# Representation Learning for Regime Detection in Financial Markets

**Alexa Orton**

**Supervisor: A/Prof. Tim Gebbie**

A minor dissertation submitted in partial fulfilment of the requirements for the degree of
Master of Science in Data Science Specialising in Financial Markets at the Department
of Statistical Sciences, University of Cape Town

2024

## Abstract

We investigate financial market regime detection from the perspective of deep representation learning of the causal (reflexive) information geometry underpinning complex (multi-scale) dynamical traded asset systems using an emergent hierarchical correlation structure to characterise evolving macroeconomic market phases. Specifically, we assess the robustness of three toy models: SPD Matrix Network (SPDNet), SPD Matrix Network with Riemannian Batch Normalisation (SPDNetBN) and U-shaped SPD Matrix Network (U-SPDNet) whose architectures respect the underlying Riemannian manifold of input block hierarchical Symmetric Positive Definite (SPD) correlation matrices by employing Log-Euclidean Metric (LEM)s. Market phase detection for each model is carried out using three data configurations: i.) Randomised Johannesburg Stock Exchange (JSE) Top 60 data, ii.) synthetically-generated block hierarchical SPD matrices, and iii.) chronology-preserving block-resampled JSE Top 60 data. We show that using a singular performance metric is misleading in our financial market use cases. We confirm that U-SPDNet performs improved latent feature extraction with better classification performance in stressed and rally market phases, despite achieving lower Out-of-Sample (OOS) backtest scenario accuracy than that of the benchmark SPDNet. The SPDNet-based models fail in capturing latent reflexive spatio-temporal block hierarchical correlation dynamics and deliver corner solutions across all input data sets. The U-SPDNet is promising in terms of its utility in regime dependent portfolio optimisation strategy generation as a model better-suited to capturing latent block hierarchical correlation structures arising from lead-lag causal feedback information loops that often drive the evolution of evolving market regimes.

**Keywords:** deep manifold representation learning, SPD matrix classification, nested block hierarchical financial market correlations, regime detection, causal feedback loops.

# Plagiarism Declaration

I am presenting this dissertation in partial fulfilment of the requirements for my degree. I know the meaning of plagiarism and declare that all of the work in this dissertation, save for that which is properly acknowledged, is my own. I hereby grant the University of Cape Town free licence to reproduce for the purpose of research either the whole or any portion of the contents in any manner whatsoever of this dissertation.

# Acknowledgements

Without the unfailing patience of my supervisor, Tim Gebbie, the completion of this dissertation would not have been possible. Thank you for lending a sense of adventure to this realm of research and for guiding the narrative of this investigation.

To my loved ones - I am grateful for your patience and support; spurring me on even in times of struggle.

In this Brave New World, I would like to acknowledge a little help from ChatGPT.

Computations were performed using facilities provided by the University of Cape Town's ICTS High Performance Computing team: `hpc.uct.ac.za`.

# Contents

# Acronyms

**BBG** Bloomberg. 26

**BiMap** Bilinear Mapping. 10

**CEL** Cross-Entropy Loss. 12, 16

**ConvNet** Convolutional Neural Network. 10

**CorrGAN** Correlation Matrix Generative Adversarial Network. 9

**DreamNet** Stacked Riemannian SPD Matrix Network. 12

**ExpEig** Eigenvalue Exponential. 12

**FC** Fully-Connected. 18

**FPHA** First Person Hand Action. 11

**GANs** Generative Adversarial Networks. 9

**GFC** Global Financial Crisis. 26

**GTN** Graph Transformer Network. 19

**HCBM** Hierarchical Correlation Block Model. 9

**HNFM** Hierarchically Nested Factor Model. 8

**JALSH** Johannesburg All-Share Index. 26

**JSE** Johannesburg Stock Exchange. 1, 3

**LEM** Log-Euclidean Metric. 1, 14, 17

**LF** Loss Function. 38

**LFE** Log-Fusion-Exp. 16

**LogEig** Eigenvalue Logarithm. 10

**ML** Machine Learning. 7, 8

**MRI** Magnetic Resonance Imaging. 10

**MSE** Mean Square Error. 12

**MSTs** Minimum Spanning Trees. 6

**MSTs** Minimum Spanning Trees. 8

**MV** Mean-Variance. 60

**MVPO** Mean-Variance Portfolio Optimisation. 4, 65

**OOS** Out-of-Sample. 1, 61, 63

**PCA** Principal Component Analysis. 7

**RA** Research Aim. 3, 4

**RAE** Riemannian Autoencoder. 11

**RBN** Riemannian Batch Normalisation. 11, 38

**RD** Regime-Dependent. 4, 60

**ReEig** Eigenvalue Rectification. 10

**RELU** Rectified Linear Unit. 10

**RET** Reconstruction Error Term. 12, 16, 38

**RMT** Random Matrix Theory. 12

**RS** Regime Switching. 1, 6, 7

**SGD** Stochastic Gradient Descent. 10

**SPD** Symmetric Positive Definite. 1, 9, 39, 61

**SPDNet** SPD Matrix Network. 1, 3

**SPDNet-$i$BiRe** SPDNet with $i$ alternating BiMap and ReEig layers. 11, 13

**SPDNetBN** SPD Matrix Network with Riemannian Batch Normalisation. 1, 3

**SR** Sharpe Ratio. 7, 31

**SRAE** Stacked Riemannian Autoencoder. 12

**SVD** Singular Value Decomposition. 38, 39

**TIPP** Time-Invariant Portfolio Protection. 6

**TMD** Transformation Matrix Dimension. 38

**U-SPDNet** U-shaped SPD Matrix Network. 1, 3, 11

**VAEs** Variational Autoencoders. 9

# 1 Introduction

Financial markets are complex dynamic systems characterised by emergent hierarchical structures and reflexive information feedback loops which drive them through varying macroeconomic regimes and semi-equilibria. These co-evolving network dynamics are encoded in correlation matrices which have provided a broad context for the data-driven development of strategies for both learning the latent risk factor representation of the market's hierarchical structure and accurate regime analysis. In light of recent financial crises, it is all the more important to the development of quantitative investment strategies to identify techniques which accurately detect shifts in macroeconomic regimes as underpinned by the hierarchical latent features and non-linear causal relationships driving market evolution (Miori and Cucuringu, 2022; Papenbrock et al., Spring 2021).

Market regime discovery has been approached from multiple angles in studies considering Regime Switching (RS) and change point detection in the context of portfolio management (Ang and Bekaert, 2004; Nystrup et al., 2018). Other researchers have investigated the dimensional reduction problem for time series modelling under the names of temporal clustering and volatility modelling (Hendricks et al., 2016; Bollerslev and Ole Mikkelsen, 1996). Specifically, researchers have claimed to have been able to identify time-scale specific hierarchical system behaviours, of market participants, from correlation matrix analysis.

Families of correlation matrices act as a useful tool in developing an understanding of the dynamics at work in the presence of bottom-up and top-down noise and self-organising effects corresponding to differing regimes despite the market changing with changes in economic interactions and innovations effected by strategic purposeful agents (Taleb, 2022; Wilcox and Gebbie, 2014). Indeed, the collective behaviour of millions of market participants alters the temporal market structure via reflexive causal links (Singo, 2022). The scale and potential information content of the data may provide the means for deep representation learning models to investigate and incorporate time as an intrinsic variable linking non-linear feedback loops and the emergence of regimes, with evolving hierarchical structures.

Causal modelling is not a novel area of research. More recently, however, the study of causal modelling has seen a revival in the applied context of quantitative finance, where practitioners and researchers have pivoted away from market time series prediction in favour of developing strategies to analyse market phases using information geometry-based modelling. There are natural concerns with any attempt to extract causal models in the context of financial markets because of their reflexive nature (Polakow et al., 2023). However, the pivot from time series methods to information and geometry based methods, seeking repeated predictable structures and relationships, remains promising. Specifically, the importance of temporal, non-linear, causal dynamics at work which give rise to both regime and hierarchical market structure evolution has been given increased weight, motivating the investigation of causal manifold-based deep learning techniques (Rodriguez Dominguez, 2023). Causal modelling provides a lens through which the Out-of-Sample (OOS) effectiveness of deep learning models may be appraised in terms of detecting shifts in market semi-equilibria amongst regimes whilst considering lead-lag correlation information links.

## 1.1 Challenges

1. **The Euclidean assumption**: A significant problem in previous studies processing and

classifying financial market correlation matrices is the assumption that their information geometry is Euclidean in nature, when in fact their underlying space is a curved Riemannian manifold, or SPD manifold (Marti, 2020a). Crucial to SPD correlation matrix learning techniques is accounting for the particular Riemannian geometry of their underlying manifold (Pennec et al., 2006).

2. **Stationarity**: Further limitations in literature presenting algorithms for hierarchical clustering and state discovery of financial time series are the selection of groupings *a priori*, and assumption of underlying distribution stationarity.

3. **Robustness**: The robustness of investment strategies depends on their being antifragile in some scenarios. The training and validation of models which aim to effect state discovery for input to portfolio construction optimisation techniques fail to encompass all possible market scenarios due to the fact that history never repeats itself (Papenbrock et al., Spring 2021). Simple backtesting techniques fail as they only provide information about past events, meaning that model predictions are not equivalent to an explanation of market dynamics. In the presence of bidirectional, nonlinear causal information diffusion affecting market state shifts and emergent hierarchical organisation in market participant behaviour, it is difficult to effect true scientific testing which requires replication of scenarios.

4. **Reflexivity**: Risk is yet to be quantified in tractable ways (Taleb, 2022). The poverty of historicism proposes that the coined 'Oedipus effect' is at work in complex systems characterised by the information geometry of risk (Popper, 1944); specifically, that the influence of an item of information on the situation to which the information refers exists and may act in either destructive or reinforcing ways, resulting in a system passing through pseudo-equilibria with non-linear causal links to historical data. Prediction may be seen as a social happening, as opposed to a quantitative identification of likely future events (Polakow et al., 2023).

## 1.2 Research Aims

In short, context has been created for the development of novel approaches in quantitative finance which strive to achieve specific goals in order to address the above problems (Marti et al., 2016):

1. Representation learning of stable features underpinning market dynamics such that the temporal, hierarchical SPD Riemannian manifold information structure is preserved;

2. The construction of realistic synthetic correlation matrices (Yelibi and Gebbie, 2021) ; and

3. Model accuracy being considered through the lens of information flow in spatio-temporal dimensions.

In turn, the successful realisation of these aims by innovative modelling techniques with respect the latent hierarchical structure present in financial markets data allows for:

4. Employing this stable factor embedding, or latent representation, in performing market regime classification with improved accuracy (Marti, 2021);

5. Using synthetic data for robust model evaluation OOS by considering both real and synthetic manifestations of market scenarios;

6. Using the learnt representations and improved market regime detection for systematic regime-dependent investment strategy construction and validation (Taleb, 2022),

amongst many other similarly-reinforcing applications in quantitative finance methods (Papenbrock et al., Spring 2021).

