additional stress scenarios always

result in equal or higher VaR numbers since we are giving the expected shortfall methodology additional opportunities to find scenarios with higher portfolio losses than before. In summary, some portfolios had no VaR impact, most portfolios have a VaR impact in the (0%, 5%] range and a few portfolios had an even higher impact than that. We can also see that the VAE scenarios tend to result in a slightly higher VaR impact compared to PCA.

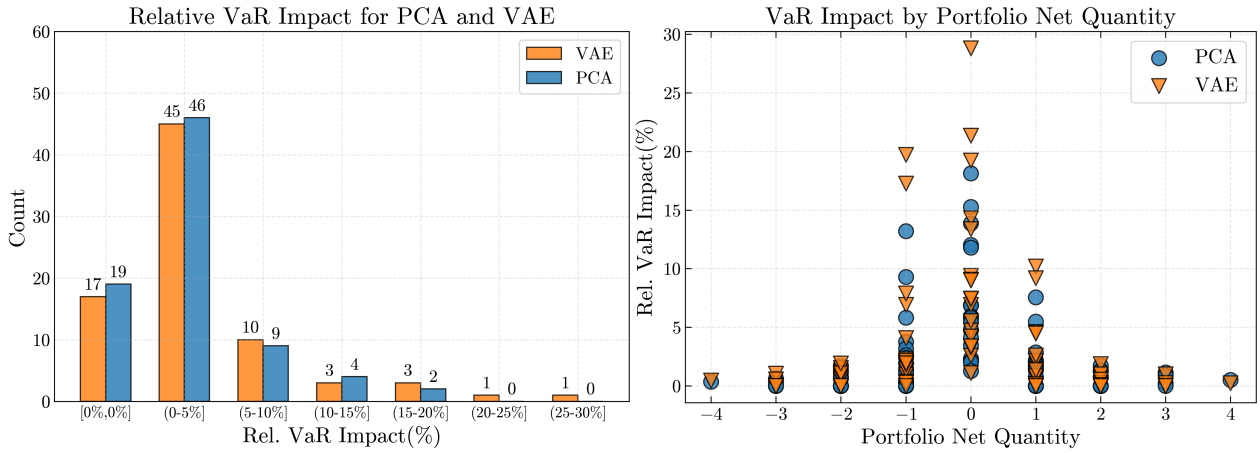
The top 5 portfolios with the highest VaR impact for VAE and PCA can be found in Table 2 and Table 3 respectively. Here we can also see an indication that portfolios with net quantity close to 0 across all instrument tenors ($\sum_{i=1}^{4} I_{\mathrm{Ni}}$) seem to have the largest VaR impact for both VAE and PCA, indicating that the synthetic scenarios contain new calendar spread scenarios that we did not have in the original set of scenarios. This tendency can also be seen in Fig. 4.1b, where we illustrate the VaR impact as a function of the portfolio net quantity using the same definition.

Table 2: Top 5 largest VaR impact portfolios for VAE generated scenarios. The scenario IDs will be defined in Section 4.2 and are based on portfolio selection frequency. The scenarios are referenced in the text as *vae01, vae02* or *pca01, pca02* etc.

I_{N1}	I_{N2}	I_{N3}	I_{N4}	**Net Qty.**	**VAE vs. Original Diff.**	**PCA vs. Original Diff.**	**VAE vs. PCA Diff.**	**VAE Selected Scenarios (*vae*–)**	**PCA Selected Scenarios (*pca*–)**
-1	1	1	-1	0	28.8%	18.1%	9.0%	01, 02	02, 15
-1	0	1	0	0	21.3%	15.3%	5.3%	01, 02	02, 04
-1	-1	1	0	-1	19.7%	13.2%	5.8%	01, 02	02, 04
-1	1	0	0	0	19.2%	13.9%	4.7%	01, 02	02, 04
-1	0	1	-1	-1	17.3%	9.3%	7.3%	01, 02	02, 15

Table 3: Top 5 largest VaR impact portfolios for PCA generated scenarios. The scenario IDs will be defined in Section 4.2 and are based on portfolio selection frequency.

