The use of Variational AutoEncoders in a financial context

Student: Maurits van den Oever
Student number: 2613642
Supervisor: Svetlana Borovkova

April 2022

Abstract

Autoencoders are a type of neural network architecture focused on reducing the dimensionality of data. With proper regulation of the reduced, usually referred to as latent, space, the model can learn distributional characteristics, complex dependencies and mechanics behind the data the model is trained on. For instance, when the latent space is regulated as a distribution, the model can be used for scenario generation. In this paper, the performance of the algorithm is analysed when these distributional regulations change. Furthermore, the autoencoder is employed for financial risk, namely to successfully obtain a Value-at-Risk. Lastly, noisy and incomplete at-the-money implied volatility curves are corrected using this architecture. This is partially facilitated by weighing the reconstruction objective of the model training with the liquidity of the data, resulting in higher reconstruction accuracy for liquid data points, and more autonomy of the model with data points that are less liquid. All in all, we establish that autoencoders are successful in learning and replicating distributions of financial data.

Contents

1 Introduction

The aim of this thesis is to investigate how efficient autoencoders are in modelling distribution of financial data, as well as some practical applications of autoencoders in a financial context. First, a general introduction to autoencoders is given. After, the problems and questions that the thesis aims to answer are discussed. Lastly, the remaining structure of the thesis is outlined.

1.1 An introduction to autoencoders

An autoencoder is a Deep Learning model that is primarily used to reduce dimensionality of data. Deep Learning refers to a subset of machine learning models that rely on Neural Networks (NN) for their performance. Comprised of an input layer, a set of hidden layers, and an output layer, the NN aims to approximate functions in a highly non-linear fashion. This can be applied to the prediction of a dependent variable given a set of independent variables, but can also be used to learn the distribution or hidden mechanism of a particular set of data. Models that aim to perform the latter are known as generative models, as they can often generate synthetic data that resembles the original data after proper training. Autoencoders fall under the generative model category, since they aim to model the distribution of the data.

In figure 1, a general architecture of an autoencoder is shown. The original data, referred to as X, is encoded to a reduced, also known as latent, space Z. After, the latent data Z is decoded again such that it matches the original data as closely as possible, or $X \approx X'$. The encoding and decoding process are both handled by NNs, where q_ϕ is the encoding process with parameters ϕ, and p_ψ is the decoding process with parameters ψ. The main reason that the original data is reconstructed in the training process, is so that the most relevant information of the data is included in Z. In an ideal situation, $X = X'$, or the reconstructed data exactly matches the original data after passing through the information bottleneck. This implies that all the information in X is captured in Z, despite the fact that Z has less dimensions than X. The main takeaway from this is that by reducing the reconstruction error implicitly forces the model to include the most relevant information from X into Z.

In a Variational AutoEncoder (VAE), the latent space Z is a probability space. The VAE is fitted by choosing the weights of the NNs such that an objective function is optimised. The objective function depends on the optimisation of reconstruction error, or the differences between X and X', and the regularisation of the latent space. Ideally, the autoencoder can transform the original data in such a way that it is distributed according to the latent space assumptions, and is subsequently able to transform the latent data back in its original form. Often, these two objectives present themselves as a trade-off. A more heavily regularised latent space will result in a larger reconstruction error.

Details on the chosen objective functions are discussed in section 4.

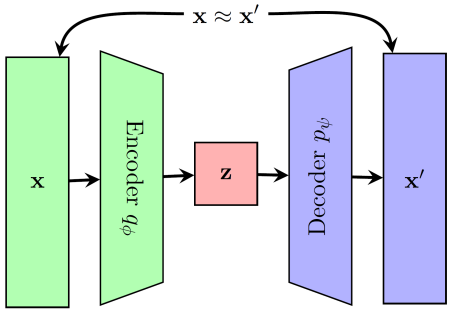

Figure 1: Typical structure of an AutoEncoder. Picture taken from (Ning et al., 2022)

1.2 Problem description

As stated in the beginning of the introduction, the aim of this thesis is to establish the applicability of VAEs on financial data. VAEs are still relatively new in financial literature, so this thesis aims to build a case on the usability and effectiveness of autoencoders for financial applications. With this in mind, the main research question is formulated as follows:

How effective are autoencoders in learning and transforming the distribution of financial data?

This question elicits three subquestions:

- *Can autoencoders effectively transform the distribution of data?*
- *Can autoencoders learn the distribution of financial returns?*
- *Can autoencoders correct partially missing implied volatility curves?*

For the first subquestion, a reconstruction error analysis is done for different types of VAEs. The distributional properties according to which the latent space is regulated are changed, to analyze how this affects model performance. This is tested according to multiple types of synthetic and real data. If the models are not effective in transforming distributions, there will be a difference in the performance between different types of VAEs. This analysis also delves into how well autoencoders can model distributions, as accurate reconstructions are the result of having properly learned the distribution of the original data. Implicitly, we also investigate how model performance changes based on the size of the latent space. Furthermore, different levels of correlation in the synthetic

datasets are checked as well, to see how the models take advantage of dependency structures. The performance of the models are also compared to PCA, to see if autoencoders can outperform a linear method.

The second subquestion is investigated by developing a novel way to obtain Value-at-Risk with the use of VAEs in combination with GARCH. This application also tackles a known problem, where GARCH in a large multivariate setting becomes unwieldy due to the exponential increase in parameters for the time-varying covariance matrix estimation. It is possible to make direct assumptions on the covariance such as constant correlation GARCH, but this is not always ideal. Another way to approach this issue is by PCA GARCH (Geng, 2007). In this situation, PCA is used to model the cross-sectional covariances, while GARCH is used to model the time series variance. In this thesis, the PCA GARCH methodology is adapted for VAEs. Details on the methodology are discussed in section 4.3. If VAEs can be used to accurately model VaR, it means that the model is capable of learning the return distribution, even when its variance is heteroskedastic.

For the last subquestion, at-the-money implied volatility curves containing missing values are corrected using the autoencoders. The objective function of the model in the training process is changed such that only the reconstructed curve is taken into account. The reconstruction error is weighted with liquidity as well. Details on how this is achieved can be found in section 4.4. The rationale behind this is that observations driven by a large amount of options are considered more robust, while observations driven by smaller amount of options are noisy and inaccurate. Thus, when the model reconstructs the implied volatility curves it should put more emphasis on robust observations. For this reason, the autoencoder is scrutinized for these robust observations, while it is given more autonomy when reconstructing noisy or even missing observations. If the autoencoders can successfully correct partially missing curves, it can be concluded that the models can learn the mechanisms that form these curves.

1.3 Structure of the thesis

The thesis onward is structured in the following manner. Section 2 outlines the literature backgroud of autoencoders, both in general and its current applications in finance. Section 3 describes the data used for these analyses. Then, section 4 describes the methods and theory in more detail, also providing some background on the optimisation of the models. Section 5 describes the results of the applied methods. After, section 6.1 discusses the main takeaways of the results as well as recommendations for further research are outlined. Lastly, the bibliography and the appendix are shown.

2 Literature review

Autoencoders were originally proposed mainly for dimensionality reduction of data. Then referred to as auto-association models, Bourlard & Kamp (1988) showed that autoencoder model parameters can be derived using linear techniques that build on singular value decomposition when the autoencoder model has linear output units. This was a useful result as limitations of gradient techniques at the time could result in the finding of local, as opposed to global, optima. Kramer (1991) outlined the uses of autoencoders as a nonlinear alternative to principal component analysis (PCA). In this paper, the autoencoders were successful in removing any type of nonlinear correlation between variables, and reducing dimensionality in the process. Kramer also mentioned that this technology experimentally outperforms PCA in reduction and explanation. The scientific literature has numerous recent additions in the connection between PCA and autoencoders for different contexts. One such context is the comparison in performance in face recognition by Siwek & Osowski (2017), where autoencoding principle showed numerical superiority over PCA. Wetzel (2017) compared performance of PCA and autoencoders of recognizing physical phase transitions, where PCA and VAEs seemed to share similarities in performance. Almotiri et al. (2017) used latent representations of images of handwriting to feed into an NN for classification purposes. The results show that latent representations produced by PCA led to slightly lower predictive accuracy than autoencoders. Dai et al. (2018) made connections between robust PCA and VAEs when analysing the stochastic optimization process of a VAE. They conclude that VAEs are a natural extension of PCA when analysing non-linear dependencies in datasets.

2.1 Autoencoders for noisy and sparse data

Autoencoders also have their literary basis in noise reduction in datasets. This application was first mentioned by Kramer (1992) when talking about correcting noise in sensory data. There are recent publications that demonstrate this application as well. Shivakumar & Georgiou (2016) applied denoising autoencoders in the area of speech enhancement, a processing tool used to clean audio data for speech recognition algorithms. They found that employing autoencoders showed significant increases in performance measured with noise reduction, speech distortion and Perceptual Evaluation of Speech Quality metrics. Bonfigli et al. (2018) used autoencoders in the denoising of Non-Intrusive Load Monitoring, a way to measure electricity usage. Again, the autoencoder is used as a pre-training tool, to make it easier for a subsequent NN to find structure in the data. Autoencoders are shown to have a positive effect of algorithm performance, and to increase the robustness to noise in the data. Bhowick (2019) used the generative properties of autoencoders to create noise free reconstructions of geophysical data. In this case, two autoencoders are stacked in such a way that certain data enter during the NN, instead of beforehand. Chiang et

al. researched noise reduction by autoencoders in electrocardiograms (ECG) in 2019. Due to the way ECGs are made, they are prone to noise, which can lead to spurious conclusions concerning a patients cardiovascular health. Convolutional autoencoders were again found to be able to reduce noise in this context. As a final example, de Oliveira & Bekooij (2020) applied convolutional autoencoders to the denoising of range-Doppler maps, a way of measuring the movement of objects from a distance. Because the data becomes less noisy, the possibility of recognizing distant objects that generate faint data is facilitated more easily.