## 1.3 Implementation

This dissertation aims to contribute to the body of research intersecting three key apexes: 1) deep representation learning of complex financial market systems characterised by SPD correlation matrices, 2) the block hierarchical structure emerging from financial market evolution as considered through the lens of market reflexivity dynamics, and 3) market regime detection for market-timed robust portfolio strategy selection.

First, we turn to the field of Computer Vision; specifically, we consider deep learning techniques for the particular information geometry of SPD correlation matrix manifolds with a block hierarchical structure for the task of market state discovery. Such models are key to understanding statistical features driving market evolution which could improved regime detection accuracy. Testing predictive economic models for risk quantification in a non-equilibrium environment for generalisation is challenging. Greater value may instead be realised by considering market regimes as characterised by *fragility*; specifically, expanding on techniques which aim to learn the possible hierarchical structure underpinning market dynamics. This may be able to replace the explicit analysis of the correlation matrices on a sector-by-sector basis within the indexes, to possibly reveal which sectors are contributing most to overall market fragility at the level of the representation.

Second, in order to address the first Research Aim (RA) 1, and by extension the realisation of RA 4 above we first establish a benchmark algorithm. Here, we first train and validate the SPD Matrix Network (SPDNet) proposed by Huang and Gool (2016) to establish a benchmark for deep representation learning of the SPD manifold of hierarchical correlation matrices for regime detection. We compare this models performance to that of the SPD Matrix Network with Riemannian Batch Normalisation (SPDNetBN) introduced by Brooks et al. (2019a) to determine whether regime classification accuracy improves as a function of batch normalisation respecting the Riemannian manifold upon which the SPD correlation matrices sit. Finally, the U-shaped SPD Matrix Network (U-SPDNet) proposed by Wang et al. (2022b) is considered to evaluate model capacity for latent block hierarchical feature extraction in conjunction with regime detection accuracy as a result of an augmented Riemannian autoencoder structure and increased information pass-through.

Third, RA 2 is addressed by considering each of these models in the context of hierarchically-ordered Johannesburg Stock Exchange (JSE) Top 60 returns correlation matrices, as computed by a single-linkage Ward algorithm, along with synthetically-constructed nested block hierarchical correlation matrices (Papenbrock et al., Spring 2021; Yelibi and Gebbie, 2021).

The dissertation is structured as follows: Section 2 describes the hierarchical information structure underlying the evolution of financial markets through varying macroeconomic regimes, as characterised by the SPD correlation matrix, and the significance of non-linear information feedback loops in this market state and structure evolution. Further, Section 2 discusses deep representation learning strategies which respect the particular information geometry of these hierarchical SPD correlation matrices found on the Riemannian manifold. Selected manifold-learning model methodologies are detailed in Section 3, namely: SPDNet, SPDNetBN and U-SPDNet. Section 4 describes the JSE Top 60 and synthetic hierarchical correlation matrix data sets considered in this analysis in conjunction with model parameterisation and implementation details. Comparison of model performance in terms of both OOS regime identification accuracy and hierarchical market structure embedding is discussed in Section 5. An alternative block sampling technique is investigated in Section 6 to uncover model effectiveness in learning temporal correlation structure dynamics through market regime changes in line with the research aims (RA 4). In addressing RA 6, a simple asset allocation problem is considered in Section 7, where Regime-Dependent (RD) Mean-Variance Portfolio Optimisation (MVPO) using OOS SPDNet and U-SPDNet market state estimation output are compared to a vanilla mean-variance portfolio optimisation on JSE Top 60 data. Finally, Section 8 concludes this research with some experimental findings and identification of future research directions.

# 2 Representation Learning of Hierarchical Information

> Hierarchy. . . is one of the central structural schemes that the architect of complexity uses.
>
> *Herbert A. Simon*

## 2.1 Literature Review

### 2.1.1 Hierarchical Structures in Financial Data

Financial markets are complex dynamic systems exhibiting hierarchical emergent properties due to their self-organising nature. The dynamics of financial markets as complex systems are significantly affected by the hierarchical structure of interactions amongst assets. Multiple studies have provided a quantitative description of financial system hierarchies by extracting information as to the group composition in which the market is intrinsically organised by virtue of hierarchical clustering methods (Borghesi et al., 2007; Bonanno et al., 2000).

Indeed, the seminal work of Mantegna (1999b) has acted as the kernel of a broad body of research verifying the hierarchical correlation structure of asset price returns time series in several different global markets (Marsili, 2002; Marti et al., 2016; Papenbrock et al., Spring 2021). Mantegna (1999b)'s econophysics-motivated approach to market sector and state detection is distribution-free and bucks the methodological trend of models previously observed in the literature which erroneously assume financial time series stationarity.

Investigations stemming from Mantegna (1999b)'s initial findings have confirmed an intuitive block hierarchical structure upon stock market correlation matrices being subjected to spectral analysis; specifically, the price returns time series of individual assets are influenced particularly by the group or sector to which they belong which gives rise to empirical correlation matrices exhibiting diagonal nested blocks corresponding to groupings revealed by clustering algorithms applied to financial time series. Application of Mantegna (1999b)'s algorithm to a portfolio of 443 stocks sampled from the S&P reveals a sub-dominant ultrametric space. The associated dendrogram is proven to encode the links between traded stocks with an associated meaningful economic taxonomy characterised by four main groups: a group of financial services, capital goods, retailers, consumables and tobacco companies; a group of oil and energy companies; a group of communication and electrical utility companies; and a group of raw material companies. Furthermore, the information regarding economic factor effects for specific groups of stocks may be obtained solely from the correlation matrices of stock price returns themselves (Mantegna, 1999a).

Further works have uncovered subtleties in the dynamic nature of this emergent hierarchical property of financial market systems, in addition to limitations of this seminal market state detection and clustering algorithm methodology. Specifically, Marti et al. (2021) identify that the choice of the Pearson correlation metric as in Section 4.2.3 as the distance input measure for stock interrelatedness is susceptible to the influence of outliers and is ill-suited to the non-Gaussianity of financial time series. It is also shown by Tumminello et al. (2007) that high correlation edges in Minimum Spanning Trees (MSTs) are unreliable depictors of strong links between two stocks.

Despite these limitations, a key relationship is identified in the literature on financial time series clustering, namely: hierarchies do indeed exist, and vary in degree of cluster separability or intensity dependent on the overarching macroeconomic regime. Thus the question is invoked:

*How are hierarchies and financial market regimes, both empirically-validated, related? (Marti, 2020a).*

### 2.1.2 Market Regimes

Evolutionary financial market dynamics vary significantly with time. These sustained underlying dynamic shifts are characterised by different macroeconomic regimes.

Accurate market regime detection has important implications for asset allocation and portfolio construction strategy selection. The study of correlation matrices is particularly relevant due to their information-rich representation of underlying structures in financial markets data whilst reflecting overall market trend.

Indeed, spectral analysis studies performed on MSTs built from stock returns' correlation matrices have revealed regime-specific persistent market network phenomena, even as the network evolves with the passage of time. One such empirically-observed phenomenon is that stressed markets display greater asset correlation, irrespective of asset class or data sampling frequency (Miori and Cucuringu, 2022).

Elevated systemic risk as reflected by correlation matrices computed using stock returns data from bear markets are evidence of *fragility* of the system as market-wide factors begin to dominate participant behaviour characterised by overreaction Taleb (2022). Economic participants in equity markets exhibiting herding behaviour in bear market periods are associated with regimes of increased volatility and interest rates, suppressed returns and high persistence. The statistically significant asymmetry of correlations observed in stressed traded asset markets as opposed to their normal or rally counterparts provides motivation for studying their relevance in correlation-dependent financial product development and implementation. Further, market-timing strategies which exploit changing regime-specific underlying data distributions have been shown to add value to portfolio protection products (Time-Invariant Portfolio Protection (TIPP), for example) and risk-off investment decisions in action spaces not only confined to those concerning equities (Ang and Bekaert, 2004; Papenbrock et al., Spring 2021).

Ang and Bekaert (2004) identify two broad market regimes, namely 'normal' and 'bear' markets. Using a Regime Switching (RS) asset returns model for portfolio asset allocation, the regimes are accurately reconstructed by assuming returns data points may be drawn from differing underlying data generation distributions, which supports the empirically-observed asymmetric correlation distribution. Recently, centrality measures computed on correlation coefficients representing evolving equity financial market networks have been employed in matrix clustering, regime detection and risk-diversified portfolio construction by extension (Borghesi et al., 2007).

Miori and Cucuringu (2022) extend these studies to returns correlations across multiple asset classes and establish a purely data-driven technique for distinguishing regimes. Influenced by spectral analysis techniques, cophenetic correlation matrices are constructed as correlations between the distance matrix found as $\sqrt{2(1-\|C\|)}$ where $C$ is Pearson's linear correlation, sitting on the Euclidean manifold, and the associated dendrogram generated by a hierarchical linkage algorithm. Six regimes are uncovered via KMeans++ clustering using the elbow method for cluster number selection.

### 2.1.3 Regimes and Hierarchies

In light of recent financial crises, it has become essential to develop techniques to understand and identify how shifts in stock market interconnection and diversification properties, as characterised by their correlation matrices, manifest in different regimes (Papenbrock et al., Spring 2021).

Exploratory analysis by Miori and Cucuringu (2022) reveals each regime being characterised by block hierarchical average correlation matrices, differing (absolute) correlations, and a varying annualised SR amongst the studied asset classes. These findings support the work of Ang and Bekaert (2004) where different regimes are associated with differing underlying distribution dynamics, and additionally confirm the consistency of the hierarchical nature of traded asset correlations through the stochastic mechanics of the market in different regimes. Marti (2020a) confirms these findings in S&P 500 market regimes, where the first eigenvalue associated with absolute average correlation is indeed greater in a stressed regime. Furthermore, both inter- and intra-industry correlation strengths are greater in bear markets which means that the market's hierarchy loses definitive cluster separability in times of market stress. Additionally, this work identifies a positive relationship between the granularity of the market hierarchy and prediction of a stressed regime via feature importance analysis of the cophenetic correlation computed using the single linkage algorithm as in Miori and Cucuringu (2022).

Against this backdrop, a pertinent question is:

*How does the empirical evidence of hierarchical structure in correlation matrices correlate to the presiding macroeconomic regime in terms of spatio-temporal dimensions?*

Cue causal modelling: a proposed area of research which may explain contemporaneous hierarchical behaviour and market regime changes as a function of bidirectional, non-linear feedback loops caused by millions of market participant interactions with news and asset price action. As a kernel idea in the causal modelling realm of research, the concept of market *reflexivity* proposed in the seminal work of Soros (2013) provides a plausible connective theory as to the semi-equilibrium dynamics at work leading to the formation of hierarchical behavioural structure and the evolution of the market through regimes as a function of self-reinforcing and destructive feedback loops created via self-organising agent interaction with market information (Polakow et al., 2023).