I_{N1}	I_{N2}	I_{N3}	I_{N4}	**Net Qty.**	**VAE vs. Original Diff.**	**PCA vs. Original Diff.**	**VAE vs. PCA Diff.**	**VAE Selected Scenarios**	**PCA Selected Scenarios**
-1	1	1	-1	0	28.8%	18.1%	9.0%	01, 02	02, 15
-1	0	1	0	0	21.3%	15.3%	5.3%	01, 02	02, 04
-1	1	0	0	0	19.2%	13.9%	4.7%	01, 02	02, 04
-1	-1	1	0	-1	19.7%	13.2%	5.8%	01, 02	02, 04
0	1	-1	0	0	14.3%	12.0%	2.0%	12, 25	14, 05



(a) Overall relative VaR impact for all portfolios.

(b) Relative VaR impact by portfolio net quantity.

Fig. 4.1: Relative change in VaR after adding VAE/PCA synthetic scenarios as hypothetical stress scenarios is shown in Fig. 4.1a. Comparison of portfolio net quantity and VaR impact for both VAE and PCA calculations is shown in Fig. 4.1b.

4.2 VaR Selected Scenarios

We will now proceed with analyzing what type of scenarios that were selected by ES-VaR from the set of synthetic PCA and VAE scenarios. Out of the approximately 500 scenarios generated by the two methods, 31 VAE scenarios and 29 PCA scenarios were selected, respectively. Some scenarios were also selected by more than one portfolio, as illustrated in Fig. 4.2. To simplify further discussions, we will introduce hypothetical scenario IDs based on the same figure. The *vae01* scenario in the figure was selected by 16 portfolios while *pca01* was selected by 13 portfolios. In total, VAE scenarios were selected 116 times while PCA scenarios were selected 114 times. This number is relatively high when compared to the theoretical maximum number of selected stress scenarios, which given the calculation configuration described in Table 1 is portfolios $\times$ stress losses $= 80 \times 2 = 160$ stress scenarios.

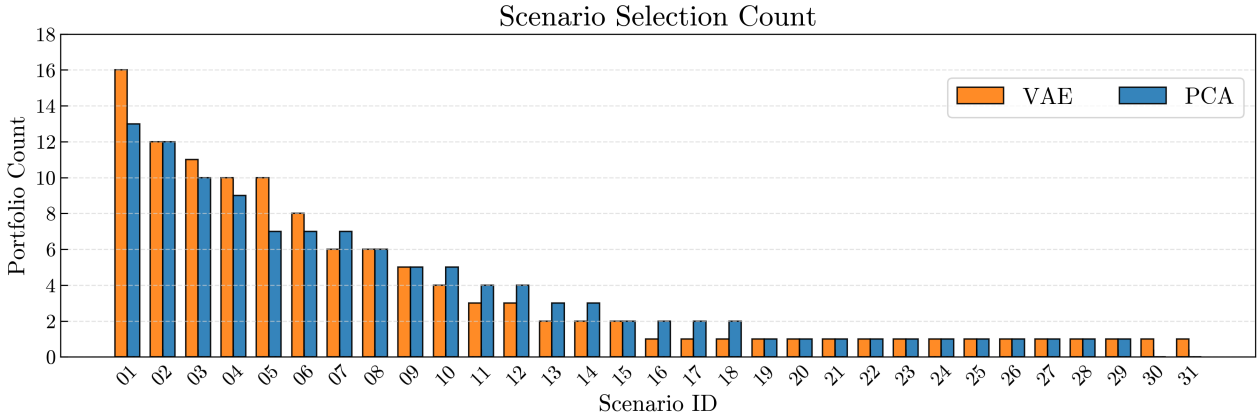
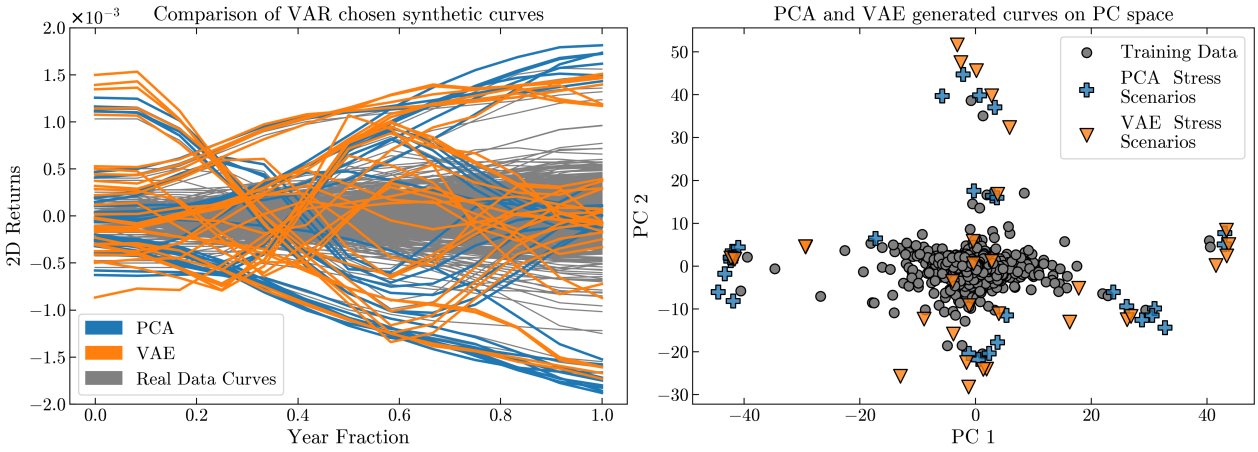