Of course, being able to correct noise in data also gives way for a similar application, namely the imputation of sparse data. Beaulieu-Jones & Moore (2017) compared ALS progression imputation performance between different methods. According to their results, autoencoders show strong imputational and predictional accuracy. Similar results were obtained by Tran et al. (2017), who used a cascaded residual autoencoder to impute missing modalities that can be the result of malfunctions in multidimensional sensory data. Furthermore, their imputational efforts also show increased accuracy in object recognition. Convolutional autoencoders were also used by Asadi & Regan (2019) to impute spatio-temporal data, again in the context of missing values in sensory output when measuring traffic flow data. Litany et al. (2017) used similarly structured autoencoders to compelete missing visual data, showing impressive results in the imputation of incomplete human body scans and face meshes. Noise reduction and other data correction applications of autoencoders are also present in financial literature, which are mentioned below.

2.2 Autoencoders in finance

In the context of finance specifically, Deep Learning is used, for instance, for exchange rate and market prediction, stock market trading, default and credit risk, portfolio management, macroeconomic predictions and oil price predictions (Huang et al., 2020). In the context of autoencoders, they have been used among other things to construct Implied Volatility (IV) surfaces when market data is partially incomplete or noisy (Bergeron et al., 2021). The authors tested the performance of a β-VAE on different foreign exchange OTC option markets, and obtaining significantly smaller errors in the construction of IV surfaces than traditional statistical methods. This implies that these types of models are a good way to analyse IV surfaces empirically. A similar methodology was applied later by Ning et al. in 2022. Their approach differs in the way that first, stochastic differential equation models were fitted to data, to subsequently use a VAE to learn the distribution of the SDE parameter space. Doing this, the VAE was successful in generating IV surfaces using its latent space distribution, implying that the generative characteristic of VAEs can also be used effectively for different ends.

As mentioned before, data denoising and compeletion through autoencoders

also have related publications in finance. Such an example can be found in the paper by Kondratyev (2018), who used VAEs to correct noisy forward curves based on brent crude oil forward prices and USD interest rate swaps. Kondratyev later extended this framework somewhat together with Sokol in 2020 in a CompatibL risk conference. Their presentation showcased their autoencoder approach being able to accurately construct currency interest rate curves, even when presented with sparse data.

So, it is apparent that autoencoders can capture dynamics of surfaces and curves, but what about distributional properties? This question is more difficult to answer, as autoencoder literature in distributional applications is limited. Literature on NNs however would suggest that it is possible for them to do so. Horger et al. (2018) successfully mapped uniform distribution samples to to any explicitly known probability density function. This was done by minimizing the distribution distance between the model output and the target distribution. The obtained result by Horger et al. would suggest that an autoencoder will be able to do the same, as the encoding and decoding process are NNs. What remains to be seen, however, is how this translates to data of which the probability density function is not explicitly known, as is the case with many real world data.

2.3 Contribution to literature

As stated in the introduction of this thesis, autoencoders are relatively new to literature in finance. This thesis aims to solidify the usability of these models within a financial context. Applications such as noise reduction and missing value imputation in financial data have been done before. Other forms of applications however, do not yet have a literary basis in finance.

Firstly, we investigate how model performance changes based on different latent space regulations. In most papers, the Kullback-Leibler divergence is used, which is serviceable for Gaussian latent space distributions only. To regulate the latent space as another distribution, another metric needs to be selected. This has not yet been discussed in a financial context. Since financial returns distributions are share no similarities to the Gaussian, it is possible that non-Gaussian latent space regulations will provide an advantage in modelling the return distribution.

Secondly, a contribution is made to literature by developing a method to obtain VaR using a variational autoencoder. Autoencoders have not yet been used in finance in this manner. The underlying rationale is that if the autoencoder can learn a return distribution. After learning this distribution, it is possible that the autoencoder can use this information to estimate a VaR. If the VaR estimate is correct, determined by backtesting, that we can conclude that the autoencoder can properly learn the return distribution.

Lastly, partially incomplete implied volatility curves are constructed using the autoencoder. Autoencoders have been used for this type of application before in finance. However, what this thesis adds to literature is that the reconstruction accuracy is emphasised for implied volatility observations driven by higher liquidity. Furthermore, the model receives full autonomy for the missing observations. If the reconstructed curves are correct, the model is concluded to be able to learn implied volatility curve characteristics.

3 Data description

In this section, the different data sets used are discussed. The methodology of this thesis is achieved using both synthetic data, as well as, real data. Synthetic data is used for the analysis of the first subquestion discussed in section 1.2. This is to ensure identically distributed observations throughout the entire sample, which also gives way for out-of-sample analysis. Concerning real data, two data sets are used. There is a dataset of daily returns, which is used for the analysis of the first two subquestions. A dataset containing at-the-money implied volatilities is used for the last subquestion, to analyze how autoencoders perform in reconstructing implied volatility curves.

3.1 Synthetic data

The performance analysis of the VAEs is done partly on simulated sets of data. The first simulated sets of data are normally distributed with an arbitrary mean and variance, and contain 12 features, each with 10,000 observations. The reason for the mean and variance being arbitrary is because the data are normalized before the VAE is fitted, and denormalized once the reconstruction is performed. The normal datasets are generated according to a Gaussian copula with different correlation parameters ρ, namely 0.25, 0.5, and 0.75. For each of these correlation parameters multiple sets of data are generated, varying in the amount of 'factors' driving the data. The different amount of factors d are 1, 2, 3, 4, 6, and 12. The motivation for different amount of correlated dimensions in simulated data is to track the performance of VAEs that contain different dimensions in the latent space, to confirm if the highest performance indeed comes from the model which number of latent dimensions match the number of simulated dimensions. For instance, if the synthetic dataset is driven by four factors, it is interesting to check if the best model is indeed the one that contains four latent variables.

The total amount of normal datasets is 18, since there are six different amounts of driving factors and three levels of correlation. To illustrate, one of these sets can be driven by four dimensions, meaning that each dimension has three features of correlated data. 12 dimensions would result in a dataset that does not contain correlations between features.

The simulated Student-t data sets are generated similarly to the normal data, the difference being the data-generation process. Now, the copula employed is a Student-t copula. Again, this results in 18 Student-t data sets. The goal of using Student-t simulated data is to see if it indeed increases model performance when the latent space distribution matches the original space distributions.

The final simulated sets of data are driven by 4 dimensions. Each of these dimensions is distributed differently, namely correlated normal, correlated Student-

t, correlated Bernoulli and another Student-t dimension, this time correlated non-linearly by a Gumbel copula. Again, they are simulated for the different correlations of 0.25, 0.5 and 0.75, resulting in 3 'mixed' datasets.

3.2 Real data

The analyses of this thesis is performed on real data as well, so see how well autoencoders perform when data acts in more unpredictable ways. For instance, real data can be heteroskedastic, mean varying, as well as other higher order moments that can change throughout time. Autoencoders should be able to compress and reconstruct data relatively well compared to PCA, but it remains to be seen how they perform on more complex datasets.

3.2.1 Returns

In table 1, some information on the features in the daily returns dataset are shown. The data runs from 01/01/2000 until 29/04/2022. As one can see, certain time series do show a large amount of missing values. This is not especially present in any one asset class, but visible in multiple variables in the dataset. Since many of the missing values occur near the beginning of the dataset, the analyses is performed on data recording from 2010 onward. This wields 3214 observations in the used dataset.

This dataset aims to replicate a well diversified portfolio of assets throughout multiple continents and asset classes, including equity composites, fixed income, real estate and commodity instruments. In terms of equities, the dataset contains a time series from European equities, namely the Eurostoxx 50. For US equities it contains series for large cap material stocks, the Russell 2000 for small cap stocks, equities traded on the Nasdaq reflected in the QQQ Trust Series, and the S&P500 composite. Asian equities are represented in this portfolio as well. Korean, Taiwan and Japanese equities are included through MSCI ETFs, while there is also a tracker present for Shanghai equities. Latin American equities are represented through a Brazil tracker. Worldwide trackers for equities are included through both developed and emerging markets trackers, courtesy of MSCI.

The dataset includes some fixed income instruments as well. For government debt, the portfolio includes instruments for inflation linked bonds and US treasury bonds for multiple maturities. Concerning corporate debt, it includes trackers for both investment grade and high yield bonds, provided by iBoxx.

Commodities are represented through a multitude of instruments. Precious and base metals are included in the form of a gold and silver bullion tracker, the Invesco DB Base Metals Fund, and the London Metal Exchange copper tracker.

Energy is represented by the S&P GSCI Energy ETF, and a tracker for Europe Brent oil. Agricultural commodities are included via the Invesco DB Agriculture fund, as well as a tracker for US corn. Commodities also have a further presence through more general trackers, namely the Refinitiv/CoreCommodity CRB index and the CMCI Commodity Composite.

This portfolio is less biased towards real estate, as there is only one instrument present that tracks this asset class. It is represented by the DOW Jones Real Estate index.

Table 13 in appendix A shows summary statistics for the financial daily returns. The returns show a tendency towards mean-zero. Furthermore, most assets show a negative skewness and excess kurtosis, which corresponds with known return characteristics. Equities are generally more volatile than other series, which also fits their known characteristics. The only series that comes above equity in terms of volatility are the energy, oil and base metals series. Fixed income seem the most risk-free investment, especially the treasury bonds. The only series showing positive skewness are the Euro Stoxx 50, inflation linked US goverment bonds, the high yield corporate bonds and the US corn price. Furthermore, all time series exhibit a leptokurtic or fat-tailed distribution, meaning it might be more appropriate for the autoecoders to model the latent space as such, as opposed to a normal distribution.