## 2.2 Applications

### 2.2.1 Hybrid Regime Detection Approaches

An integration of unsupervised and supervised techniques in regime identification and subsequent prediction driven by asset correlation matrix information alone is gaining traction in the context of RS strategies for portfolio management and trading strategy selection (Marti, 2020a).

Akioyamen et al. (2020) exhibit this integration of Machine Learning (ML)-based approaches, as in the work of Miori and Cucuringu (2022), by using Principal Component Analysis (PCA) for dimensionality reduction of correlation matrices before applying the $k$-means algorithm to identify regimes without explicitly modelling the underlying generative distributions. A hybrid approach is proposed where supervised ML models are trained on the identified regimes as inferred from the unsupervised ML methods and are shown to generate value when tactical asset allocation and tail hedging strategies are back-tested on S&P 500 equity returns data. Similarly, Marti (2020a) fits a random forest model to S&P 500 correlation matrices arbitrarily classified according to a SR-motivated decision boundary. Despite the regime class construction being manually conducted without unsupervised ML methods, the classifier achieves an OOS prediction accuracy of 80% which corroborates correlation matrices encoding deep information features regarding the structures underpinning market evolution.

These approaches fail to incorporate the information geometry particular to the SPD correlation matrices being scrutinised for the extraction of signals characterised by hierarchies specific to different regimes. An approach proposed by Marti (2020a), which integrates deep SPD manifold representation learning with data preprocessing employing data-informed clustering techniques, provides an appropriate synthesis of spectral analysis-motivated correlation matrix labelling with supervised deep learning techniques better-suited to learning the underlying block hierarchical structure of the market across different regimes. Specifically, it is posited that regime detection effected via the SPDNet algorithm would benefit in terms of OOS prediction accuracy if applied to cluster-determined labelled correlation matrices due to the widely-evidenced embedding of hierarchical market dynamics in such data structures (Marti, 2020a).

### 2.2.2 Realistic Hierarchical Correlation Matrices

History does not perfectly repeat itself, and so future market regime shifts cannot be identical to those previously observed. If models trained with the aim of accurate market regime identification are back-tested on historical data alone, they will fail to perform should novel shifts in the underlying market structure arise (Papenbrock et al., Spring 2021). Thus it is important to simulate new market scenarios which realistically capture empirically-validated phenomena in order to test supervised market regime detection models for use in robust correlation-sensitive investment strategy selection.

Such phenomena, or stylised facts, may be used to evaluate the validity of synthetic matrices. Marti (2020b) describes these as:

1. The distribution of pairwise correlations has a significantly long right tail;
2. Eigenvalues follow the Marçhenko-Pastur distribution with the exception of a large first eigenvalue representing the market and other outlying larger eigenvalues representing broad industry clusters (López de Prado, 2018);
3. The Perron-Frobenius property is satisfied;
4. A hierarchical structure is present in the correlations; and
5. The scale-free property of the corresponding MSTs is present.

Multiple approaches to the systematic generation of hierarchical financial time series have been established aiming to satisfy these matrix evaluation requirements. Following the work of Mantegna (1999b), Tumminello et al. (2007) propose the Hierarchically Nested Factor Model (HNFM) to pursue Item 4 from the above list. As complex systems, financial markets are characterised by hierarchical trees which may be used to extract factor models where independent factors exert an effect on stocks belonging to different clusters with varying intensities in a nested fashion (Tumminello et al., 2010). The HNFM is constructed by association of the similarity matrix produced by the hierarchical clustering of financial returns data to the correlation matrix which characterises a factor model of the system. The HNFM produces multivariate time series retaining the whole information set revealed by hierarchical clustering of the stock price returns data (Tumminello et al., 2007).

The incorporation of multiple factors in generating more realistic complex time-series models with block hierarchical structures is expounded upon by Yelibi and Gebbie (2021), where the HNFM proposed by Tumminello et al. (2007) is associated with a financial multivariate dataset whilst retaining all nested system information as given by the correlation matrix implied by the data. This technique is effective in constructing realistic multi-factor correlation structures from nested time series where the diagonal block structure is clear; five hierarchies, each with three sub-clusters, are constructed using this specialised version of Tumminello et al. (2007)'s nested linear model.

As a component of testing clustering methodologies on financial market returns data, Marti et al. (2016) extend the Hierarchical Correlation Block Model (HCBM) which is motivated by the hierarchical, noise-corrupted correlation structure of traded assets initially proposed by Balakrishnan et al. (2011). To this end, a noise model is described where a similarity matrix is additively perturbed. The similarity matrices are decomposed into ideal matrices with high-variance noise terms. These ideal matrices represent the case as in financial markets data where a data set is comprised of clusters with complex shapes that lie on arbitrary manifolds; each cluster corresponds to a block, and the correlations particular to blocks nested within hierarchies follow a decreasing nested pattern themselves (Marti et al., 2016).

A popular class of methods used by quantitative funds and trading firms is Monte Carlo simulation to construct asset processes characterised by hierarchies and regime shifts (Lopez De Prado Lopez, 2022).

More recently, AI-based, nonparametric techniques have been employed in data generation where the underlying distribution structure is assumed to be approximated. In this vein, neural network-based approaches have been extended to the context of simulating financial market correlation matrices such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) (Lopez De Prado Lopez, 2022). This approach has produced realistic scenarios in order to systematically test investment strategies. Specifically, Marti (2020b) develop the Correlation Matrix Generative Adversarial Network (CorrGAN) model which requires few assumptions or input data to sample realistic SPD correlations from an $n-$dimensional elliptope. Additionally, a novel technique applied to financial correlation matrices is that of *matrix evolutions*, a multi-objective evolutionary algorithm that performs simultaneous optimisation of objective functions characterised by trade-offs along the Pareto frontier. Each optimisation is evaluated according to the set of stylised facts listed in 2.2.2, iterating until the distance from the original correlation matrix reaches a desired level (Papenbrock et al., Spring 2021).

## 2.3 SPD Matrix Representation Learning

Symmetric Positive Definite (SPD) covariance matrices represent deep structures present in complex dynamic systems. Recently, they have entered the scope of machine learning applications due to their usefulness in statistical feature extraction which may be leveraged for tasks varying from analysis of Magnetic Resonance Imaging (MRI) to drone recognition from radar observations (Brooks et al., 2019a). Further, these matrices have been encountered and made use of ubiquitously in second-order pooling and object classification tasks (Ionescu et al., 2016) in addition to analysis of financial market stock price returns data embeddings for portfolio construction (Dolphin et al., 2023). Financial market data provides a popular context for these methods because of the domain dependencies of classical portfolio optimisation that require appropriate learning of statistical representations of returns data as characterised by the population covariance matrix (Taleb, 2022).

Crucial to SPD matrix learning techniques is accounting for the particular Riemannian geometry of their underlying manifold (Pennec et al., 2006). In order to circumvent the inability of Euclidean metrics to avoid problems such as swelling of diffusion tensors when applied on the Riemannian manifold, tailored metrics have necessarily been developed to encode their curved geometry (Sra, 2013; Arsigny et al., 2007). Recently, these techniques have attracted considerable attention in the realm of deep Riemannian manifold learning and have been incorporated into machine learning models for improved feature space replication and classification accuracy.

### 2.3.1 Backbone SPD Matrix Learning Architectures

Machine learning techniques are observed to produce markedly improved classification results upon respecting the Riemannian geometry of their input space in the context of analysing systems with complex structures (Brooks et al., 2019a). In the particular instance of deep neural networks, the seminal work of Huang and Gool (2016) develops a model architecture analogous to a classical Euclidean Convolutional Neural Network (ConvNet), namely, the SPDNet. This re-imagining of the classical ConvNet provides the backbone, or kernel, for many manifold-based learning architectures designed thereafter, with modifications made in the interest of improved classification accuracy and information retention (Brooks et al., 2019a; Wang et al., 2022a; Wang et al., 2022b).

As a crucial contribution to the realm of manifold learning research, the SPDNet developed by Huang and Gool (2016) is adapted specifically to account for SPD matrices where non-linear computations and objective function learning through backpropagation algorithms maintain the SPD structure through the layers. The Riemmanian geometry is integrated into the classical approach by executing data compression and weight space update algorithms in conjunction with a Stiefel-constrained Stochastic Gradient Descent (SGD) optimisation approach.

Specifically, the SPDNet comprises blocks of alternating Bilinear Mapping (BiMap) and Eigenvalue Rectification (ReEig) layers analagous to ConvNet fully-connected and Rectified Linear Unit (RELU) layers respectively. Each layer processes a point on the Riemannian manifold. The BiMap layer aims to generate more compressed and discriminative SPD matrices which preserve the original input data structure, which corresponds to a newly-formed SPD manifold. The ReEig layer rectifies the discriminative performance of the model by tuning up the smallest positive eigenvalues of the SPD matrix which introduces non-linearity to the SPDNet. Alternative techniques exist, as in the work of Cybenko (1989) where sigmoidal function approximation is used to create non-linearity in the input data space. Furthermore, an Eigenvalue Logarithm (LogEig) layer is introduced in the SPDNet such that the feature manifold is converted to the Euclidean space for subsequent classification tasks. This seminal work of Huang and Gool (2016) has provided the backbone structure for Riemannian deep learning algorithms, or 'backbone' as coined by Wang et al. (2022a).

Brooks et al. (2019a) improve on the original experiments of Huang and Gool (2016) and achieve improved classification accuracy by adding Riemannian Batch Normalisation (RBN) to the SPDNet to construct the SPDNetBN. This novel independent layer ensures that no prior knowledge is introduced during model training, and is implemented via use of geometric operations on the manifold, namely, the Riemannian barycenter, parallel transport and non-linear matrix transformations. The RBN layer is learned contemporaneously with the original SPDNet layers and weights space by virtue of a manifold-constrained SGD backpropagation algorithm derived from Huang and Gool (2016). This constrained learning technique leans on backpropagation effected by structured matrix functions which preserve the SPD matrices' eigenspace whilst acting non-linearly on their respective eigenvalues to learn the barycenter through Riemannian gradient updates.