Fig. 4.2: Scenario selection count by scenario. The *vae01* scenario was selected by 16 portfolios while the *pca01* scenario was selected by 13 portfolios.

In Fig. 4.3a we can additionally see an overview of all hypothetical VAE and PCA curves that were selected by any portfolio in the ES calculations. In general, the PCA scenarios tend to be smoother while the VAE scenarios have more curve shape variations. The same scenarios can also be seen in Fig. 4.3b, where we project them on the first and second PCA components. Worth noticing is that the ES selected PCA scenarios are mostly found in the tails of the first and second PCA component distributions, which correspond to the main variations in the data (i.e., the dominant curve shapes). In contrast, some of the selected VAE scenarios

appear closer to the center of these distributions, suggesting that the VAE sampling introduces stressful scenarios that involve more subtle variations that are not explained by PCA1 and PCA2.



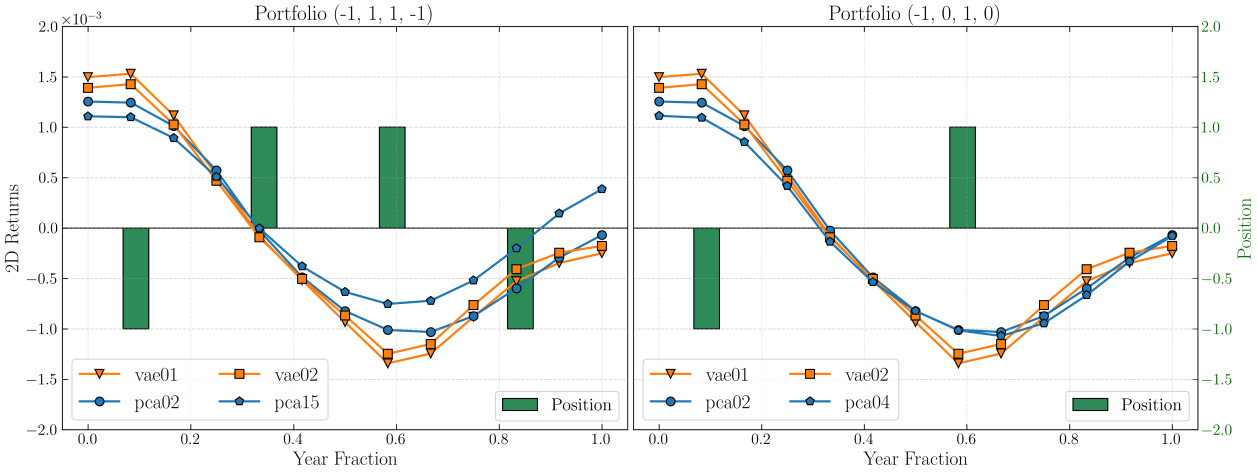
(a) Selected synthetic curves.

(b) Selected curves projected onto the PC1&2.

Fig. 4.3: VAE and PCA scenarios selected by ES for hypothetical portfolio calculations in data space and PC space.

4.3 Portfolio Specific Analysis

To get a better understanding about why a specific scenario was selected, we need to compare it with the specific portfolio contents. In Table 2 and Table 3, we observed that the two largest VaR impacts for both VAE and PCA were for a (-1, 1, 1, -1) and a (-1, 0, 1, 0) portfolio. The respective portfolios and scenarios are illustrated in Fig. 4.4a and Fig. 4.4b respectively.



(a) Portfolio (-1, 1, 1, -1) with the largest VaR impact.

(b) Portfolio (-1, 0, 1, 0) with the 2nd largest VaR impact.