	Obs	NaN count	Description
ISHARES COREEUR (XET) STOXX50 UCITS ETF EUR	5825	1481	European equities
DOW JONES US REAL ESTATE	5825	239	US real estate
SECT.SPDR TST.SBI INTER INDS.	5825	207	US large cap materials equities
ISHARES INFL.LKD. GOVT BD UCITS ETF EUR A	5825	2894	US inflation linked govt bonds
ISHARES RUSSELL 2000 ETF	5825	309	small cap US equities
ISHARES IBOXX $ HIY.CBD. ETF	5825	2033	High yield corporate bonds
INVESCO QQQ TRUST SERIES 1	5825	207	Nasdaq 100 series
ISHARES 10-20 YEAR TREASURY BOND ETF	5825	1972	US govt bonds
ISHARES CORE FTSE 100 UCITS ETF GBP	5825	787	UK equities
ISHARES 7-10 YR.TRSY.BD.	5825	849	US govt bonds
Gold Bullion LBM $/t oz DELAY	5825	122	Gold Bullion
INVESCO MSCI JAPAN UCITS ETF ACC	5825	2614	Japanese equities
S&P GSCI Energy Total Return	5825	199	Energy
INVESCO DB AGRICULTURE FUND	5825	1968	DBIQ Diversified Agriculture Index
INVESCO DB BASE METALS FUND	5825	1968	Commodity futures
RF/CC CRB ER	5825	1960	Commodities
MSCI EM U$	5825	0	Emerging markets equities
MSCI BRAZIL 25-50 $	5825	0	Brazil mid-large cap equities
MSCI TAIWAN	5825	0	Taiwan equities
MSCI EMU SMALL CAP	5825	0	Developed markets small cap equities
MSCI KOREA 25-50	5825	3	Korea mid-large cap equities
Silver, Handy&Harman (NY) U$/Troy OZ	5825	0	Silver bullion
Crude Oil BFO M1 Europe FOB $/BBl	5825	0	Europe brent oil
Corn US No.2 South Central IL $/BSH	5825	0	US corn
LME-Copper Grade A Cash U$/MT	5825	0	London Metal Exchange copper
ISHARES 1-3 YR.TRSY.BOND	5825	849	US Treasury bond
ISHARES 3-7 YR.TRSY.BOND	5825	1972	US Treasury bond
IBOXX $ LIQUID INVESTMENT GRADE INDEX	5825	2825	Investment grade corporate debt
CMCI Composite TR (USD) - RETURN IND. (OFCL)	5825	237	Various commodity contracts
S&P 500 COMPOSITE	5825	207	US equities
SHANGHAI SE A SHARE	5825	418	Shanghai equities

Table 1: Size, NaN count and description of features of the daily financial returns dataset

3.2.2 Implied volatilities data

To analyze the third research subquestion, a dataset on implied volatilities is used. The data used is At-the-Money (ATM) option data on WTI Crude Oil futures. Shown in table 2 are the count, missing value count and description of the data set features. The categorical features of this dataset are discarded for the analysis, since there is only one unique value, so it does not provide any usable information. These features are the symbol of the future, the option root symbol, and the exhange the option is traded on. The features of the future's expiration, and the option expiration are omitted as well, since the dataset includes numerical time to maturities for both already. The rest of the features of the data set are the adjusted close of the future, the at-the-money implied volatility denoted as $atmVola$, the amount of options determining the implied volatility denoted as $numOptions$, the time to maturity of the future as $futTTM$, and lastly the time to maturity of the option denoted as $opTTM$.

The ATM implied volatility is determined by an average of implied volatilities of 2 puts and 2 calls that are the closest to the ATM point. As one can see, only the feature containing the implied volatility exhibits missing values. Early exploratory analysis shows some correlation between the number of options available and missing value observations. The average of available options

for the missing volatility values is 8.39, while the average available options for the non-missing volatility values is equal to 186.87.

Shown in table 14 are summary statistics from the implied volatility data set. There is abundant diversity in the distribution of the features. The reason for this is that most variables have different levels. It is important to note that all variables are strictly positive. For this reason, it is not appropriate to normalise the data by subtracting the mean and dividing by the standard deviation. Instead, every variable is divided by its maximum, rendering a dataset which is bounded between zero and one. This is done for stability purposes in the model optimisation process. More information on the optimisation can be found in section 4.1.

	Obs	NaN count	Description
symbol	141423	0	CL for crude oil (WTI)
option root symbol	141423	0	CL for crude oil (WTI)
future symbol	141423	0	Future symbol as on exchange
exchange	141423	0	Exchange where the option is traded – NYMEX
future expiration	141423	0	Expiry date of the futures contract
adjusted close	141423	0	Close of the underlying future
expiration	141423	0	Expiration date of the option
atmVola	141423	8910	ATM volatility of the option chain
numOptions	141423	0	Number of options available for this skew
futTTM	141423	0	Time to maturity of the future in years
opTTM	141423	0	Time to maturity of the option in years

Table 2: Size, NaN count and description of features of the ATM implied volatility dataset

4 Methodology

In this section, the methodology behind the applications is discussed. First, it is analyzed how latent space distributional assumptions affect the reconstruction error. This is to see how well the algorithm can transform the data to different distributions. Second, the autoencoder is employed to obtain the Value-at-Risk of financial returns of a portfolio. This is to test how the autoencoder estimates and reconstructs the tails, or extreme values, of the return distribution. If the VaRs estimated are correct and sensible, it can be concluded that the model can learn the distribution driving the return generation process. Lastly, the reconstruction of partially incomplete implied volatility curves are discussed. This is done to investigate how well an autoencoder can learn the dynamics of these curves, and subsequently correct the curves that are partially missing.

4.1 On the optimisation of a VAE

Variational autoencoders are complex deep learning models, and require some background knowledge on how they are optimised. Machine learning is notorious for optimisation inconsistencies, even as a result of slight changes in the training routine. For this reason, this subsection aims to give more information on the optimisation techniques.

As mentioned in section 1, VAEs are fitted according to two criteria, the reconstruction error, and a measure to regulate the latent space. The reconstruction error is defined as the following:

$$RE = \frac{1}{N}\sum_{i=1}^{N}(X_i - X_i')^2 \tag{1}$$

X denotes the original data, while X' denotes the reconstructed data. The measure that is most often used to regulate the latent space is the Kullback-Leibler (KL) divergence between two distributions P and Q:

$$KL(P|Q) = \int_{-\infty}^{\infty} p(x)\left(log\left(\frac{p(x)}{q(x)}\right)\right)dx \tag{2}$$

This statistic measures the distributional distance of two sets of data. When comparing normal distributions, the KL divergence has an analytical solution in the form of:

$$KL(P|Q) = \log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2(\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2} \tag{3}$$

In this thesis, the model is set up as a β-VAE. The final objective function becomes:

$$\arg\min_{\phi,\psi} RE + \beta KL \tag{4}$$

Where ϕ are the parameters of the neural network handling the encoding process, and ψ are the parameters used in the neural network handling the decoding process. The parameters of the model are determined by an iterative process known as stochastic gradient descent. The optimizer determines in which direction the parameters should change to minimize the objective function by approximating the derivative of the objective function with respect to the parameters. After this, a step in the parameters is taken, and the process is repeated for a set amount of times, also known as epochs. In stochastic gradient descent, a random set of observations are taken to determine this derivative, instead of the entire dataset. This is done for two reasons. The first reason is simply computational speed. When determining a gradient using a handful of data points, it will be quicker than determining a gradient for the entire dataset. The second reason is that the model is more likely to find a global minimum, rather than a local minimum. When the step-sizes of the parameters are varying due to the stochastic nature of the determined gradient, it is more likely to be able to jump out of a local minimum. However, even with the reduced chance of getting stuck in a local minimum, there is still some variance in the models convergence. Overall, the model does find similar parameter estimates for every time it is optimised, but sometimes fails to converge due to the stochastic nature of the optimisation process.

Another relevant topic when discussing the training process of deep learning models is the tuning of hyperparameters. Hyperparameters are parameters that control the learning process, as opposed to the parameters that the model derives from the training process that are used to transform the data. The hyperparameters relevant for this thesis are the optimizer itself, the learning rate, the weight-decay, the number of layers and nodes in the layers, and lastly the activation function used in the nodes.

The optimization algorithm used is AdamW, a version of Adam with improved weight decay implementation. The learning rate determines the step-size when updating the model parameters for each iteration in the training process. The weight-decay is an added term in the objective function that regularizes the calculated error. The calculated gradient is therefore also weighed with this term, and leads to more stable steps in model parameters in the model optimisation. Essentially, the learning rate and weight decay have the same function achieved in different ways, and work together to produce a training routine that is stable in its convergence and resistant to overfitting. The amount of layers in the neural networks of the model present a trade-off in the model optimization. An increase in layers makes the model more complex and capable of tackling more complex issues, but leaves it more prone to overfitting and convergence issues. The number of nodes per layer is derived from the size of the data set. For data with D variables, the encoding neural network has D nodes for the input

layer, d nodes in the output layer that results in a latent space with d variables, and $(D+d)/2$ nodes in the layers in between. The decoding neural network has d nodes in the input layer, D nodes in the output layer and $(D+d)/2$ nodes in the layers in between. For autoencoders, $d < D$, resulting in the compression of data, also known as the information bottleneck. Lastly, the activation function chosen in the neural networks is a Leaky Rectified Linear Unit, or Leaky ReLU. This function is chosen as other activation functions lead to poorly regulated latent spaces, or poorer model convergence. For each analysis discussed hereafter in this chapter, the hyperparameters are determined via a grid search, and presented shortly in the results.