A comparison of classification accuracy between SPDNet and SPDNetBN on the AFEW dataset reveals a significant gain in performance upon introduction of RBN independent of the model depth or training sample size. Of interest is that only marginal accuracy improvement was observed with an increase in model depth agnostic to the integration of RBN, with the addition of three BiMap and ReEig (SPDNet with $i$ alternating BiMap and ReEig layers (SPDNet-$i$BiRe), where $i = 3$) blocks yielding an accuracy improvement of 3.8% and 2.2% for the SPDNet and SPDNetBN respectively (Brooks et al., 2019a). Further, the SPDNetBN is robust to sparse data. This effectiveness is evidenced in the work of Marti (2021) on performing market regime classification using S&P 500 SPD correlation matrices as input for the manifold learning model.

### 2.3.2 Riemannian Autoencoder Architectures

While seminal deep Riemannian classification networks may effectively preserve Riemannian geometry through the model architecture, a crucial failing lies in information loss occurring as a result of iterative data compression mapping on the manifold (Wang et al., 2022b). Echoing observations made in the Brooks et al. (2019a) AFEW dataset experiment, Wang et al. (2022b) identify main structural information, or signal, loss as a limitation of deep SPD manifold networks. This phenomenon is corroborated by their finding no discrepancy in test errors across training epochs between the shallow SPDNet-8BiRe and deeper SPDNet-14BiRe model trained on FPHA data without RBN.

With the stage thus being set for improvements on structural information retention in representation learning on SPD manifolds, in conjunction with increased discriminability for classification amongst input SPD matrices being required, the SPDNet architecture is necessarily extended to the RAE structure. To address these requirements, Wang et al. (2022b) develop a U-shaped SPD Matrix Network (U-SPDNet) for improved visual classification.

The encoder-decoder subsystems of the U-SPDNet are constructed by drawing inspiration from Ronneberger et al. (2015)'s U-Net and using the SPDNet building blocks proposed by Huang and Gool (2016); specifically, a compact RAE-like representation of the input data is achieved by a shrinking encoder path of SPDNet-$i$BiRe layers which is mirrored by an expanding decoder path for upsampling the learned latent features. SPD matrix decoding is performed by replicating the ReEig operations, and the inverse BiMap computation relies on the corresponding weight matrices being constrained to the compact Stiefel manifold.

The effectiveness of U-SPDNet in mining granular geometric features is further ameliorated by appending shortcut connections between the descending and ascending paths, realised via Riemannian metrics, which simultaneously effect simplification of the deep network training by allowing for former information to be exploited for data reconstruction.

Specifically, the Riemannian barycenter approach exploited by Brooks et al. (2019a) is leveraged in this information up-sampling and pooling module, with the introduction of an Eigenvalue Exponential (ExpEig) layer for optimising on the SPD manifold.

Additionally, a critical adaptation to the SPDNet is the introduction of a Reconstruction Error Term (RET). By integrating an RET resembling the Mean Square Error (MSE) loss with the Cross-Entropy Loss (CEL), first used for SPDNet training guided by classification accuracy alone, an identity input-output mapping is encouraged which further supports the U-SPDNet being sensitive to latent feature variation. This RET-steered mapping renders the model more effective in learning a stable representation of the data even when the underlying structure of the correlation coefficients may change dynamically. The U-SPDNet architecture improves on classification accuracy across a number of comparative datasets, namely, the MDSD, FPHA by 6.92% and 8.67% respectively (Wang et al., 2022b).

Wang et al. (2022a) put forth an alternative strategy with the same goal of improving classification accuracy and enriching the representation learning capacity of existing Riemannian networks. The Stacked Riemannian Autoencoder (SRAE) named Stacked Riemannian SPD Matrix Network (DreamNet) is proposed which builds directly on the SPDNet backbone. Similarly to the U-SPDNet, an RET is incorporated into the objective function which encourages the RAE sub-components and overarching stacked architecture to assume an identity mapping whilst diminishing information loss through the embedding. The integration of shortcut connections and associated residual-like blocks further facilitates information interaction between the RAE modules and supports improvement over the baseline SPDNet in classification accuracy when trained on the AFEW, FPHA and UAV-Human datasets. Most crucially, it is shown that with an increased DreamNet depth, namely from 27 to 92 layers, an improvement of 0.88%, 1.4% and 0.34% is seen in these visual classification contexts, respectively, which contrasts with the lack of improvement evidenced by Brooks et al. (2019a)'s experiments with the SPDNetBN.

In terms of structural information degradation, visualisations of latent layers present in DreamNets of varying depths (27, 47, 92) expose a key pattern in the multi-layer learning process, namely that downstream feature matrices are characterised by larger diagonal and diminished off-diagonal values as compared to their encoded predecessors, a result reminiscent of Random Matrix Theory (RMT) methods for correlation matrix cleaning. This result indicates that the shortcut connections which allow for continuous incremental learning as a function of reconstruction residuals does indeed lead to ameliorated capture of structural information present in SPD matrix input data.

# 3 Model Architecture & Learning Methodologies

This section outlines the algorithms implemented coupled with underpinning geometric operation explanations as found in Appendix B. Representative models are selected from the broader deep SPD manifold learning research areas identified for comparison in the context of market regime detection accuracy and block hierarchical correlation matrix feature learning. These include: SPDNet, SPDNet with $i$ alternating BiMap and ReEig layers (SPDNet-$i$BiRe), SPDNetBN and U-SPDNet. For details regarding notation specific to these authors, please refer to Appendix C.

## 3.1 SPDNet Architecture

The architecture of the SPDNet emulates the general components which characterise classical ConvNet's comprising an initial representation computation of the input data space followed by pertinent target function learning and final classification. Here, this approach is couched in operations which respect the particular geometry of the Riemannian manifold $\mathcal{S}_*^+$ of the SPD covariance input matrices as formulated by three main layers shown in Figure 1 (Huang and Gool, 2016).

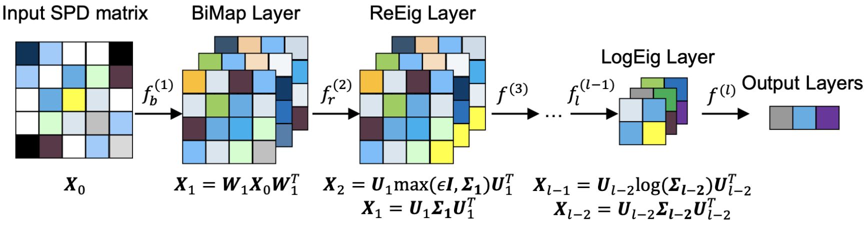


**Figure 1:** Conceptual illustration of the Huang and Gool (2016) SPDNet architecture.

### 3.1.1 BiMap Layer

This fully connected convolution-like layer employs *bilinear mapping* to transform the inputs on an initial SPD manifold into outputs in a more compact and discriminative hierarchical sequence of SPD matrices on a new manifold through representation learning whilst preserving the original data structure. The dimension reduction technique diminishes computation requirements of the learning algorithm. The layer $(l-1)$ transforms an input SPD matrix $\mathbf{X}^{(l-1)} \in \mathcal{S}^+_{n^{(l-1)}}$ of size $n^{(l-1)}$ to a corresponding SPD matrix $\mathbf{X}^{(l)} \in \mathcal{S}^+_{n^{(l)}}$ of dimension $n^{(l)}$ by means of a transformation matrix of connection weights $\mathbf{W}^{(l)}$:

$$\mathbf{X}^{(l)} = \mathbf{W}^{(l)}\mathbf{X}^{(l-1)}\mathbf{W}^{(l)^T} \tag{1}$$

The transformation, or basis change, matrix $\mathbf{W}^{(l)}$, $(n^{(l)} < n^{(l-1)})$ is required to be both full-rank and constrained to a compact Stiefel manifold of semi-orthogonal rectangular matrices of order $\mathcal{O}(n^{(l-1)}, n^{(l)})$. These requirements ensure that the output matrix $\mathbf{X}^{(l)}$ is indeed a valid SPD matrix, and that optimal solutions for the weights exist due to the weight space residing on a manifold on which direct optimisation may be executed.

### 3.1.2 ReEig Layer

The ReEig layer draws on the RELU concept of introducing non-linearity to the architecture of ConvNets to improve on discriminative performance. The ReEig layer jointly leverages the concepts of the ReLU $\max(0,x)$ and eigenregularisation:

$$\mathbf{X}^{(l)} = \mathbf{U}^{(l-1)} \max\big(\mathbf{\Sigma}^{(l-1)}, \epsilon \mathbf{I}_{n^{(l-1)}}\big) \mathbf{U}^{(l-1)^T}, \tag{2}$$

where $\mathbf{X}^{(l-1)} = \mathbf{U}^{(l-1)} \mathbf{\Sigma}^{(l-1)} \mathbf{U}^{(l-1)^T}$ designates the eigendecomposition of $\mathbf{X}^{(l-1)}$ and $\epsilon$ is a rectification threshold. Since $\mathbf{I}$ is the identity matrix, $\max\big(\mathbf{\Sigma}^{(l-1)}, \epsilon \mathbf{I}_{n^{(l-1)}}\big)$ is a diagonal matrix $\mathbf{M}$ with diagonal elements defined as

$$\mathbf{M}(i,i) = \begin{cases} \mathbf{\Sigma}^{(l-1)}(i,i), & \mathbf{\Sigma}^{(l-1)}(i,i) > \epsilon \\ \epsilon, & \mathbf{\Sigma}^{(l-1)}(i,i) \leq \epsilon. \end{cases} \tag{3}$$

This regularisation step ensures that the input SPD covariance matrices do not degenerate by tuning the small positive eigenvalues upwards, where the regularisation term $\epsilon$ prevents singularities around 0 when the covariance matrix is not full rank (Brooks et al., 2019a).

This is analagous to RMT-derived techniques for denoising sample eigenvalues proposed by Laloux et al. (1999), generally known as *eigenvalue clipping*. Following manifold learning theory, this approach supports optimal representation learning of the true underlying data structure. First the input sample covariance matrix is diagonalised while leaving the eigenvectors unchanged and then the noisy eigenvalues (as defined by the isotropic Marčhenko-Pastur law) are shifted upward toward a constant value which reduces the effects of low-variance eigenmodes on robust covariance estimation (Bouchaud and Potters, 2011).

In the context of the SPDNet this constant value $\epsilon$ must be selected as being greater than the $n$-th smallest top eigenvalue.

### 3.1.3 LogEig Layer

The LogEig layer projects and vectorises the output feature SPD manifold of the ReEig layer to a Euclidean space. When viewed in combination with the ReEig layer, this architecture component is contributing a second-order pooling layer for deep feature extraction on a tangent space, as used in computer vision modeling (Rui et al., 2015). This logarithmic mapping, based on Riemannian computing, allows for output layers which may be fed into classical Euclidean dense layers.