Fig. 4.4: Selected VAE and PCA scenarios for portfolios (-1, 1, 1, -1) in Fig. 4.4a and (-1, 0, 1, 0) in Fig. 4.4b. Portfolio (-1, 1, 1, -1) had the largest VaR impact of all portfolios for both PCA and VAE generated scenarios. Portfolio (-1, 0, 1, 0) in Fig. 4.4b had the second largest VaR impact of all portfolios for both PCA and VAE. Note that the scenario returns with opposite signs to the portfolio quantities result in losses.

As expected, the return scenarios that result in the highest portfolio losses have opposite signs as the position quantities. The (-1, 1, 1, -1) portfolio has a short-long-long-short position combination and hence up-down-up scenarios such as *vae01, vae02, pca02* and *pca15* are causing large losses - especially on the first and third tenors. In Fig. 4.4a we can also see that the VAE and PCA scenarios have similar shapes, but the VAE

scenarios have higher and lower returns for the first and third tenor respectively, which explains the VaR impact difference that was observed in Table 2 and Table 3. A similar behavior can be seen in Fig. 4.4b, where a short-long position combination resulted in up-down scenarios being selected for both VAE and PCA.

An example where ES found different scenario curve shapes for VAE and PCA can be found for the (-1, 1, -1, 1) portfolio as illustrated in Fig. 4.5. For this portfolio, the VAE VaR impact was 5.5% while the PCA VaR impact was 1.3%, resulting in the sixth-largest difference between VAE and PCA. As seen in the same figure, the PCA scenarios resulted in losses for the first tenor (short position, positive scenario returns) and fourth tenor (long position, negative scenario returns) that were to a large degree offset by the profits in the third tenor (short position, negative scenario returns). This can be compared to the VAE scenarios that resulted in losses for most tenors (short-long-short-long portfolio, up-down-up-down scenario returns).

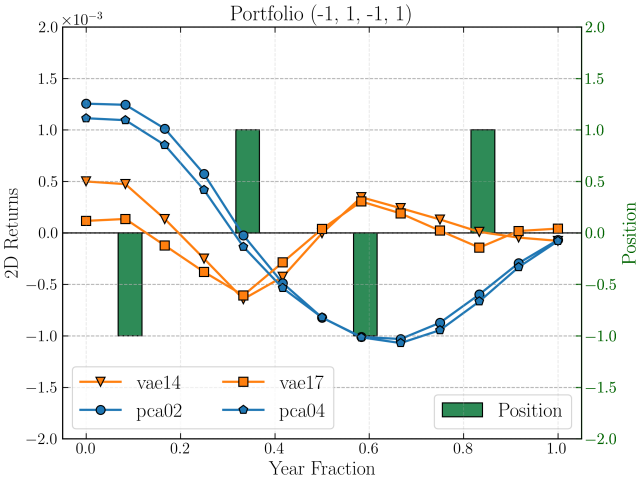


Fig. 4.5: Selected VAE and PCA scenarios for portfolio (-1, 1, -1, 1). Note that the scenario returns with opposite signs to the portfolio quantities result in losses.

4.4 Synthetic and Historical Scenario Comparison

The VaR results that we have discussed in previous sections were created by adding new synthetic scenarios to the pool of hypothetical stress scenarios. In theory, these scenarios could be identical to real historical training data scenarios and still result in a VaR impact though, if those scenarios were both outside the historical VaR window and not selected as a historical stress date. This motivates us to make an additional comparison where we compare the VaR impact of using:

1. JSCC original scenarios + training data scenarios.
2. JSCC original scenarios + training data scenarios + VAE synthetic scenarios.
3. JSCC original scenarios + training data scenarios + PCA synthetic scenarios.

All other calculation parameters are the same as defined in Table 1.

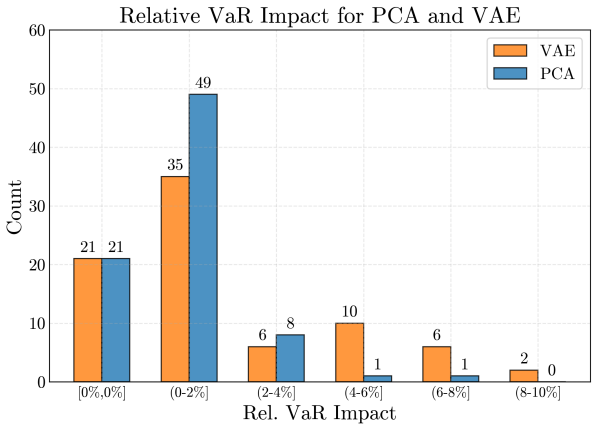


Fig. 4.6: Relative VaR impact after adding VAE and PCA synthetic scenarios as hypothetical stress scenarios on top of original stress scenarios and training data scenarios.