4.2 Testing distributional assumptions

As mentioned in section 1.2, we first test how distributional assumptions affect the autoencoder reconstruction performance. This does present us with an immediate issue. When comparing non-normal distributions, the KL divergence is calculated through an integral, which is computationally intensive. Because of this computational load, the optimisation is not practical, especially since data sets for machine learning models tend to be large. For this reason, other methods of regulating the latent space are considered. An example is the Kolmogorov-Smirnov test. Consider a sample of data $x_1, ..., x_n$ with distribution function F. $y_i = F(x_i|x_{i-1}, x_{i-2}, ...)$, or the conditional empirical distribution function. G_n represents the empirical distribution function of y_i.

$$KS = \max_{j=1,2,...} \sup_{y^j} |G_n(y^j) - y_1^j ... y_p^j| \ , \tag{5}$$

The KS in a multivariate setting needs to be calculated for every one of j permutations of the data set, and was therefore discarded due to unattainable computational power in higher dimensions. Maximum likelihood was considered as well, but discarded due to loss of variance in the latent space. Ultimately, a method of moments was chosen in line with the paper from Beaulac (2021), which stipulates that a multivariate approach to moment calculation can be used as a way to assess a goodness of fit of a distribution. This method is computationally quicker than numerically integrating the KL divergence, and is better suitable as a metric due to its symmetry, a property that the KL divergence does not exhibit. The chosen metric takes the difference between an estimated vector or matrix and a target, and takes the norm of the resulting vector or matrix. This is done for the first four standardized moments. The norms are calculated as:

$$\begin{aligned} |v|_q &= (\sum_i |v_i|^q)^{(1/q)} \\ |M|_q &= (\sum_{ij} |M_{ij}|^q)^{(1/q)} \end{aligned} \tag{6}$$

For vector v and matrix M respectively. In the case of the first, third and fourth standardized moment, the norm of the vector is taken. For the second moment, the matrix norm is taken, as we want to take covariances in account when regulating the latent space. For the normal distribution, mathematical formulation for the metric as a weighted average of individual moment scores is shown below:

$$\begin{aligned} MM &= \sum_{i}^{4} w_i S_i \\ S_1 &= |v_{means}|_q \\ S_2 &= |M_{covariance} - \mathbf{I}|_q \\ S_3 &= |v_{skewness}|_q \\ S_4 &= |v_{kurtosis} - \mathbf{3}_n|_q \end{aligned} \tag{7}$$

For other distributions, the target moments change. For instance, kurtosis will become larger when we want to regulate the latent space as a fat-tailed student t distribution. Furthermore, the weights w_i are determined by a grid search that optimizes reconstruction error. For the analysis itself, the reconstruction errors of different latent space regulations are obtained and compared. It could be the case that when regulating the latent space to a distribution that more closely resembles the original data generation process, less information is lost in the encoding process, resulting in a more accurate reconstruction. The distributions tested are a Gaussian distribution and a Student-t distribution. These models are compared for the following applications as well. The reconstruction errors are analyzed for the simulated and return data. In the case of returns, the performance of the VAEs are also compared to PCA, to obtain evidence whether or not these algorithms can outperform more established methods. Out-of-sample performance is analyzed as well, on mixed simulated data with $\rho = 0.5$. The reason for out-of-sample performance analysis on simulated data is that we can guarantee homogeneity in the sample, so large increases in error is an indication of overfitting. A last thing to note, is that the reconstruction errors are obtained by taking the average of ten separate optimisations. Due to the stochastic nature of the model optimisation discussed in section 4.1, letting the model optimize once might not give a full picture how the model performs. The reason for taking an average error as opposed to the best error is that we are also interested in how well the model converges on average. If one model does not converge regularly, but produces one run that outperforms all other models, it would be inappropriate to perpetrate this model as the best choice.

4.3 Multivariate GARCH and VaR for portfolios

The aim of this application is to establish how well the model can learn the distribution of financial returns, while simultaneously proposing a solution to issues

of multivariate GARCH volatility modelling. Multivariate GARCH models are a natural extension of the univariate case introduced by Bollerslev (1986). This class of models are an observation driven way to model time varying variances. With a multivariate approach, one can also model time varying dependencies between assets, resulting in a more nuanced view of variance throughout the sample. However, problems arise when dimensions get too high, since the amount of parameters for the covariance matrix increase exponentially. It is possible to make simplifying assumptions about the dependency structures in data to combat this issue, such as the CCC-GARCH model introduced by Bollerslev (1990), but this is not always appropriate.

Another approach is to reduce the dimensions of the data first before obtaining volatility estimates. When the volatilities are obtained, one can translate them back to the dimensions of the original data. An example of this is the principle component GARCH (PC-GARCH) by Geng (2007). The approach presented is to compress data using principle component analysis. Then, the volatilities are estimated using a series of univariate GARCH(1,1) models, also known as Orthogonal GARCH (O-GARCH). This is facilitated by the fact that principle components are orthogonal by design. Then, the diagonal covariance matrices are transformed to the original dimensions by the weights used to obtain the principle components. The intuition behind this is to understand times series variance through GARCH models, and cross-sectional dependencies through PCA. Shown in figure 2 is a flow chart explaining the process.

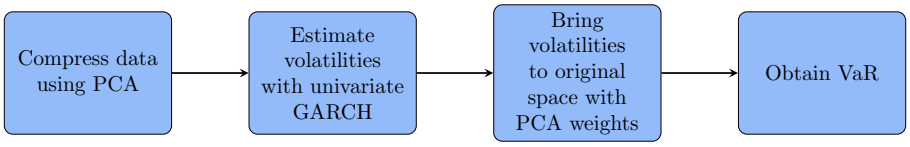

Figure 2: Flow chart showing the methodology of PCA O-GARCH

However, there are some limitations to this approach. First, the use of PCA limits covariances to be a product of linear transformations. This might not be appropriate for financial assets, as it is known that correlations are non-linear. The second problem with this approach arises with the fact that this approach does not take time-varying dependencies into account, as the weights that construct the principle components do not change throughout time. Conversely, there exist more nuanced ways of reducing the dimensionality of data. VAEs can take non-linear dependency structures into account, since neural networks can transform data in a more complicated manner than the linear transformation associated with PCA.

Adapting the PCA-GARCH methodology to VAEs leads to a similar process, but with some implications. We can transform the original data to a low dimensional space. We employ the same O-GARCH as PCA, since the latent variables are regulated to be uncorrelated. Recall that with PCA-GARCH it is trivial to transform the obtained variance estimates to the original space. This is not trivial with VAEs, since the transformation of original data to latent space is highly non-linear. Instead, we make use of the continuity and regularity properties of the VAEs. Continuity refers to the property that small changes in the latent space lead to small changes in the original space when they are decoded. Regularity refers to the property that wherever a point is in the latent space, the decoding of this location will result in a sensible outcome in the original space. Current literature has not yet proposed a method of testing these properties directly, but they are essential as the presence of these properties ensure that the autoencoder has properly learned the distribution of the original data, and can therefore effectively mimic this distribution by simulation. To achieve usable VaR forecasts in the original space, we simulate data in the latent space based on the time varying variance estimates, and subsequently decode the heteroskedastic data. If the continuity and regularity properties hold, it means that we obtain accurate return representations that correspond to different volatility regimes, and can therefore predict risk.

The decoded data is aggregated into an equally weighted portfolio return for stability purposes. Next, one-day-ahead VaR estimates are obtained by taking a quantile from the transformed simulations, and back tested on exceedance coverage and independence. The coverage is tested with a binomial test, and independence is tested with Christoffersen's test. As with PCA, the latent variables in the VAE approach are regulated to be orthogonal. The motivation for this is that by regulating the latent variables as orthogonal, it means that there is no limit of how many latent variables there are while still being able to employ GARCH. If the latent variables are not orthogonal, there would be a need to model their covariances, using multivariate GARCH. If for some reason the dataset can not be properly compressed to a small number of latent variables, the same issues this methodology was supposed to solve are encountered again.

Shown in figure 3 is a flow chart explaining this methodology. Comparable to PCA O-GARCH, GARCH models the time series volatilities while the encoder and decoder learn the cross-sectional dependencies. On a final note, we assume that the amount of latent dimensions and principle components are fixed at three, which gives a good trade-off between data compression and reconstruction accuracy.

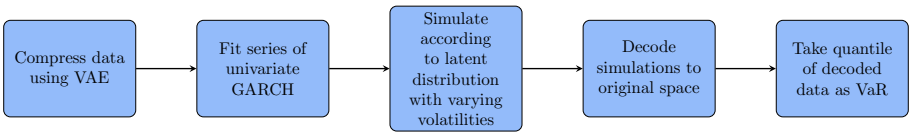

Figure 3: Flow chart showing the methodology of VAE O-GARCH

4.4 Correcting At-the-Money implied volatility (IV) curves with incomplete and noisy data

As mentioned in section 1, autoencoders can capture dynamics of curves and surfaces, to subsequently use them to correct curves that are noisy or sparse. In the data used, the IV of options on WTI crude oil futures is given as well as the amount of options in the skew. Naturally, IV data that is derived from more options is less noisy than data obtained from maturities where there is no liquidity. Hence, we can take this into account by weighting the reconstruction errors by liquidity. Essentially, the model is punished strongly for inaccurate reconstructions where data is backed by more liquidity. In order to take this into account, the reconstruction error RE is changed to be a weighted average:

$$RE = \frac{1}{N} \sum_{i=1}^{N} w_i (X_i - X_i')^2 \ , \tag{8}$$

where the weight w_i of a data point is determined by:

$$w_i = \frac{\text{numOptions}_i}{max(\text{numOptions})} \tag{9}$$

Furthermore, since the main objective is to construct curves, only the weighted reconstruction error of the implied volatility is calculated. The missing values of the implied volatility dataset are imputed linearly, and then passed through the fitted autoencoder. Furthermore, the amount of options backing observations with missing values are set to zero, so the autoencoder is not enforced in any way in the reconstruction at these points. Ideally, the model learns how to fill in the values from other observations, and changes the linear imputation to a more appropriate shape. The reconstructed curves are then presented and discussed in section 5.3.

5 Results

This section discusses the results obtained from the analyses for the different research questions. First, the reconstruction error analysis for different latent space regulations is tested, to examine how well the autoencoder can transform different distributions. Second, a method to obtain VaR estimates using autoencoders is shown, to test the accuracy of financial return distribution modelling. Lastly, incomplete implied volatility curves are corrected using autoencoders, to analyse if the models can model the characteristics of these models.

5.1 Testing distributional assumptions

As discussed in section 4.2, the model performance is tested by changing the latent space regularisation to different distributions. This is done for different levels of correlation, amount of latent space dimensions, and amount of dimensions driving the data generation process. When optimizing the fit of the autoencoders, it is necessary to tune the hyperparameters that impact the training routine. Shown in table 3 are the hyperparameters used. The models performed best when the weight decay was really small, so it seems that scaling the errors to be smaller led to more stability in the training process. The model is not too complex either, with two hidden layers between the input and output layer for both the encoder and decoder. The model convergence times were also relatively quick, training in 1000 epochs. Paired with the moment matching latent space regulation mentioned in section 4.2, the models trained in about 20 seconds.

Hyperparameter	**Value**
Learning rate	0.01
Weight decay	0.001
Nr of hidden layers	2
Epochs	1000

Table 3: Tuned hyperparameters for the reconstruction error analysis

Shown below and in appendix B are the mean reconstruction error tables for the different types of data and distributional assumptions in the latent space of the autoencoders. Since the optimisation process is stochastic, the results presented are obtained as an average of ten runs. The latent dimension refers to the amount of variables in the latent space, while the simulated dimensions refers to the amount of factors driving the simulated data. Thus, when the latent dimension becomes lower, the compression of the data becomes greater. When the amount of simulated dimensions decreases, the amount of information in the data becomes less diverse. Moreover, the results are presented for different correlations as well, denoted in the table as ρ. Note that the errors are

shown for standardized data, so the results are comparable across datasets.