The Log-Euclidean Metric (LEM) framework, described in Equation 65, provides the scaffolding for the use of this logarithmic mapping from the manifold $\mathcal{S}_*^+$ to the flat Euclidean space $\mathcal{S}^+$ of symmetric matrices since it implies a Lie group structure on the Riemannian manifold of SPD matrices (Arsigny et al., 2007). In $\mathcal{S}^+$, Euclidean operations may be employed, thus reducing the computational complexity of otherwise-necessary Riemannian computations on $\mathcal{S}_*^+$ whilst maintaining second-order statistical properties. This metric implies a geometric mean which emulates the arithmetic mean in the domain of matrix logarithms, and possesses invariant properties. Furthermore, it is easier to compute than its affine-invariant counterpart described in Section B (Arsigny et al., 2007).

The LEM is used to define the LogEig layer function as:

$$\mathbf{X}^{(l)} = \mathbf{U}^{(l-1)} \log(\mathbf{\Sigma}^{(l-1)}) \mathbf{U}^{(l-1)^T}, \tag{4}$$

where, once again, the eigenspace $\mathbf{U}^{(l-1)}$ is obtained via the eigendecomposition of $\mathbf{X}^{(l-1)}$ and $\log(\mathbf{\Sigma}^{(l-1)})$ is the diagonal matrix of logged eigenvalues. As a final step, the output of this layer function is vectorised in order to produce a layer in the Euclidean space which may be used for further processing using classical Euclidean metric-based layers.

### 3.1.4 Classical Euclidean Network Layer

The LogEig layer, by virtue of projecting and vectorising the output feature manifold to the Euclidean space, facilitates use of a traditional dense layer for further classification tasks. As in experiments executed by Huang and Gool (2016) and Brooks et al. (2019a) a fully-connected Euclidean layer may be inserted directly after the final LogEig layer. The necessary dimensions for these fully-connected filters are $d_k \times d_{k-1}$ where $d_k$ is the total number of separable classes, and $d_{k-1}$ represents the vector dimensionality of the LogEig layer output.

### 3.1.5 Riemannian Batch Normalisation

This batch normalisation layer, which is integrated with the basic SPDNet architecture between the Bimap and ReEig layers, is executed in a different manner during the training and testing phases of the learning process with RBN. The Brooks et al. (2019a) definition of the batch normalisation asserts it as an independent layer meaning leeway is provided for any formulation of the other layers in the SPDNet. During training, normalisation is performed iteratively, where a running sample mean, denoted $\mathfrak{G}_\mathcal{S}$ is updated with each successive batch Riemannian mean $\mathfrak{G}_\mathcal{B}$. Contrastingly, during the testing phase batch normalisation, the final learnt running mean and bias estimates from the training phase are used as statistics for inference.

SPD transport is used to transport the running mean towards the current batch mean by an amount $(1 - \eta)$. If the running mean is a Riemannian mean weighted by a momentum parameter $\eta$, then this is analogous to taking the weighted average of the batch mean and the current running mean in the Euclidean setting. Following Bonnabel and Sepulchre (2010) and adapting Equation 68 to the scenario of barycenter weighting where $\mathbf{w} = \{\eta, 1-\eta\}$, a closed-form solution exists, which corresponds to the geodesic along which the running mean is transported. The running mean $\mathfrak{G}_\mathcal{S}$ is updated by the batch mean $\mathfrak{G}_\mathcal{B}$ as:

$$\mathfrak{G}_\mathcal{S} = \mathfrak{G}_\mathcal{B}^{\frac{1}{2}} (\mathfrak{G}_\mathcal{B}^{-\frac{1}{2}} \mathfrak{G}_\mathcal{S} \mathfrak{G}_\mathcal{B}^{-\frac{1}{2}})^\eta \mathfrak{G}_\mathcal{B}^{\frac{1}{2}} \quad \text{with} \quad \eta \in [0, 1] \tag{5}$$

In the Euclidean setting, momentum is often set as 0.9.

## 3.2 U-SPDNet Architecture

The U-SPDNet proposed by Wang et al. (2022b) emulates the seminal SPDNet model whilst hybridising the architecture of the U-Net (Ronneberger et al., 2015). The problem of information

**Algorithm 1 Riemannian batch normalisation: Training**

```
Require: Batch of $N$ SPD matrices $\{X_i\}_{i\leq N}$, bias $G$, running Riemannian mean $\mathfrak{G}_{\mathcal{S}}$ and momentum $\eta$
  $\mathfrak{G}_{\mathcal{B}} \leftarrow B(\{X_i\}_{i\leq N})$        ▷ compute batch mean
  $\mathfrak{G}_{\mathcal{S}} \leftarrow B_\eta(\mathfrak{G}_{\mathcal{S}}, \mathfrak{G}_{\mathcal{B}})$        ▷ update running mean
  while $i \leq N$ do
    $\bar{X}_i \leftarrow \Gamma_{\mathfrak{G}_{\mathcal{B}} \to I_d}(X_i)$        ▷ center
    $\tilde{X}_i \leftarrow \Gamma_{I_d \to G}(\bar{X}_i)$        ▷ bias
  end while
      return Normalised batch $\{\tilde{X}_i\}_{i\leq N}$
```

**Algorithm 2 Riemannian batch normalisation: Testing**

```
Require: Batch of $N$ SPD matrices $\{X_i\}_{i\leq N}$, learnt bias $G$, final running Riemannian mean $\mathfrak{G}_{\mathcal{S}}$
  while $i \leq N$ do
    $\bar{X}_i \leftarrow \Gamma_{\mathfrak{G}_{\mathcal{B}} \to I_d}(X_i)$        ▷ center using final $\mathfrak{G}_{\mathcal{S}}$
    $\tilde{X}_i \leftarrow \Gamma_{I_d \to G}(\bar{X}_i)$        ▷ bias using learnt $G$
  end while
      return Normalised batch $\{\tilde{P}_i\}_{i\leq N}$
```

degradation whilst learning compact information representation from the input SPD matrices is tackled by augmenting the SPDNet architecture and target function. In particular, as is seen upon comparing Figure 2 and Figure 1, the U-SPDNet establishes an encoder-decoder architecture utilising the SPDNet modules as building blocks, in conjunction with additional shortcut connection Log-Fusion-Exp (LFE) modules and an augmented target function which combines the original CEL with a novel RET. These alterations aim to improve information up-sampling and representational capacity of the model for improved classification accuracy.

As in the work of Brooks et al. (2019b) the LEM detailed in Section B is the selected distance measure used in the necessary geometric operations performed by the U-SPDNet layers and in Riemannian optimisation by SGD. Classification is performed in line with the methods used by Huang and Gool (2016) and Brooks et al. (2019a). The latent feature matrix is transformed to the lower-dimension Euclidean manifold and finally classified according to LogEig and fully-connected Euclidean layers. Furthermore, the U-SPDNet introduces ExpEig layers to compute Riemannian barycenters $\mathcal{P}$ from their Euclidean counterparts $\tilde{\mathbf{A}}$ within the LFE modules. Therefore, Wang et al. (2022b) propose a methodology for performing Riemannian matrix backpropagation to incorporate this additional layer.

The technical details of these architectural and Riemannian backpropagation learning methodology augmentations are found below, with the methodology maintaining the notation employed by Wang et al. (2022b). Please refer to Appendix C to align symbolism with that of Huang and Gool (2016)'s used above.

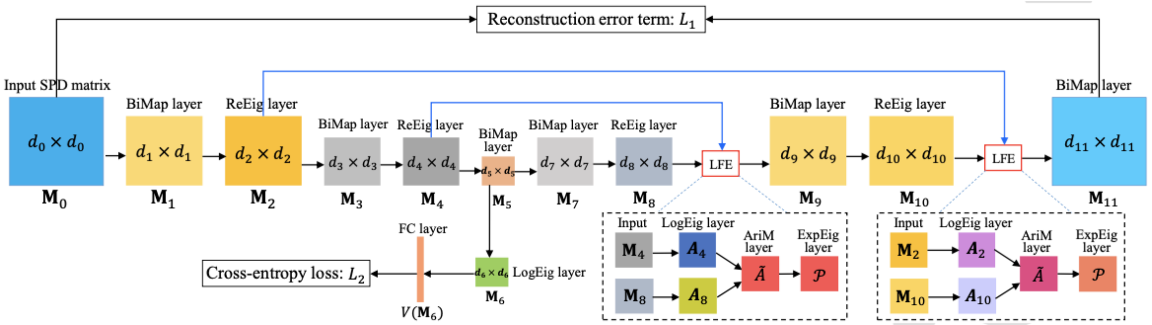


**Figure 2:** A simplified illustration of U-SPDNet as proposed by Wang et al. (2022b). The model backbone comprises BiMap and ReEig layers as in Figure 1. Log-Fusion-Exp (LFE) shortcut connection modules are included, exploiting both the Euclidean and Riemannian barycenters, $\tilde{\mathbf{A}}$ and $\mathcal{P}$ for information up-sampling. The reconstruction error term (RET), $L_1$, or MSE, is introduced to encourage one-to-one mapping between the input and output SPD matrices.

### 3.2.1 LFE Modules & Barycenter Learning

In order to realise SPD backpropagation in conjunction with the introduction of shortcut connections, three auxiliary layers are introduced which comprise elements of the original SPDNet backbone model blocks. This information fusion technique is implemented specifically by construction of a Log-Fusion-Exp (LFE) module. First, the up-sampled matrices from the encoder and decoded matrices are mapped separately to the Euclidean space by virtue of the LogEig layer as in Equation 67 (matrices $\mathbf{M}_2$, $\mathbf{M}_4$ and $\mathbf{M}_8$, $\mathbf{M}_{10}$, respectively, as depicted in Figure 2). Subsequently the arithmetic mean of the matrices output by these two different ReEig layers produces the Euclidean barycenter $\tilde{\mathbf{A}}$. Lastly, $\tilde{\mathbf{A}}$ is re-embedded on the SPD manifold via the ExpEig layer as the Riemannian barycenter $\mathcal{P}$.

As in previous works, the Riemannian barycenter is leveraged in order to pool feature representations, with the additional module allowing for the necessary forward and backward passes of the target Fréchet mean $\mathcal{P}$ as in Figure 2. This approach is motivated by previous work using the Riemannian barycenter for feature map merging (Brooks et al., 2019a). The mean, or median described in Equation 68 is applied to the set of matrices $\{\mathbf{M}^h\}_{h=1}^{\mathcal{H}}$ with respect to the LEM detailed in Equation 65. The resulting Fréchet mean is defined over the SPD Manifold.[1]

$$\mathcal{P}^* = \underset{\mathcal{P} \in \mathcal{S}_{++}^{d*}}{\operatorname{argmin}} \sum_{h=1}^{\mathcal{H}} \delta_{LEM}(\mathbf{M}, \mathcal{P}), \tag{6}$$

Here, $\mathcal{H} = H_\alpha + H_k$ and $\mathcal{P}$ is the target Fréchet mean. $\{\mathbf{M}^h\}_{h=1}^{\mathcal{H}}$ is comprised of the sum of the up- and down- stream SPD matrix outputs of different ReEig layers (matrices $\mathbf{M}_2$, $\mathbf{M}_4$ and $\mathbf{M}_8$, $\mathbf{M}_{10}$, respectively, as depicted in Figure 2). Wang et al. (2022b) detail the specific proof to find the Riemannian barycenter *i.e.* the Fréchet mean $\mathcal{P}^*$ and is given in their Equation 10.