As seen in Fig. 4.6, the VaR impact is not as high as in previous comparisons since our reference calculation produces higher base numbers than before. Nonetheless, we still have 59 portfolios for both VAE and PCA each that have a non-zero VaR impact, indicating that both the VAE and PCA sampling methods produce new scenario combinations that result in higher portfolio losses than the historical training data. The top 5 portfolios with the largest VAE and PCA VaR impact can be found in Table 4 and Table 5 respectively. We can additionally see the selected scenario for the portfolio with the fifth largest VAE impact, (1, -1, -1, 1), in Fig. 4.7.

Table 4: Top 5 largest VaR impact portfolios for VAE generated scenarios compared to adding synthetic scenarios on top of original stress scenarios and training data scenarios. N/A indicates that a non-synthetic scenario was selected.

I_{N1}	I_{N2}	I_{N3}	I_{N4}	**Net Qty.**	**VAE vs. Original Diff.**	**PCA vs. Original Diff.**	**VAE vs. PCA Diff.**	**VAE Selected Scenarios (*vae*–)**	**PCA Selected Scenarios (*pca*–)**
0	1	-1	0	0	8.6%	6.4%	2.0%	12, 25	14, 05
-1	1	0	0	0	8.4%	3.6%	4.7%	01, 02	02, 04
-1	1	1	-1	0	7.9%	0.0%	7.9%	01, 02	N/A, N/A
-1	1	1	-1	0	7.1%	1.7%	5.3%	01, 02	02, 04
1	-1	-1	1	0	6.7%	2.5%	4.1%	09, 26	18,05

Table 5: Top 5 largest VaR impact portfolios for PCA generated scenarios compared to adding synthetic scenarios on top of original stress scenarios and training data scenarios. N/A indicates that a non-synthetic scenario was selected.

I_{N1}	I_{N2}	I_{N3}	I_{N4}	**Net Qty.**	**VAE vs. Original Diff.**	**PCA vs. Original Diff.**	**VAE vs. PCA Diff.**	**VAE Selected Scenarios**	**PCA Selected Scenarios**
0	1	-1	0	0	8.6%	6.4%	2.0%	12, 25	14, 05
1	1	-1	0	1	6.6%	4.1%	2.4%	10, 12	05, 14
-1	0	0	1	0	4.6%	3.6%	0.9%	01, N/A	N/A, 02
-1	1	0	0	0	8.4%	3.5%	4.7%	01, 02	02, 04
-1	-1	0	1	-1	4.2%	3.2%	1.0%	01, 02	02, N/A

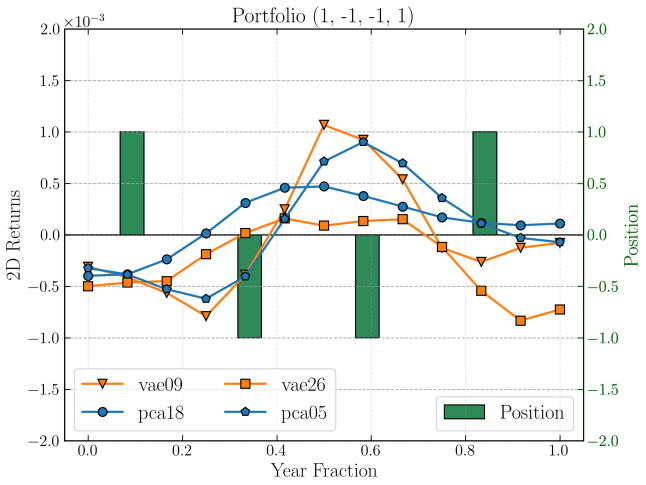
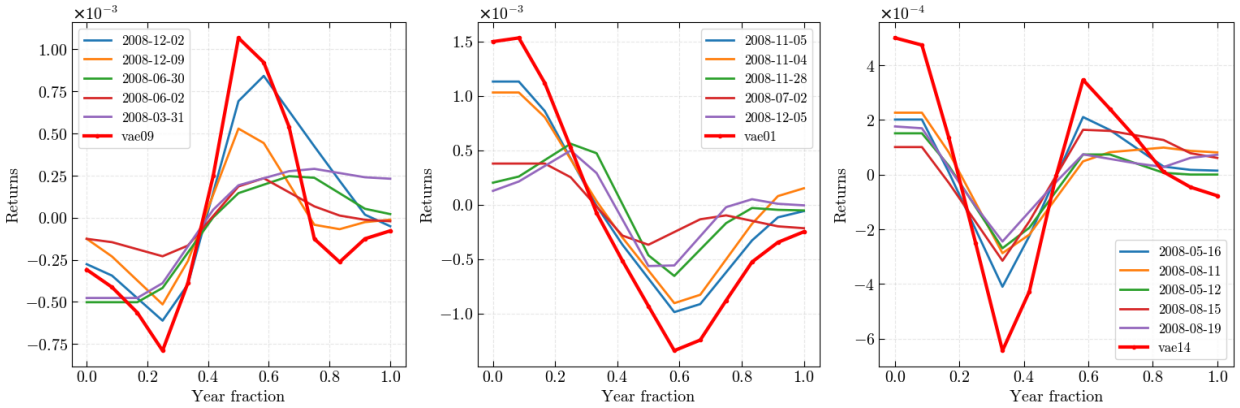