When considering the reconstruction errors presented, it is possible to see a positive link between the amount of dimensions simulated in the data and the reconstruction error. This is expected since the more diverse information is, the harder it is to compress it. Furthermore, increasing the amount of variables in the latent space results in a more accurate reconstruction, since the model is allowed to store more information in the latent space. It is also possible to see that the reconstruction error decreases as the correlation within factors in the data increases. This points to the model being able to take advantage of strong structures in the data for more accurate compression. These observations are present across all datasets and model types.

Tables 15 and 16 show the Gaussian and Student-t autoencoder respectively on simulated Gaussian data. It shows the same patterns as discussed in the paragraph above. The reconstruction error is at its lowest when there is strong correlation present in the data set, the driven by one singular factor, and the model is allowed to store a lot of information in the latent space. The Gaussian VAE performs somewhat better than the Student-t VAE. This is driven by the fact that the Student-t VAE tends to converge slightly less often. They do show similar errors, but due to weaker convergence the Gaussian VAE is considered the better option in modeling Gaussian data.

Shown in tables 4 and 5 are the mean reconstruction error of the Gaussian and Student-t autoencoder respectively on simulated Student-t data. The patterns concerning correlation, amount of driving factors in the data and amount of latent variables are the same as with the models fitted on Gaussian data. The error again decreases with increasing correlation in the dataset, decreasing the amount of driving factors and increasing the information stored in the latent space. When comparing the Gaussian VAE to the Student-t VAE, similar conclusions can be drawn as with the Gaussian simulated data. The Gaussian VAE performs better as a result of its consistency in convergence.

$\rho = 0.25$ latent dimensions	1	2	3	4	6	12
simulated dimensions						
1	0.728	0.711	0.634	0.571	0.443	0.219
2	0.859	0.729	0.657	0.589	0.446	0.143
3	0.899	0.784	0.66	0.574	0.453	0.13
4	0.917	0.825	0.704	0.612	0.457	0.122
6	0.931	0.853	0.758	0.663	0.481	0.135
12	0.945	0.876	0.814	0.734	0.569	0.141

$\rho = 0.5$ latent dimensions	1	2	3	4	6	12
simulated dimensions						
1	0.479	0.473	0.442	0.399	0.332	0.218
2	0.701	0.519	0.488	0.408	0.353	0.18
3	0.842	0.68	0.513	0.435	0.314	0.171
4	0.865	0.732	0.598	0.439	0.339	0.132
6	0.915	0.808	0.704	0.57	0.34	0.13
12	0.945	0.88	0.812	0.728	0.572	0.154

$\rho = 0.75$ latent dimensions	1	2	3	4	6	12
simulated dimensions						
1	0.259	0.249	0.237	0.229	0.217	0.151
2	0.381	0.313	0.273	0.227	0.209	0.142
3	0.683	0.504	0.321	0.255	0.192	0.133
4	0.828	0.672	0.491	0.272	0.196	0.125
6	0.895	0.775	0.632	0.501	0.225	0.116
12	0.946	0.889	0.809	0.72	0.571	0.132

Table 4: Mean reconstruction error for Gaussian VAE on Student-t simulated data for different correlations, amount of simulated dimensions and latent dimensions

$\rho = 0.25$ latent dimensions	1	2	3	4	6	12
simulated dimensions						
1	0.757	0.704	0.647	0.576	0.44	0.222
2	0.858	0.724	0.644	0.592	0.434	0.144
3	0.903	0.786	0.663	0.569	0.434	0.123
4	0.917	0.814	0.729	0.59	0.457	0.134
6	0.929	0.845	0.759	0.68	0.463	0.11
12	0.945	0.883	0.821	0.741	0.567	0.157

$\rho = 0.5$ latent dimensions	1	2	3	4	6	12
simulated dimensions						
1	0.508	0.476	0.442	0.404	0.361	0.217
2	0.686	0.528	0.48	0.41	0.333	0.176
3	0.825	0.657	0.523	0.439	0.323	0.145
4	0.88	0.755	0.586	0.444	0.327	0.13
6	0.912	0.815	0.699	0.588	0.347	0.113
12	0.944	0.883	0.802	0.732	0.559	0.134

$\rho = 0.75$ latent dimensions	1	2	3	4	6	12
simulated dimensions						
1	0.26	0.251	0.244	0.225	0.221	0.163
2	0.479	0.336	0.279	0.225	0.203	0.141
3	0.677	0.498	0.31	0.264	0.191	0.132
4	0.803	0.645	0.472	0.284	0.193	0.128
6	0.886	0.768	0.652	0.496	0.223	0.11
12	0.944	0.874	0.81	0.735	0.561	0.142

Table 5: Mean reconstruction error for Student-t VAE on Student-t simulated data for different correlations, amount of simulated dimensions and latent dimensions

Tables 6 shows the performance of Gaussian and Student-t autoencoders on mixed simulated data. Similar patterns of reconstruction errors can again be seen in this data type. Strong dependencies are used to better reconstruct the data after compression. A lesser degree of compression again means better reconstruction accuracy. The Gaussian and Student-t models behave similarly. Here, they have a similar rate of good versus bad fits, which result in highly similar reconstruction errors. The models employed on this data type do outperform the models employed on other data types, but only when the latent space is large.

Gaussian VAE latent dimensions	1	2	3	4	6	12
$\rho = 0.25$	0.864	0.679	0.579	0.451	0.317	0.054
$\rho = 0.5$	0.755	0.581	0.482	0.362	0.235	0.057
$\rho = 0.75$	0.733	0.58	0.396	0.256	0.142	0.067
Student-t VAE latent dimensions	**1**	**2**	**3**	**4**	**6**	**12**
$\rho = 0.25$	0.81	0.678	0.553	0.479	0.314	0.055
$\rho = 0.50$	0.763	0.614	0.447	0.34	0.223	0.063
$\rho = 0.75$	0.777	0.54	0.346	0.241	0.144	0.061

Table 6: Mean reconstruction error for Gaussian VAE on mixed simulated data for different correlations and latent dimensions

Shown in table 7 are the out-of-sample reconstruction errors for the two types of autoencoders. In contrast to the in-sample analyses, the Gaussian evidently performs better than the Student-t VAE. This likely has to do with issues in convergence in the Student-t autoencoder, which was also the case in the in-sample analyses. Here, it is amplified due to the increased complexity of the task. Instead of purely reconstructing data, the model needs to learn the distributional characteristics to judge how data behaves, even when the model has not seen this data before. In a stable latent space, this seems to work relatively well. In a more volatile latent space produced by the Student-t autoencoder, this seems a more challenging task.

assumed dimensions	1	2	3	4	6	12
Gaussian VAE	1.0000	0.6789	0.4374	0.277	0.206	0.1470
Student-t VAE	1.0000	0.741	0.5060	0.4112	0.297	0.224

Table 7: Out-of-sample mean reconstruction error for Gaussian and student-t VAE on mixed simulated data for different assumed dimensions

Table 8 shows the performance of the two autoencoders on return data, this time compared to PCA. Even on average, the Gaussian VAE can outperform PCA for all number of latent dimensions. Due to a lesser rate of convergence, the Student-t autoencoder is not able to reconstruct the data in the same degree of accuracy as PCA and the Gaussian VAE. For the rest, the same principles governing the reconstruction errors are true, namely that a larger latent space corresponds to a smaller reconstruction error.

assumed dimensions	1	2	3	4	6
PCA	0.6993	0.600	0.5189	0.4696	0.4228
Gaussian VAE	0.696	0.556	0.494	0.447	0.402
Student t VAE	0.753	0.625	0.535	0.488	0.437

Table 8: Mean reconstruction error for different compression methods on return data for different latent dimensions

Overall, it can be said that transforming distributions is no challenge for autoencoders, as the chosen latent space distributions do not seem to affect the level of the reconstruction errors. However, strong correlation, information diversity and the amount of information stored in the latent space does affect how the model performs. Furthermore, it seems that a latent space with comparably less gaps is more stable in its convergence, which potentially ties in with the continuity and regularity properties of the models.

5.2 Multivariate GARCH and VaR for portfolios

As discussed in section 4.3, the VaR for a large portfolio is obtained by reducing the dimensionality of the data, fitting a series of univariate GARCH(1,1) models, and subsequently transforming the estimated volatilities to the original space. With PCA, this is done through a direct linear transformation facilitated by the principle component weights. The VAE approach requires an indirect method, namely through decoding heteroskedastic simulations to obtain a representation of the return distribution. Shown in table 9 are the hyperparameters for the model as a result of a grid-search. The learning rate is now increased, the model is less complex, and it is trained for a shorter amount of time. This makes the

fit of the model more variant. The reason for this is that the model does produce correct VaR estimates, but the correct VaR does not correlate with lower reconstruction errors. Likely, the model focuses too much on reconstructing the data, rather that learning its distribution.

Hyperparameter	**Value**
Learning rate	0.02
Weight decay	0.001
Nr of hidden layers	1
Epochs	500

Table 9: Tuned hyperparameters for the GARCH analysis

Shown in table 10 are the VaR exceedance ratios, their p-values, and the p-values for the Christoffersen's independence test for O-GARCH using PCA. q denotes the chosen Value-at-Risk quantile. Unsurprisingly, PCA suffers from variance loss in its reconstruction. Due to the way principle components are obtained, excluding a set of principle components will by definition reduce variance in the reconstructed data. This is reflected in the VaR exceedance ratios. PCA underestimates risk through this mechanism, and is therefore not a good suitor to properly assess risk when data dimensionality reduction is also a goal. The underestimation of risk is reflected in the VaR exceedance ratios, which are all higher than the chosen quantile. This means that the model is systematically underestimating risk as a result of this variance loss. This problem can be avoided when all the principle components are included. The p-values from the independence tests show that the exceedances are not independent, although it is known that the power of the Christoffersen's independence test is not high.