---

[1] The SPD Manifold is $\mathcal{S}_{++}^d := \{\mathbf{M} \in \mathbb{R}^{d \times d} : \mathbf{M} = \mathbf{M}^T, \mathbf{v}^T\mathbf{M}\mathbf{v} > 0, \forall \mathbf{v} \in \mathbb{R}^d \ /\{0_d\}\}$ following Wang et al. (2022b). The reason this is formally done is to ensure that this forms a group of $d \times d$ SPD matrices as a commutative Lie group. The resulting manifold structure can then exploit the manifold's geodesic and metric structure, and its tangent space can be described using the Log-Euclidean Metric. This allows a mapping in an LogEig layer from the input space into the tangent space, and then an inverse map from the tangent space back into the output space in an associated ExpEig layer. The notational differences in the key literature are given in Appendix C.

### 3.2.2 Reconstruction Error Term

The updated design of the objective function is key to achieving an improved classification by manifold-to-manifold network embedding of embedded information structures, as captured by financial markets correlation matrices. Specifically, Wang et al. (2022b) propose a solution to the information degradation issue experienced by the SPDNet and SPDNetBN models by introducing a loss function that combines the CEL used in these baseline models with a reconstruction error term. Minimising this loss function aims to increase sensitivity to deviations in the reconstructed matrices away from their input counterparts which enforces effective representation learning (Wang et al., 2022b). This loss function is:

$$L(\theta_3, \phi; \mathcal{M}) = \lambda \sum_{i=1}^{N} L_1(\mathbf{M}_i, \hat{\mathbf{M}}_i) + \sum_{i=1}^{N} L_2(\mathbf{M}_i, c_i), \tag{7}$$

for $N$ matrices, where $\theta_3 = \{\theta_1, \theta_2, \mathcal{Z}\}$, $\mathcal{Z}$ is the projection of the FC layer for classification learning, and $\lambda$ is the learning rate. The first term, $L_1$, is the RET which acts as the supervisor of the multi-layer information compression and reconstruction at work within the U-SPDNet model to improve the reconstruction of structural information provided by input SPD matrices. The $L_1$ component of the objective function is given as:

$$L_1(\theta_1, \theta_2, \phi; \mathcal{M}) = \sum_{i=1}^{N} \left\| \mathbf{M}_i - \hat{\mathbf{M}}_i \right\|_F^2, \tag{8}$$

where $\hat{\mathbf{M}}_i$ represents the reconstructed input SPD matrix data which resides on manifold $\mathcal{M}$. Practically, this LEM-based distance computation, rooted in the formulation of Equation 67, is replaced by its Euclidean counterpart in the interest of computational efficiency achieved by avoiding the inversion of $\mathbf{M}_i$ during optimisation. The Euclidean metric meets sufficient conditions for measuring a statistically-intuitive similarity between input and reconstructed SPD correlation matrices, as with the Fisher-Rao distance in Equation 64. Thus it is utilised experimentally analogously to the MSE loss which is popular in the context of vanilla AE training.

The second term of the objective function to be minimised is the CEL as used in the experiments of Huang and Gool (2016) and Brooks et al. (2019a) for training of the SPDNet and SPDNetBN models, respectively. This function minimises the distance between the input-target pairs $(\mathbf{M}_i, c_i)$ for $i \in [1, N]$, where $c_i$ is the output of the LogEig and FC layer module. Specifically, the CEL takes the form of:

$$L_2(\mathbf{M}_i, c_i) = -\sum_{\gamma=1}^{G} \eta(c_i, \gamma) \times log \frac{e^{\mathcal{Z}^{\gamma} V(\mathbf{M}_{\mathbb{K}_e} + 1)}}{\sum_{\sigma} e^{\mathcal{Z}^{\sigma} V(\mathbf{M}_{\mathbb{K}_e} + 1)}}, \tag{9}$$

where $V(\mathbf{M}_{\mathbb{K}_e+1})$ is the vectorised forms of $\mathbf{M}_{\mathbb{K}_e+1}$, $\mathcal{Z}^{\gamma}$ denotes the $\gamma^{\text{th}}$ row of the projection matrix $\mathcal{Z}$ and $\eta(c_i, \gamma)$ is a binary pointer function with:

$$\eta(c_i, \gamma) = \begin{cases} 1, & c_i = \gamma, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

### 3.2.3 SPD Matrix Decoding

While the ReEig layer methodology remains identical to that of the Huang and Gool (2016) formulation in the U-SPDNet decompression channel, the BiMap layers must be reformulated as:

$$\mathbf{M}_\alpha = f_b^{(\alpha)}(\mathbf{W}_{\alpha-1}, \mathbf{W}_\alpha) = \mathbf{W}_\alpha \mathbf{M}_{\alpha-1} \mathbf{W}_\alpha^T, \tag{11}$$

where $\mathbf{M}_{\alpha-1} = f_r^{(\alpha-1)}(\mathbf{M}_{\alpha-2})$, $\mathbf{M}_{\alpha-1} \in \mathbf{S}^+_{d_{\alpha-1}}$ and $f_r$ refers to the ReEig function for layer $\alpha$. As in Equation 1, the weight matrices are confined to the Steifel manifold (Huang and Gool, 2016).

### 3.2.4 ExpEig Layer

Here, the ExpEig layer is described as:

$$\mathbf{M}_k = f_e(\mathbf{M}_{k-1}) = \exp(\mathbf{M}_{k-1}) = \mathbf{U} \exp(\mathbf{\Sigma}) \mathbf{U}^T, \tag{12}$$

where $k$ represents any given layer of the network. $\mathbf{U}$ and $\mathbf{\Sigma}$ are the eigenvalue and eigenvector matrices, respectively.

## 3.3 Riemannian Matrix Backpropagation

### 3.3.1 Motivations

Deep network training may be executed employing a number of methods which rely on the application of the chain rule to compute the gradient of a target function, a technique referred to as backpropagation. SGD is one such method. The classical formulation of SGD in the Euclidean space may be extended to the Riemannian manifold for the SPDNet model. ConvNet models are typically formulated as a series of composite functions $f = f^{(l)} \circ f^{(l-1)} \circ ... f^{(1)}$ over a total of $l$ layers. This structure has an associated weight space comprising the set of $l$ connection weight transformation matrices $\mathbf{W} = (\mathbf{W}_l, \mathbf{W}_{l-1}, ..., \mathbf{W}_1)$. The $k$-th layer has an associated loss function $\mathcal{L}^{(k)} = l \circ f^{(l)} \circ ... f^{(k)}$.

backpropagation is applied to this loss function at each layer $k \in [1, l]$ to compute the gradient with respect to the weight $\mathbf{W}_k$ and input data $\mathbf{X}_{k-1}$, respectively. Classically these chain rule functions are vector functions in $\mathbb{R}^n$ as used in Graph Transformer Network (GTN)s (Lecun et al., 1998).

$$\frac{\partial \mathcal{L}^{(k)}(\mathbf{X}_{k-1}, y)}{\partial \mathbf{W}_k} = \frac{\partial \mathcal{L}^{(k+1)}(\mathbf{X}_k, y)}{\partial \mathbf{X}_k} \frac{\partial f^{(k)}(\mathbf{X}_{k-1})}{\partial \mathbf{W}_k}, \tag{13}$$

$$\frac{\partial \mathcal{L}^{(k)}(\mathbf{X}_{k-1}, y)}{\partial \mathbf{X}_{k-1}} = \frac{\partial \mathcal{L}^{(k+1)}(\mathbf{X}_k, y)}{\partial \mathbf{X}_k} \frac{\partial f^{(k)}(\mathbf{X}_{k-1})}{\partial \mathbf{X}_{k-1}}, \tag{14}$$

where $y$ is the gradient output and $\mathbf{X}_k = f^{(k)}(\mathbf{X}_{k-1})$.

Two broad difficulties associated with learning an SPDNet lie in adapting the classical backpropagation technique to structured Riemannian functions and in manifold-constrained optimisation (Edelman et al., 2006; Huang and Gool, 2016). These SPD manifold constraint difficulties are identified explicitly by Huang and Gool (2016) as:

1. Optimising the weight space in the BiMap layers by using the Euclidean gradient as computed in Equation 13 will yield invalid orthogonal weights denoted as $\nabla \mathcal{L}^{(k)}_{\mathbf{W}_k}$ since these weights are constrained to the Stiefel manifold (Huang and Gool, 2016).
2. The chain rule identified in Equation 14 is not compatible with the structured eigendecomposition functions employed in the ReEig and LogEig layers (Huang and Gool, 2016).

Solutions to these problems employs the generalised matrix manipulation frameworks proposed by Ionescu et al. (2016), first employed by Huang and Gool (2016) without RBN, and Edelman et al. (2006). The simplicity and scalability of the standard backpropagation framework are used invas the scaffolding for SPD matrix learning. These solutions are detailed in the following.

### 3.3.2 BiMap Learning

SGD may be extended to the setting of constrained manifold optimisation. The weight update is accomplished in a similar fashion as for the parameter matrix $G$, detailed below, where the Riemannian gradient $\tilde{\nabla} \mathcal{L}^{(k)}_{\mathbf{W}_k}$ is found by subtracting the normal component of the Euclidean gradient (Brooks et al., 2019a). The result is then taken as the steepest tangential descent direction along which the update may be computed in the tangent space of the Steifel manifold and subsequently mapped back by the retraction operation $\Gamma$ as expounded on in Edelman et al. (2006) and Absil et al. (2010).

More specifically, the Stiefel manifold-constrained weight update respects the differential update system described by (employing the notation of Huang and Gool (2016)):

$$\tilde{\nabla} \mathcal{L}^{(k)}_{\mathbf{W}^t_k} = \nabla \mathcal{L}^{(k)}_{\mathbf{W}^t_k} - \nabla \mathcal{L}^{(k)}_{\mathbf{W}^t_k} (\mathbf{W}^t_k)^T \mathbf{W}^t_k, \tag{15}$$

where $\nabla \mathcal{L}^{(k)}_{\mathbf{W}^t_k} (\mathbf{W}^t_k)^T \mathbf{W}^t_k$ is the Euclidean component of the gradient from Equation 13:

$$\nabla \mathcal{L}^{(k)}_{\mathbf{W}^t_k} = 2 \frac{\partial \mathcal{L}^{(k+1)}}{\partial \mathbf{X}_k} \mathbf{W}^t_k \mathbf{W}_{k-1}. \tag{16}$$

Using Equation 15 the weight update equation is then:

$$\mathbf{W}^{t+1}_k = \Gamma(\mathbf{W}^t_k) - \lambda \tilde{\nabla} \mathcal{L}^{(k)}_{\mathbf{W}^t_k}, \tag{17}$$

where $\lambda$ is the learning rate and $\Gamma$ is the retraction function described in Equation 66.