Fig. 4.7: Selected VAE and PCA scenarios for a (1, -1, -1, 1) portfolio with 6.7% and 2.5% VaR impact respectively. VAE produces a 4.1% higher VaR result than PCA.

An additional comparison can be seen in Fig. 4.8a, where we compare the selected VAE synthetic scenario with the closest surrounding real data scenarios in latent space. As seen in the figure, the synthetic scenario has magnified highs and lows, resulting in it being selected over its real data counterparts. The same type of comparisons can be seen in Fig. 4.8b and Fig. 4.8c for the previously discussed *vae01* and *vae14* scenarios, where we can observe similar behaviors. As seen in all three figures, all selected synthetic scenarios were close to historical 2008 scenarios, but with some variation that made the losses even higher for specific portfolios.



(a) Synthetic *vae09* sample. **(b)** Synthetic *vae01* sample. **(c)** Synthetic *vae14* sample.

Fig. 4.8: Selected synthetic scenarios (thick red curve) compared to the closest training data neighbors (thin-colored).

4.5 VaR Impact Summary

In this chapter we explored how introducing VAE and PCA scenarios affected the expected shortfall calculations for a set of hypothetical portfolios. We showed that both methods were able to find new scenarios that resulted in a VaR impact in the 0% – 30% range. We further discussed on portfolio level why specific scenarios were selected and compared some characteristics between the selected PCA and VAE scenarios. One observation was that the selected VAE scenarios were able to capture more subtle variations that could not be explained by the first and second PCA components. In summary, the VAE synthetic scenarios resulted in higher VaR impact than PCA due to having more curve shape variations in combination with more extreme highs and lows.

5 Discussion

Real world data is often complicated and has non-linear relationships. It usually does not follow a normal distribution or any other standard distributions. If we try to model real world data using simpler models, we will be imposing assumptions which are not realistic. Financial world data is typically non-gaussian and non-linear. Correlations can strengthen, change, and break in different regimes. Generative AI methods go beyond classical statistical methods and learn directly from the data without imposing any simplifying assumptions on the data. Due to the flexibility of neural networks, they can capture complicated dependencies among variables. They can learn smooth data distributions and can help us generate synthetic yet realistic looking data.

In this white paper, we looked at the 3-Month TONA futures 2-day returns curves and explored generating synthetic data using both PCA and VAE. We saw that sampling PCA space using a multivariate gaussian failed to capture tail events in the data which is expected as the gaussian distribution underestimates the tail probability present in real data. PCA also does not provide us a general-purpose sampling methodology. Taking a particular data sample and adding gaussian perturbation according to the principal component covariance matrix leads to dominance of first and second principal component which translates to level and slope in general. Generating new samples via perturbing existing extreme samples using global covariance in a region where that covariance might not hold will lead to poor and non-authentic data samples.