Distribution	**q**		
Normal	0.10	ratio	0.119
		p-value	0.000
		p-val independence	0.000
	0.05	ratio	0.074
		p-value	0.000
		p-val independence	0.000
	0.01	ratio	0.033
		p-value	0.328
		p-val independence	0.000
Student-t	0.10	ratio	0.109
		p-value	0.094
		p-val independence	0.000
	0.05	ratio	0.068
		p-value	0.000
		p-val independence	0.000
	0.01	ratio	0.029
		p-value	0.000
		p-val independence	0.000

Table 10: Obtained Value-at-Risk exceendance ratio for quantile q, its p-value, and the Christoffersen's indepedendence test p-value using PCA O-GARCH.

Table 11 shows VaR exceedance ratios, their p-values, and the p-values for the Christoffersen's independence test for O-GARCH using VAEs. One can see that for the normal distribution, the VAE O-GARCH produces correct exceedance ratios for all different quantiles. However, the Student-t VAE does overestimate risk. This likely has to do with the increase in amount of extreme values present in the simulations. Likely, the decoder does not properly handle these observations, resulting in an inflated risk estimate. This means that the regularity property is broken somewhat, probably as a result of the presence of gaps in the latent space courtesy of an increase in extreme values. The same issues with VaR exceedance independency with PCA arise in the VAE approach. The p-values of the independence test show similar values as the PCA approach, although their value is a bit higher when looking at the unrounded numbers. This means that the exceedances are slightly less dependent, but still not independent.

Distribution	**q**		
Normal	0.10	ratio	0.101
		p-value	0.746
		p-val independence	0.000
	0.05	ratio	0.050
		p-value	0.840
		p-val independence	0.000
	0.01	ratio	0.009
		p-value	0.929
		p-val independence	0.000
Student-t	0.10	ratio	0.107
		p-value	0.186
		p-val independence	0.000
	0.05	ratio	0.034
		p-value	0.000
		p-val independence	0.000
	0.01	ratio	0.003
		p-value	0.000
		p-val independence	0.000

Table 11: Obtained Value-at-Risk exceendance ratio for quantile q, its p-value, and the Christoffersen's indepedendence test p-value using VAE O-GARCH.

Figure 4 shows the normal distribution VaR estimates for both the PCA and VAE approach. As one can see, the VAE approach adjusts itself more to new observations, as the line of VaR estimates moves more. One conclusion that can be drawn from this is that the VAE models the impact of extreme movements better. This likely has to do with the fact that the VAE is capable of modelling non-linear dependencies between assets, an effect that is increasingly important as volatility regimes become more extreme. As correlations increase in extreme return quantiles, it is necessary to model this change in dependency. It seems that the VAE is better capable of doing this than PCA, which uses fixed transformation weights for the entire sample, essentially having fixed correlation structure. Another conclusion that can be drawn from the varying VaR estimate is that the continuity and regularity properties hold in the VAE, since quantiles in the latent space distribution tend to correspond to similar quantiles in the original space. This is exceedingly important as the continuity and regularity properties of the VAE facilite this approach in the first place.

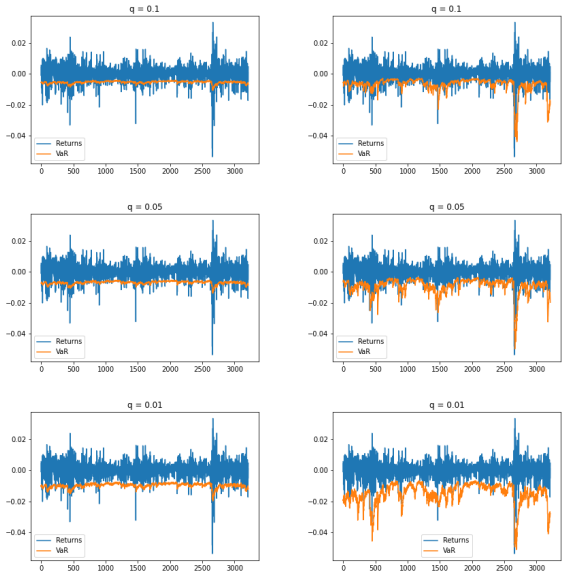

Figure 4: Gaussian VaR estimates for different quantiles. PCA O-GARCH is shown on the left hand side, and VAE O-GARCH is shown on the right hand side.

Figure 6 in appendix C shows similar plots, this time employing the Student-t distribution. This time, PCA seems to produce more variant VaR estimates. The conclusion drawn from this is that the continuity and regularity properties get broken. Again, the presence of kurtosis is likely the culprit. Most likely, the presence of more extreme outliers cause gaps in the latent space, making is harder for the model to establish proper return representations from simulations that land in the gaps of the latent space the model is trained on. More research on this topic is needed to confirm this, however. Concretely, one can test for correlations in the kurtosis in the latent space and the goodness of fit of decoded simulations. If there are less gaps in the latent space, it might mean that the obtained return distributions from simulations are better.

All in all, the methodology of obtaining VaR with autoencoders is promising. The models were able to properly obtain the correct VaR quantile, even when the continuity and regularity properties were not quite present, as was the case with the Student-t autoencoders. For this reason the Student-t VAEs need improving such that the continuity and regularity properties are improved. This is difficult, since a lower reconstruction error does not necessarily result is better distribution learning. Instead of simulating in the latent space, it could be that bootstrapping standardized latent space observations, similar to filtered

historical simulation, will lead to better return representations. It could be that these bootstrapped observations do not end up in the gaps of the latent space, and will lead to more accurate VaR estimates. More research is needed however. This thesis also lacks proper out-of-sample analysis for this methodology, so it would also be interesting to see how well the models perform out-of-sample.

5.3 Implied Volatility curve correction

The methodology behind the IV curve correction is discussed in section 4.4. In short, the autoencoder is trained to reconstruct IV curves from the data set. There is more emphasis on the observations that are driven by higher liquidity in the data. Furthermore, the model receives full autonomy over filling in the observations that exhibit a missing value in their implied volatility, as the reconstruction error of these observations is given no weight in the objective function over which the model is optimised. Shown in table 12 are the obtained hyperparameters obtained from the grid search. This time, the model benefits from a more stable optimisation routine. This likely has to do with the fact that this task is less complex. Only one variable is considered when reconstructing. Furthermore, a decrease in reconstruction error correlates with smooth curves, so the optimisation process is made more accurate by small consistent steps over a longer time.

Hyperparameter	**Value**
Learning rate	0.001
Weight decay	0.01
Nr of hidden layers	2
Epochs	2000

Table 12: Tuned hyperparameters for the GARCH analysis

Figure 5 shows corrected implied volatility curves for the Gaussian VAE. In appendix D shows the same curves, this time corrected using the Student-t VAE. The left hand side shows the curve as present in the data set, as well as the curves for the day before and after the given observation. The right hand side shows the reconstructed data after forwarding it through the trained autoencoder. It is possible to see that the linearly imputed values are changed to a more asymptotic shape, reflecting shapes regularly seen in volatility curves. This implies that the model indeed learns the characteristics driving particular curve shapes, and is fit to use in correcting curves that are either noisy or incomplete. The Gaussian and Student-t VAEs again perform similarly in this situation, both in terms of curve shape and in reconstruction error. In this case, it seems that gaps in the latent space do not affect model performance as much. This could have to do with the fact that this task is somewhat less complex, so slightly less optimised latent spaces might not affect the outcome of the curve

correction so much. One thing to note is that only contemporaneous data was used. Future research might include lagged data, to see how autoencoders can use that information to their advantage. This blurs the line between supervised and unsupervised learning however, so this analysis was not included in this thesis.

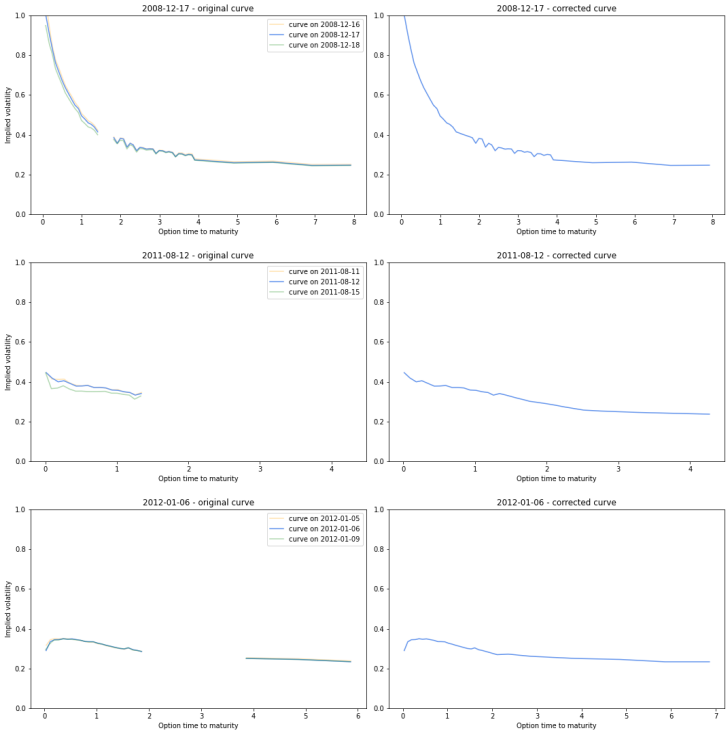

Figure 5: Implied volatility curves corrected using a Gaussian VAE

6 Conclusions and further research

6.1 Conclusion

Autoencoders are still relatively new models in the financial literature. This thesis aims to establish the usability of these models in a financial setting. This involved testing their ability to transform distributions, to compress data, and to learn distributions. This was tested on synthetic and real data in multiple settings and methodologies.

Considering the reconstruction errors as a product of varying distributional assumptions in the latent space regulation, it becomes apparent that regulating the latent space differently does not yield significantly different results. Modelling the latent space as having less outliers does seem to aid in regular convergence, however. There are also some general conclusions to be drawn from the results of this analysis. Unsurprisingly, increasing the size of the latent space results in better reconstruction performance. The model is simply allowed to store more information in the latent space, and is therefore able to better reconstruct the original data. This is the case for both in- and out-of-sample. The VAEs are also able to take advantage of strong dependency structures present in the data, as stronger correlations lead to lower reconstruction errors. Notably, we see that VAEs can outperform linear methods, like PCA, on financial returns. This likely has to do with the non-linear dependency characteristics that returns generally exhibit. Taking these results into consideration, we can conclude that the models are well capable of transforming distributions.