### 3.3.3 ReEig and LogEig Learning

To tackle problem 2 Huang and Gool (2016) use the matrix backpropagation generalisation proposed by Ionescu et al. (2016) for learning the ReEig and LogEig layers.

Here $\mathcal{F}$ is a function describing variable variations in upper layers with respect to those in previous layers; specifically $d\mathbf{X}_k = \mathcal{F}(\mathbf{X}_{k-1})$. This non-linear matrix function may be integrated with the classical chain rule identified in Equation 14 to define matrix backpropagation as:

$$\frac{\partial L^{(k)}(\mathbf{X}_{k-1}, y)}{\partial \mathbf{X}_{k-1}} = \mathcal{F}^* \left( \frac{\partial L^{(k+1)}(\mathbf{X}_k, y)}{\partial \mathbf{X}_k} \right), \tag{18}$$

where $\mathcal{F}^*$ is a non-linear adjoint operator of $\mathcal{F}$. Since both the LogEig and ReEig layers involve eigendecomposition, $\mathbf{X}_{k-1} = \mathbf{U}_{k-1}\mathbf{\Sigma}_{k-1}\mathbf{U}_{k-1}^T$, an interim virtual layer $k^{\triangle}$ is introduced, as in the work of Wang et al. (2022b) for learning of the ExpEig layer in the U-SPDNet.

Equation 18 may be applied to derive an update rule for this new layer as follows (layer sub-scripts are removed for readability):

$$\begin{aligned}
\frac{\partial L^{(k)}}{\partial \mathbf{X}_{k-1}} : d\mathbf{X}_{k-1} &= \mathcal{F}^* \left( \frac{\partial L^{(k^{\triangle})}}{\partial \mathbf{U}} \right) : d\mathbf{X}_{k-1} + \mathcal{F}^* \left( \frac{\partial L^{(k^{\triangle})}}{\partial \mathbf{\Sigma}} \right) : d\mathbf{X}_{k-1} && (19)\\
&= \frac{\partial L^{(k^{\triangle})}}{\partial \mathbf{U}} : \mathcal{F}(d\mathbf{X}_{k-1}) + \frac{\partial L^{(k^{\triangle})}}{\partial \mathbf{\Sigma}} : \mathcal{F}(d\mathbf{X}_{k-1}) && (20)\\
&= \frac{\partial L^{(k^{\triangle})}}{\partial \mathbf{U}} : d\mathbf{U} + \frac{\partial L^{(k^{\triangle})}}{\partial \mathbf{\Sigma}} : d\mathbf{\Sigma}, && (21)
\end{aligned}$$

where $d\mathbf{U}$ and $\mathbf{\Sigma}$, the eigenspace and eigenvectors of the SPD matrix are found as in equations 31 and 32 below. Applying substitution of equations 19 and 21 into Equation 18 allows for the partial derivatives of the loss function in the LogEig and ReEig layers to be obtained as follows:

$$\frac{\partial L^{(k)}}{\partial \mathbf{X}_{k-1}} = 2\mathbf{U}(\mathbf{\Lambda} \circ (\mathbf{U}^T\mathbf{\Omega}_1)_{sym} + \mathbf{U}\mathbf{\Omega}_2\mathbf{U}^T, \tag{22}$$

with $\lambda$, $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ defined as in Equation 22.

Since the ReEig and LogEig layer operations exist as in equations 2 and 4 respectively, their variations take the forms of:

$$\begin{aligned}
d\mathbf{X}_k &= 2(d\mathbf{U}\max(\epsilon\mathbf{I}, \mathbf{\Sigma})\mathbf{U}^T)_{sym} + (\mathbf{U}\mathbf{Q}d\mathbf{\Sigma}\mathbf{U}^T)_{sym}; && (23)\\
d\mathbf{X}_k &= 2(d\mathbf{U}\log(\mathbf{\Sigma})\mathbf{U}^T)_{sym} + (\mathbf{U}\mathbf{\Sigma}^{-1}d\mathbf{\Sigma}\mathbf{U}^T)_{sym}, && (24)
\end{aligned}$$

where $\max(\epsilon\mathbf{I}, \mathbf{\Sigma})$ takes the form of Equation 3.

Therefore, in the introduced virtual layer $k^{\triangle}$, the partial derivatives of the loss function for the ReEig and LogEig layers with respect to the SPD matrix eigenspace $\mathbf{U}$ and eigenvalue $\mathbf{\Sigma}$ matrices are derived as:

$$\frac{\partial L^{(k^{\triangle})}}{\partial \mathbf{U}} = 2\left( \frac{\partial L^{(k+1)}}{\partial \mathbf{X}_k} \right)_{sym} \mathbf{U}\max(\epsilon\mathbf{I}, \mathbf{\Sigma}) \tag{25}$$

$$\frac{\partial L^{(k^{\triangle})}}{\partial \mathbf{\Sigma}} = \mathbf{Q}\mathbf{U}^T \left( \frac{\partial L^{(k+1)}}{\partial \mathbf{X}_k} \right)_{sym} \mathbf{U}, \tag{26}$$

where $\mathbf{Q}$ is the gradient of $\max(\epsilon \mathbf{I}, \mathbf{\Sigma})$ with diagonal elements found as:

$$\mathbf{Q}_{(i,i)} = \begin{cases} 1, & \mathbf{\Sigma}(i,i) > \epsilon, \\ 0 & \mathbf{\Sigma}(i,i) \leq \epsilon, \end{cases} \tag{27}$$

in the case of the LogEig layer.

In the case of the ReEig layer the partial derivatives are defined as:

$$\frac{\partial L^{(k^{\triangle})}}{\partial \mathbf{U}} = 2 \left( \frac{\partial L^{(k+1)}}{\partial \mathbf{X}_k} \right)_{sym} \mathbf{U} \log(\mathbf{\Sigma}) \tag{28}$$

$$\frac{\partial L^{(k^{\triangle})}}{\partial \mathbf{\Sigma}} = \mathbf{\Sigma}^{-1}\mathbf{U}^T \left( \frac{\partial L^{(k+1)}}{\partial \mathbf{X}_k} \right)_{sym} \mathbf{U}. \tag{29}$$

### 3.3.4 ExpEig Learning

The gradient computations of the BiMap, LogEig and ReEig layers in the U-SPDNet model are identical to those proposed by Huang and Gool (2016). The ExpEig layer used by Wang et al. (2022b) requires a tailored Riemannian backpropagation strategy. The ExpEig layer may be separated into two components. Singular value decomposition is applied to the first. To execute this, an intermediate layer $k^{\triangle}$ is introduced where the input matrix $\mathbf{M}_{k-1}$ is decomposed into eigenvector and eigenvalue matrices $\mathbf{U}$ and $\mathbf{\Sigma}$ respectively such that $\mathbf{M}_{k^{\triangle}} = (\mathbf{U}, \mathbf{\Sigma})$. The chain rule as in Equation 14 is derived for the interim layer as:

$$\frac{\partial L^{(k)}}{\partial \mathbf{M}_{k-1}} : d\mathbf{M}_{k-1} = \frac{\partial L^{(k^{\triangle})}}{\partial \mathbf{U}} : d\mathbf{U} + \frac{\partial L^{(k^{\triangle})}}{\partial \mathbf{\Sigma}} : d\mathbf{\Sigma}. \tag{30}$$

The computations of $d\mathbf{U}$ and $d\mathbf{\Sigma}$ may be found at (Wang et al., 2022b; Huang and Gool, 2016; Ionescu et al., 2016):

$$d\mathbf{U} = 2\mathbf{U}(\mathbf{\Lambda}^T \circ (\mathbf{\Sigma}^T\mathbf{U}^T d\mathbf{M}_{k-1}\mathbf{U})_{sym} \tag{31}$$

$$d\mathbf{\Sigma} = (\mathbf{U}^T d\mathbf{M}_{k-1}\mathbf{U})_{diag}, \tag{32}$$

where $\circ$ is the Hadamard product, and *sym* and *diag* mean that these elements are symmetric and have off-diagonal elements of value 0, respectively. By introducing equations 31 and 32 to Equation 30, the partial derivative may be reformulated as:

$$\frac{\partial L^{(k)}}{\partial \mathbf{M}_{k-1}} = 2\mathbf{U}(\mathbf{\Lambda} \circ (\mathbf{U}^T\mathbf{\Omega}_1)_{sym} + \mathbf{U}\mathbf{\Omega}_2\mathbf{U}^T, \tag{33}$$

where $\mathbf{\Omega}_1 = \frac{\partial L^{(k^{\triangle})}}{\partial \mathbf{U}}$ and $\mathbf{\Omega}_2 = (\frac{\partial L^{(k^{\triangle})}}{\partial \mathbf{\Sigma}})_{diag}$. Finally, $\mathbf{\Lambda}$ is written as:

$$\mathbf{\Lambda}_{ij} = \begin{cases} \frac{1}{\pi_i^2 - \pi_j^2}, & i \neq j, \\ 0, & i = j, \end{cases} \tag{34}$$

where $\pi_i$ represents the $i^{\text{th}}$ eigenvalue of $\mathbf{\Sigma}$.

The second component of the ExpEig layer is dealt with as follows: $\mathbf{U}$ and $\mathbf{\Sigma}$ are inputs to the $k^{\text{th}}$ layer from layer $k^{\triangle}$. This output is written as $\mathbf{M}_k = \mathbf{U}\exp(\mathbf{\Sigma})\mathbf{U}^T$. As above, by applying chain rule, the partial derivatives for the variation of $\mathbf{M}_k$ are deduced (Wang et al., 2022b):

$$\frac{\partial L^{(k)}}{\partial \mathbf{U}} = 2\left(\frac{\partial L^{(k+1)}}{\partial \mathbf{M}_k}\right)_{sym} \mathbf{U}\exp(\mathbf{\Sigma}) \tag{35}$$

$$\frac{\partial L^{(k)}}{\partial \mathbf{\Sigma}} = \exp(\mathbf{\Sigma})\mathbf{U}^T \left(\frac{\partial L^{(k+1)}}{\partial \mathbf{M}_k}\right)_{sym} \mathbf{U}. \tag{36}$$

## 3.4 Riemannian Batch-Normalised Backpropagation

Brooks et al. (2019a) introduce independent RBN layers to the seminal SPDNet model proposed by Huang and Gool (2016) in the SPDNetBN. The use of the Riemannian barycenter necessitates use of two results as in Section 3.4.2 and Section 3.4.3 in order to fit the learning of the RBN to the existing Riemannian backpropagation techniques.