VAEs do a non-linear transformation on the data and create a probabilistic latent space. Since they are probabilistic models by design, they can be used for generative modelling. VAE provides a general mechanism for sampling in the latent space using the gaussian prior imposed on the latent space. The gaussian prior on latent space regularizes it towards being closer to (but not-necessarily) gaussian. By sampling in the VAE latent space and decoding back we can capture nonlinear dependencies between features as well. Unlike PCA based sampling where first and second principal components dominate, if we do gaussian perturbations in VAE latent space, these are translated into non-gaussian perturbations in data space. VAE lets other modes and factors in the data have more weight when needed rather than placing most weight on PC1&2 only. This can be seen in the right-most column of Fig. 3.10 as well.

VAE's latent space is particularly useful for realistic interpolation. As shown in Fig. 3.6, when linearly interpolating between scenarios in VAE latent space, the interpolation trajectory closely follows the contours of the data distribution unlike data space linear interpolation. Furthermore, we employed hierarchical VAEs to sample from the whole data distribution using only multivariate gaussian samples of the latent space.

We also sample extreme events in our data distribution using the strategy mentioned in Section 3.6. We use the same methodology for both VAE and PCA latent representations, and find that PCA based extreme point sampling fails to capture extreme scenarios described by low variance PC dimensions. It also suffers from the unnatural dominance of first and second PCA in regions where they should not be dominant. We can have scenarios where the highest returns are small but distributed in such a way that they can create significant losses for certain portfolios.

Subsequently, in Section 4 we explored how introducing PCA and VAE generated hypothetical stress scenarios affected our expected shortfall calculations for a set of hypothetical sample portfolios. When adding the generated scenarios to the set of original historical scenarios and stress scenarios, we noticed a VaR impact for both PCA and VAE in the 0%-30% range. In Section 4.4 we made a similar comparison while also including the training data set of historical scenarios in both calculations, which resulted in VaR impacts in the 0%-10% range. Since both methods were able to output new scenarios that resulted in higher losses than our original scenarios, we can argue that they help us to improve our risk coverage and risk understanding. Adding new synthetic scenarios to VaR should be done with great care though to avoid too big or unwanted impact on the calculation output. When comparing VAE and PCA results, we noticed that VAE had a higher VaR impact due to having more curve shape variations and more extreme highs and lows. These are both desirable properties when looking for extreme scenarios that can improve our risk coverage. When further looking at the selected scenarios, we could see that they shared similar characteristics and shapes as some original training data though. This is by design but still worth mentioning – we should not expect our introduced sampling methodology to generate scenarios that are completely different from the training data. Also worth mentioning is that within the scope of this paper, we tested a small set of hypothetical portfolios

using a single calculation date. The time to maturity of the instruments that we are using and the position composition of the portfolios will determine what type of scenarios that generate the highest losses. Testing additional portfolio combinations and more calculation dates would be beneficial but was determined out of scope for the purposes of this paper.

Our original goal was to find a way to sample extreme but plausible scenarios. Plausibility is hard to define in the case of abstract financial data. For data such as pictures, voices or music, evolution has equipped us with excellent ability to discern real from fake and plausible from implausible. The challenge, however, in the case of financial data, is that it is hard for someone to say what data is realistic or not just by looking at it apart from a highly trained expert. Therefore, we can only work with statistical measures and see if our generated data follows a similar data distribution as the actual data or not. If it follows a similar distribution, we can assume that it is realistic and plausible enough. We see in Fig. 3.6 and Fig. 3.9 that our interpolations and synthetic samples follow the contours of the distribution and preserve the shape of the data distribution respectively. Furthermore, many of these scenarios were selected by ES showing that they are extreme enough to create a significant change in a portfolio's value.

Briefly, the use of generative AI methods, especially for generating synthetic market data, helps us in:

1. Jointly modelling complex dependencies among multiple risk factors.
2. Remove or reduce the need for ad hoc methods such as copula fittings, ad hoc sampling techniques.

However, with any technology there are some drawbacks, and cautions we need to take. Generative AI is based on complex models and architecture and have complex mathematics behind them. They are black boxes in two ways:

1. Due to their architectural complexity, their decision-making and predictions are hard to explain.
2. Mathematics governing them is not easily accessible.

This makes AI powerful but also hard to understand at the same time which can make it difficult for users to fully understand the limitations. Another bottleneck to adoption is the lack of data. AI models require a large amount of high-quality data to be trained. Moreover, there is no algorithm to help us determine the best architecture, and various other hyperparameters involved in training neural networks, or even how long to train. Usually, it is a combination of experimentation, intuition, and some problem specific details. There is indeed a lot of trial and error involved in training AI models.