Shown in section 5.2, we see that VAEs can correctly learn the distributional characteristics of return data, and relate it to a reduced form. Since the correct VaR ratios were obtained by taking a quantile of decoded simulations, we can conclude that the model does indeed learn the return distribution, and can accurately generate this data as well. The fact that the VAEs can even learn what the tails of the distribution are when dependencies between assets are known to be non-linear shows potential for risk modelling using this approach. Furthermore, we see that the VAEs outperform linear methods. Since PCA is incapable of modelling non-linear dependencies, this implies that the VAEs are capable of modelling these dependencies. It is also possible to see that the VaR estimates adjusts itself over time, corresponding so different volatility regimes. This implies a link between the latent and original space, which can be useful for risk factor identification. More research needs to be done however, since the convergence of the models is an issue, and optimising the model on a lower reconstruction error does not necessarily imply a better VaR ratio.

VAEs are also shown to be able to correct incomplete IV curves. This is a more known application, but in this case the optimisation function has been weighted with liquidity. This approach puts more emphasis on the data driven by more liquid markets, and allows the model more freedom in reconstructing

noisy data that is a product of less liquid markets. Furthermore, the model was allowed full autonomy in reconstructing missing values by weighing these errors as zero. Using this approach, we were able to reconstruct the curves, even when a large part was missing. Note that while the data is first imputed linearly, the model learns the dynamics of the curves from the curves on different days, and corrects the linear imputation to a more appropriate shape. This shows that it is possible for the model to learn the characteristics of a implied volatility curve, and succesfully project that knowledge on situations where the curve is missing. Such applications could be especially useful for data pre-processing, since it is a sophisticated and nuanced method of missing value value imputation.

All in all, autoencoders seem to be able to transform and learn distribution effectively. This is demonstrated in the similarities of the reconstruction errors for different latent space regulations, the ability to properly assess return quantiles in its distributional tails, and the ability to properly reconstruct partially missing implied volatility curves.

6.2 Limitations and future research

Even with the initial successes in distribution transformation and modelling, there are still issues with this implementation.

One issue with this implementation is that model convergence is not guaranteed. For this reason, the methodology has drawbacks if it is to be used for material decisions or risk management. Ideally, further research can increase model training stability, and guarantee proper optimization. We also see that with gaps in the latent space, the performance of the model becomes more variant across separate optimizations. More research is needed for this as well, to confirm where this problem comes from.

Although the VaR can be modelled successfully using autoencoders, the model performance is impacted greatly by the latent space optimization. Making the latent space distribution more kurtose had a negative effect on the models ability to learn the distribution of the original data. This problem could potentially be aided if the approach did not use simulation, but rather used the bootstrapping of standardized innovations in the latent space, likening the approach more to filtered historical simulation. Furthermore, it seems that the continuity and regularity properties were broken by employing a fat-tailed latent space distribution, so more research is recommended on methods to ensure these properties in the model training process. The reparameterization trick proposed by Kingma & Welling (2013) does improve these properties, but was not included in the analysis of this thesis due to lack of time.

The liquidity-weighted curve correction is limited as well. For instance, no-arbitrage conditions can be added to the optimisation process, resulting in more arbitrage free curves. This has been done in financial literature, but it would be

interesting to see how this condition would interact with the liquidity weighing of the reconstruction errors. Furthermore, this framework can be applied to volatility surfaces as well, or different option Greeks, to build a stronger case for weighing the reconstruction errors with liquidity, especially in a data pre-processing context.

References

Almotiri, J., Elleithy, K., & Elleithy, A. (2017, May 1). *Comparison of autoencoder and principal component analysis followed by neural network for e-learning using handwritten recognition* [Pages: 5]. https://doi.org/10.1109/LISAT.2017.8001963

Asadi, R., & Regan, A. (2019). A convolution recurrent autoencoder for spatio-temporal missing data imputation. *arXiv:1904.12413 [cs, stat]*. Retrieved April 20, 2022, from http://arxiv.org/abs/1904.12413

Beaulac, C. (2021, October 4). *A moment-matching metric for latent variable generative models* (arXiv:2111.00875) [type: article]. arXiv. https://doi.org/10.48550/arXiv.2111.00875

Beaulieu-Jones, B. K., & Moore, J. H. (2017). MISSING DATA IMPUTATION IN THE ELECTRONIC HEALTH RECORD USING DEEPLY LEARNED AUTOENCODERS. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, *22*, 207–218. https://doi.org/10.1142/9789813207813_0021

Bergeron, M., Fung, N., Poulos, Z., Hull, J. C., & Veneris, A. (2021, April 15). *Variational autoencoders: A hands-off approach to volatility* (SSRN Scholarly Paper ID 3827447). Social Science Research Network. Rochester, NY. https://doi.org/10.2139/ssrn.3827447

Bhowick, D., Gupta, D. K., Maiti, S., & Shankar, U. (2019). Stacked autoencoders based machine learning for noise reduction and signal reconstruction in geophysical data. *arXiv:1907.03278 [physics]*. Retrieved April 19, 2022, from http://arxiv.org/abs/1907.03278

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*(3), 307–327. https://doi.org/10.1016/0304-4076(86)90063-1

Bollerslev, T. (1990). Modelling the coherence in short-run nominal exchange rates: A multivariate generalized arch model [Publisher: The MIT Press]. *The Review of Economics and Statistics*, *72*(3), 498–505. https://doi.org/10.2307/2109358

Bonfigli, R., Felicetti, A., Principi, E., Fagiani, M., Squartini, S., & Piazza, F. (2018). Denoising autoencoders for non-intrusive load monitoring: Improvements and comparative evaluation. https://doi.org/10.1016/J.ENBUILD.2017.11.054

Bourlard, H., & Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, *59*(4), 291–294. https://doi.org/10.1007/BF00332918

Chiang, H.-T., Hsieh, Y.-Y., Fu, S.-W., Hung, K.-H., Tsao, Y., & Chien, S.-Y. (2019). Noise reduction in ECG signals using fully convolutional denoising autoencoders. *IEEE Access*, *7*, 60806–60813. https://doi.org/10.1109/ACCESS.2019.2912036

Dai, B., Wang, Y., Aston, J., Hua, G., & Wipf, D. (2018). Connections with robust PCA and the role of emergent sparsity in variational autoencoder

models. *Journal of Machine Learning Research*, *19*(41), 1–42. Retrieved April 14, 2022, from http://jmlr.org/papers/v19/17-704.html

de Oliveira, M. L. L., & Bekooij, M. J. G. (2020). Deep convolutional autoencoder applied for noise reduction in range-doppler maps of FMCW radars [ISSN: 2640-7736]. *2020 IEEE International Radar Conference (RADAR)*, 630–635. https://doi.org/10.1109/RADAR42522.2020.9114719

Geng, J. (2007, December 10). *Principal component GARCH model* (SSRN Scholarly Paper No. 1068945). Social Science Research Network. Rochester, NY. https://doi.org/10.2139/ssrn.1068945

Horger, F., Würfl, T., Christlein, V., & Maier, A. (2018). Deep learning for sampling from arbitrary probability distributions.

Huang, J., Chai, J., & Cho, S. (2020). Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China*, *14*(1), 13. https://doi.org/10.1186/s11782-020-00082-6

Kondratyev, A. (2018, April 11). *Learning curve dynamics with artificial neural networks* (SSRN Scholarly Paper ID 3041232). Social Science Research Network. Rochester, NY. https://doi.org/10.2139/ssrn.3041232

Kondratyev, A., & Sokol, A. (2020, November 13). *Machine learning for long risk horizons: Market generator models* [Risk Live 2020].

Kramer, M. A. (1992). Autoassociative neural networks. *Computers & Chemical Engineering*, *16*(4), 313–328. https://doi.org/10.1016/0098-1354(92)80051-A

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks [_eprint: https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/aic.690370209]. *AIChE Journal*, *37*(2), 233–243. https://doi.org/10.1002/aic.690370209

Litany, O., Bronstein, A., Bronstein, M., & Makadia, A. (2018). Deformable shape completion with graph convolutional autoencoders. *arXiv:1712.00268 [cs]*. Retrieved April 20, 2022, from http://arxiv.org/abs/1712.00268

Ning, B., Jaimungal, S., Zhang, X., & Bergeron, M. (2022). Arbitrage-free implied volatility surface generation with variational autoencoders. *arXiv:2108.04941 [cs, q-fin, stat]*. Retrieved March 29, 2022, from http://arxiv.org/abs/2108.04941

Pereira, R. C., Santos, M. S., Rodrigues, P. P., & Abreu, P. H. (2020). Reviewing autoencoders for missing data imputation: Technical trends, applications and outcomes. *Journal of Artificial Intelligence Research*, *69*, 1255–1285. https://doi.org/10.1613/jair.1.12312

Shivakumar, P. G., & Georgiou, P. (2016). Perception optimized deep denoising AutoEncoders for speech enhancement. *Interspeech 2016*, 3743–3747. https://doi.org/10.21437/Interspeech.2016-1284

Siwek, K., & Osowski, S. (2017). Autoencoder versus PCA in face recognition. *2017 18th International Conference on Computational Problems of Electrical Engineering (CPEE)*, 1–4. https://doi.org/10.1109/CPEE.2017.8093043

Tran, L., Liu, X., Zhou, J., & Jin, R. (2017). Missing modalities imputation via cascaded residual autoencoder [ISSN: 1063-6919]. *2017 IEEE Con-*

ference on Computer Vision and Pattern Recognition (CVPR), 4971–4980. https://doi.org/10.1109/CVPR.2017.528

Wetzel, S. J. (2017). Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders [Publisher: American Physical Society]. *Physical Review E*, *96*(2), 022140. https://doi.org/10.1103/PhysRevE.96.022140