5.1 Final Words

The linearity of PCA makes it very explainable, straightforward to implement and use. However, generating data using PCA transform requires careful sampling which is often non-trivial. Sparsity and lack of continuity of the principal component space additionally hinders its use as a generative model. In smaller dimensions this might not be a major issue for experts but when the dimensionality of the data increases, our intuitions can fail us, and computational complexity explodes. In short synthetic data generation by manually varying principal components does allow us to generate data but requires careful human expert's supervision. On the other hand, VAE converts complex high-dimensional data distributions to simple and continuous latent distributions and has sound theoretical framework behind it ensuring its ability as a generative model. The simpler and lower-dimensional latent spaces allow better sampling and are easier to handle due to their lower dimensionality. Both approaches have their own merit and ideally can be used together to generate synthetic data. For example, we can remove the first and second principal components in data which are usually highly correlated with the market and model the remaining complex dependencies in the data using VAE. Conditional models can also be built to predict the residual principal components distribution based on first and second principal components. Furthermore, using both approaches together also provides more explainability to VAE or other neural network based generative models.

5.2 Future Directions

In this white paper we explored the feasibility of using generative AI models for generating synthetic financial data at JSCC. Further directions in this research are considering the time dependency in the data, modelling multiple risk factors, and using more advanced models such as generative adversarial networks (GANs) [11], adversarial autoencoders [17] or diffusion models [14] [15]. Predictive modelling in the time series is particularly interesting from the perspective of risk management as well. Apart from that generative AI models can also be used for anomaly detections where they can flag unusual patterns in the data.

6 Acknowledgements

We are grateful for the support we received for writing this working paper. First and foremost, we would like to express our sincere gratitude to Mr. Konuma, Yasuyuki, President and CEO of JSCC, Mr. Miwa, Mitsuo, Executive officer, and Mr. Tamura, Yasuhiko, Executive officer, for giving us the opportunity to write this paper in the critical field of AI usage in CCP risk management. We would also like to extend our sincerest thanks to Mr. Hamasaki, Keiichi, and Mr. Fujimoto, Yuta, from the Listed Products Clearing Department at JSCC for their selection of the theme, analysis approach, and valuable feedback.

References

[1] A. Rehlon and N. Dan, "Central counterparties: What are they, why do they matter and how does the Bank supervise them?" Bank of England, Jun. 2013.

[2] "3-Month TONA Futures," *Japan Exchange Group*. https://www.jpx.co.jp/english/derivatives/products/interest-rate/3m-tona-futures/01.html.

[3] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback." arXiv, Mar. 2022. doi: 10.48550/arXiv.2203.02155.

[4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models." arXiv, Apr. 2022. doi: 10.48550/arXiv.2112.10752.

[5] A. Ramesh *et al.*, "Zero-Shot Text-to-Image Generation." arXiv, Feb. 2021. doi: 10.48550/arXiv.2102.12092.

[6] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.

[7] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[8] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[9] K. Pearson, "Principal components analysis," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 6, no. 2, p. 559, 1901.

[10] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes." arXiv, 2013. doi: 10.48550/arXiv.1312.6114.

[11] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in neural information processing systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014.

[12] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using Real NVP." arXiv, Feb. 2017. doi: 10.48550/arXiv.1605.08803.

[13] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep Unsupervised Learning using Nonequilibrium Thermodynamics." arXiv, Nov. 2015. doi: 10.48550/arXiv.1503.03585.

[14] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, pp. 6840–6851.

[15] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-Based Generative Modeling through Stochastic Differential Equations." arXiv, Feb. 2021. doi: 10.48550/arXiv.2011.13456.

[16] "Midjourney," *Midjourney*. https://www.midjourney.com/website.

[17] A. Plumerault, H. L. Borgne, and C. Hudelot, "AVAE: Adversarial variational auto encoder," *CoRR*, vol. abs/2012.11551, 2020, Available: https://arxiv.org/abs/2012.11551

[18] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models." arXiv, Apr. 2022. doi: 10.48550/arXiv.2112.10752.

[19] D. J. Rezende and S. Mohamed, "Variational Inference with Normalizing Flows," *arXiv.org*. https://arxiv.org/abs/1505.05770v6, May 2015.

[20] Japan Exchange Group, "Archive of Weekday Files." https://jscc-h.jpx.co.jp/jscc/listed-derivatives/weekday/VaRParameter_20250217_1600.csv.