A Summary statistics

	Mean	**Min**	**Max**	**Std dev**	**Skewness**	**Kurtosis**
ISHARESCOREEUR (XET) STOXX50 UCITS ETF EUR	-0.000	-0.147	0.336	0.016	1.560	52.961
DOW JONES US REAL ESTATE	0.000	-0.214	0.172	0.017	-0.284	21.959
SECT.SPDR TST.SBI INTER INDS.	0.000	-0.143	0.101	0.014	-0.446	8.033
ISHARES INFL.LKD. GOVT BD UCITS ETF EUR A	0.000	-0.176	0.184	0.009	0.767	105.633
ISHARES RUSSELL 2000 ETF	0.000	-0.155	0.135	0.015	-0.343	8.299
ISHARES IBOXX $ HIY.CBD. ETF	-0.000	-0.062	0.104	0.007	0.766	33.638
INVESCO QQQ TRUST SERIES 1	0.000	-0.115	0.166	0.018	-0.141	7.560
ISHARES 10-20 YEAR TREASURY BOND ETF	0.000	-0.051	0.047	0.006	-0.022	6.938
ISHARES CORE FTSE 100 UCITS ETF GBP	0.000	-0.129	0.117	0.013	-0.528	11.563
ISHARES 7-10 YR.TRSY.BD.	0.000	-0.028	0.035	0.004	-0.113	4.328
Gold Bullion LBM $/t oz DELAY	0.000	-0.089	0.104	0.011	-0.201	6.733
INVESCO MSCI JAPAN UCITS ETF ACC	0.000	-0.106	0.057	0.011	-0.799	8.928
S&P GSCI Energy Total Return	0.000	-0.279	0.155	0.021	-0.784	11.196
INVESCO DB AGRICULTURE FUND	-0.000	-0.371	0.366	0.014	-0.231	250.450
INVESCO DB BASE METALS FUND	0.000	-0.398	0.347	0.017	-1.062	121.659
RF/CC CRB ER	0.000	-0.100	0.063	0.011	-0.570	5.854
MSCI EM U$	0.000	-0.100	0.101	0.012	-0.529	7.843
MSCI BRAZIL 25-50 $	0.000	-0.194	0.164	0.022	-0.513	8.494
MSCI TAIWAN	0.000	-0.103	0.072	0.014	-0.175	3.720
MSCI EMU SMALL CAP	0.000	-0.124	0.088	0.011	-0.795	8.340
MSCI KOREA 25-50	0.000	-0.132	0.117	0.015	-0.378	6.712
Silver, Handy&Harman (NY) U$/Troy OZ	0.000	-0.135	0.137	0.018	-0.600	6.097
Crude Oil BFO M1 Europe FOB $/BBl	0.000	-0.442	0.215	0.023	-1.188	29.365
Corn US No.2 South Central IL $/BSH	0.000	-0.379	0.399	0.022	0.077	51.798
LME-Copper Grade A Cash U$/MT	0.000	-0.104	0.117	0.016	-0.142	4.578
ISHARES 1-3 YR.TRSY.BOND	0.000	-0.007	0.007	0.001	-0.536	6.212
ISHARES 3-7 YR.TRSY.BOND	0.000	-0.017	0.018	0.002	-0.043	3.954
IBOXX $ LIQUID INVESTMENT GRADE INDEX	-0.000	-0.052	0.032	0.004	-1.462	22.799
CMCI Composite TR (USD) - RETURN IND. (OFCL)	0.000	-0.065	0.060	0.010	-0.459	4.118
S&P 500 COMPOSITE	0.000	-0.095	0.116	0.012	-0.295	8.640
SHANGHAI SE A SHARE	0.000	-0.116	0.127	0.016	-0.386	6.082

Table 13: Summary statistics for the daily financial returns dataset

	Mean	Min	Max	σ	Skewness	Kurtosis
adjusted close	74.102	11.570	146.940	20.594	0.304	-0.306
atmVola	0.276	0.014	2.481	0.093	3.323	39.128
numOptions	175.629	1.000	594.000	185.724	0.784	-0.900
futTTM	2.155	0.011	10.997	1.922	1.563	2.244
opTTM	2.143	0.003	10.989	1.922	1.563	2.243

Table 14: Summary statistics for the implied volatility dataset

B Reconstruction error tables

$\rho = 0.25$ latent dimensions	1	2	3	4	6	12
simulated dimensions						
1	0.88	0.807	0.741	0.651	0.495	0.148
2	0.916	0.814	0.727	0.643	0.475	0.131
3	0.922	0.825	0.706	0.642	0.457	0.135
4	0.926	0.841	0.729	0.631	0.463	0.136
6	0.931	0.844	0.759	0.653	0.462	0.155
12	0.943	0.882	0.798	0.729	0.575	0.165

$\rho = 0.5$ latent dimensions	1	2	3	4	6	12
simulated dimensions						
1	0.658	0.645	0.591	0.515	0.428	0.244
2	0.797	0.651	0.587	0.503	0.371	0.148
3	0.868	0.729	0.594	0.505	0.354	0.121
4	0.897	0.781	0.66	0.507	0.344	0.137
6	0.917	0.823	0.713	0.585	0.358	0.136
12	0.943	0.877	0.809	0.731	0.558	0.128

$\rho = 0.75$ latent dimensions	1	2	3	4	6	12
simulated dimensions						
1	0.392	0.379	0.353	0.328	0.305	0.208
2	0.569	0.455	0.351	0.295	0.254	0.163
3	0.723	0.556	0.39	0.324	0.223	0.127
4	0.864	0.685	0.528	0.355	0.253	0.114
6	0.899	0.766	0.651	0.516	0.258	0.12
12	0.943	0.876	0.798	0.714	0.558	0.132

Table 15: Mean reconstruction error for Gaussian VAE on Gaussian simulated data for different correlations, amount of simulated dimensions and latent dimensions

$\rho = 0.25$ latent dimensions	1	2	3	4	6	12
simulated dimensions						
1	0.886	0.807	0.732	0.659	0.484	0.138
2	0.912	0.81	0.703	0.637	0.492	0.125
3	0.921	0.824	0.725	0.63	0.477	0.144
4	0.929	0.836	0.75	0.655	0.451	0.116
6	0.931	0.856	0.765	0.657	0.463	0.137
12	0.943	0.877	0.811	0.73	0.554	0.135

$\rho = 0.5$ latent dimensions	1	2	3	4	6	12
simulated dimensions						
1	0.667	0.64	0.588	0.528	0.407	0.224
2	0.808	0.667	0.596	0.522	0.383	0.137
3	0.88	0.725	0.582	0.482	0.362	0.123
4	0.897	0.77	0.644	0.501	0.359	0.131
6	0.916	0.82	0.716	0.61	0.343	0.143
12	0.943	0.88	0.804	0.723	0.564	0.144

$\rho = 0.75$ latent dimensions	1	2	3	4	6	12
simulated dimensions						
1	0.4	0.372	0.344	0.333	0.312	0.207
2	0.555	0.416	0.358	0.31	0.246	0.172
3	0.796	0.567	0.391	0.343	0.228	0.143
4	0.853	0.689	0.546	0.333	0.222	0.116
6	0.902	0.791	0.644	0.517	0.244	0.115
12	0.943	0.87	0.805	0.708	0.565	0.133

Table 16: Mean reconstruction error for student-t VAE on Gaussian simulated data for different correlations, amount of simulated dimensions and latent dimensions

C VaR estimates plots

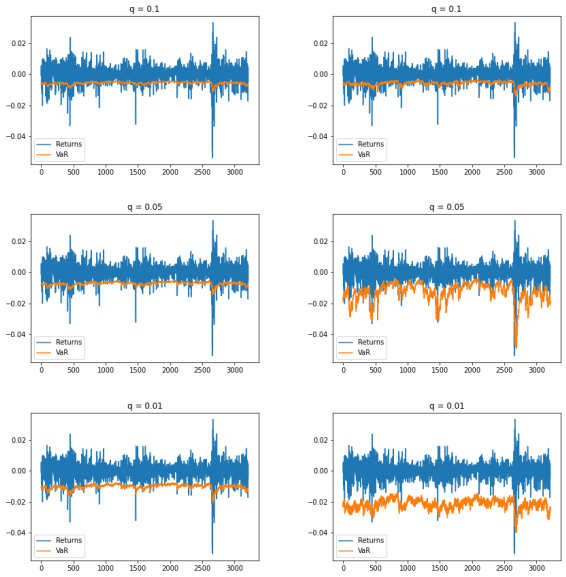

Figure 6: Student-t VaR estimates for different quantiles. PCA O-GARCH is shown on the left hand side, and VAE O-GARCH is shown on the right hand side.

D Corrected implied volatility curves

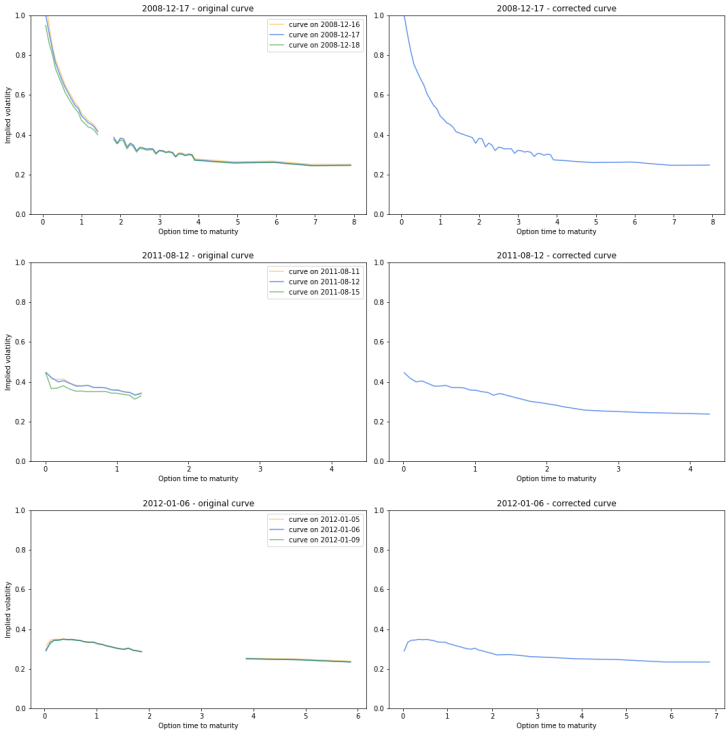

Figure 7: Implied volatility curves corrected using a Student-t VAE