### 3.4.1 Motivations

As in the work of Huang and Gool (2016), there are problems inherent to classical Euclidean backpropagation algorithms as applied to the SPD matrix manifold in the context of barycenter learning:

1. Updating the bias parameter matrix $G$ in the RBN layer using the Euclidean gradient defined as $\frac{\partial \mathcal{L}^{(k)}(\mathbf{X}_{k-1},y)}{\partial G}$ and denoted $\nabla \mathcal{L}_G$ will not respect its constructed constraint on the SPD manifold (Brooks et al., 2019a).

2. The optimisation of the batch normalisation layer involves structured matrix functions of $G$ in addition to $\mathfrak{G}$; specifically $G^{\frac{1}{2}}$ and $\mathfrak{G}^{\frac{1}{2}}$, meaning that the Euclidean gradient with respect to the input covariance matrices will lead to incorrect backpropagation by not accounting for the non-linear effects these functions have on the matrices' eigenvalues (Brooks et al., 2019a).

### 3.4.2 Bias Parameter Learning

The parameterised bias matrix $G$ is constrained to the manifold by construction. Thus the Euclidean gradient $\nabla \mathcal{L}_G$ must necessarily be extended to the Riemannian version to respect this constraint whilst accounting for the system's loss function $\mathcal{L}$. This two-part update process is detailed explicitly by Edelman et al. (2006), illustrated in Figure 3 below, and involves the following:

1. The Euclidean gradient of the network loss function with respect to the bias parameter matrix, $\frac{\partial \mathcal{L}}{\partial G}$, or $\partial G_{eucl}$, is projected to the tangent space of the manifold $\mathcal{M}$ at $G$ using the projection operator $\Gamma\mathcal{T}_G$ described in Equation 37 below.

2. The gradient update is then found using the Exp mapping in Equation 66 to compute the geodesic at point $G$ on the manifold in the direction of the projected Riemannian gradient $\partial G_{riem}$.

The projection operator particular to $G$ employed here is found as:

$$\forall X, \Pi\mathcal{T}_G(X) = G\frac{X + X^T}{2}G \in \mathcal{T}_G \subset \mathcal{S}^+. \tag{37}$$

Thus the parameter matrix $G$ is learnt in the Steifel manifold-constrained manner. Note that no layer $k^{\triangle}$ is introduced here since the batch normalisation is independently appended subsequent to each BiMap layer.

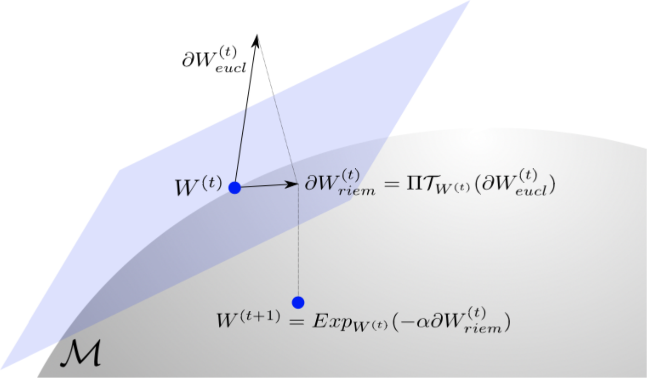


**Figure 3:** Illustration of Edelman et al. (2006)'s manifold-constrained learning of the $G$ parameter. The Euclidean gradient is projected onto the tangent space and then mapped onto the (Stiefel) manifold $\mathcal{M}$.

### 3.4.3 Structured Matrix Backpropagation

The square root $(\cdot)^{\frac{1}{2}}$ and inverse square root $(\cdot)^{-\frac{1}{2}}$ functions applied to $G$ and $\mathfrak{G}$, respectively, are structured symmetric matrix functions known in $\mathcal{S}^+_*$. The chain rule must be generalised to the case where data resides on the Riemannian manifold, and where an RBN layer has been introduced, making use of these functions.

Both $(\cdot)^{\frac{1}{2}}$ and $(\cdot)^{-\frac{1}{2}}$ are considered monotonous non-linear functions $f$. Ionescu et al. (2016) develops a general formula for the gradient of $f$. Brooks et al. (2019a) refer to Engin et al. (2018) as showing equivalence of this formulation and that of 'soviet mathematics' - the simpler version lets $X$, the output matrix with succeeding gradient, be the eigendecomposition function $f$ of input matrix $X$, i.e. for a function $X^{(l)} \longrightarrow X^{(l+1)} = f(X^{(l)})$, the output gradient is computed as:

$$\frac{\partial L^{(l)}}{\partial X^{(l)}} = U\left(L \odot (U^T \left(\frac{\partial L^{(l+1)}}{\partial X^{(l+1)}}\right) U)\right) U^T, \tag{38}$$

for input gradient $\frac{\partial L^{(l+1)}}{\partial X}$ at layer $l$. Here, $U$ is the eigenspace of matrix $P$ and $L$ is the Loewner matrix defined by:

$$L_{ij} = \begin{cases} \frac{f(\sigma_i)-f(\sigma_j)}{\sigma_i-\sigma_j}, & \sigma_i \neq \sigma_j, \\ \\ f'(\sigma_i) & \text{otherwise}, \end{cases} \tag{39}$$

where $(\sigma_i)_{i\leq n}$ are the SPD matrix' eigenvalues grouped in the diagonal of the eigenvalue matrix $\Sigma$.

As in the work of Engin et al. (2018), $\left((\cdot)^{-\frac{1}{2}}\right)' = -\frac{1}{2}(\cdot)^{-\frac{3}{2}}$ and $\left((\cdot)^{\frac{1}{2}}\right)' = \frac{1}{2}(\cdot)^{\frac{1}{2}}$.

# 4 Data

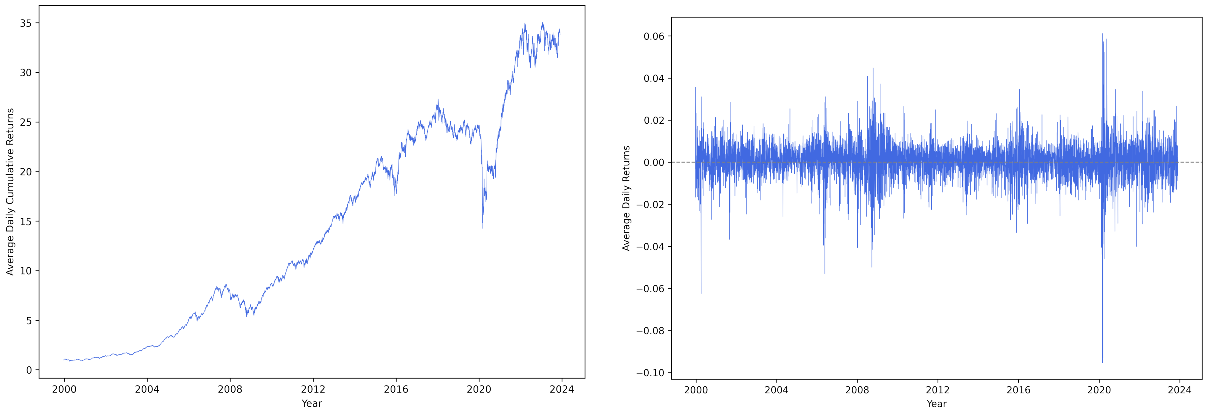
Model evaluation is performed using real JSE Top 60 equity correlation matrices alongside synthetically generated block hierarchical correlation matrices. These two datasets provide both real-world and counterfactual scenarios, which means that the temporal component of the mechanism driving financial markets through varying regimes may be isolated from the emergent hierarchical structure present in differing phases. Comparative analysis allows for the determination of whether the models successfully learn a latent statistical feature representation of the market structure by accounting for both information lag effects and the emergence of hierarchical structures.

## 4.1 JSE Top 60 Data

### 4.1.1 Exploratory Data Analysis

To conduct a real-world application of the selected manifold learning techniques for regime classification, the set of daily equity returns of the top 60 high-capitalised stocks traded on the JSE over the period January 2000 - December 2023 will be examined. Johannesburg All-Share Index (JALSH) price data are accessed via Bloomberg (BBG) which are then filtered according to a market capitalisation-determined ranking (see data processing code at: Orton and Gebbie (2024)).

The JSE Top 60 stocks comprise an index of the most-traded stocks in the South African financial market. Figure 4 below provides a visual summary of average price action of the stocks comprising the data set over the considered period. The subfigures 8a and 8b show the average cumulative daily returns and average daily returns, respectively. Considered through the lens of market regime detection, it is observed that the market passes through several significantly different structural periods characterised by particular price volatility dynamics. Specifically, stressed periods seen in 2007-2008 (Global Financial Crisis (GFC)), 2015-2016 and March 2020 (Covid-19) characterised by market downtrends as in Figure 8a correspond with greater volatility periods in average price action away from 0 in Figure 8b. Thus the statistically significant relationships identified by Ang and Bekaert (2004), where bear markets are characterised by increased volatility are confirmed to be at work in the context of the South African financial market. It is expected that the SPDNet, SPDNetBN and U-SPDNet models will be capable of detecting the differences between these stressed market states and their normal, or rally, counterparts which feature lower levels of price action volatility.



**(a)** JSE Top 60 average daily cumulative returns.

**(b)** JSE Top 60 average daily returns.

**Figure 4:** Here we show average market price action for the top 60 stocks listed on the JALSH index by market capitalisation for the period: January 2000 - December 2023. Subfigure 8a displays at least two periods of market stress interspersed by rally and normal periods. Subfigure 8b confirms this, displaying the market price action moving through at least two different states as characterised by high and low deviation from 0.

Figure 6 exhibits the hierarchical tree-like structure present in this financial market system as characterised by the correlation matrix, computed using the Pearson linear correlation as in Equation 49, and a distribution-free distance metric $D = \sqrt{2(1 - C)}$ where $C$ is the empirical correlation matrix. The dendrogram confirms the discoveries of Mantegna (1999b) in considering S&P 500 data; a meaningful hierarchical geometric structure underpins the dynamic evolution of this complex financial market system as evidenced by stocks falling into broad sector categories, or clusters, such as precious metal miners, industrials, retailers, energy and oil, financial services (banks and insurers), pharmaceuticals, food and beverages, telecommunications, raw materials and packaging (see Appendix A for BBG stock tickers). These classifications align closely with the GICS classifications as in Table 4.

Thus valuable economic taxonomy information is carried by these time series data alone which may be exploited by deep learning models in accounting for the SPD manifold upon which these hierarchical factor-driven returns data sit.

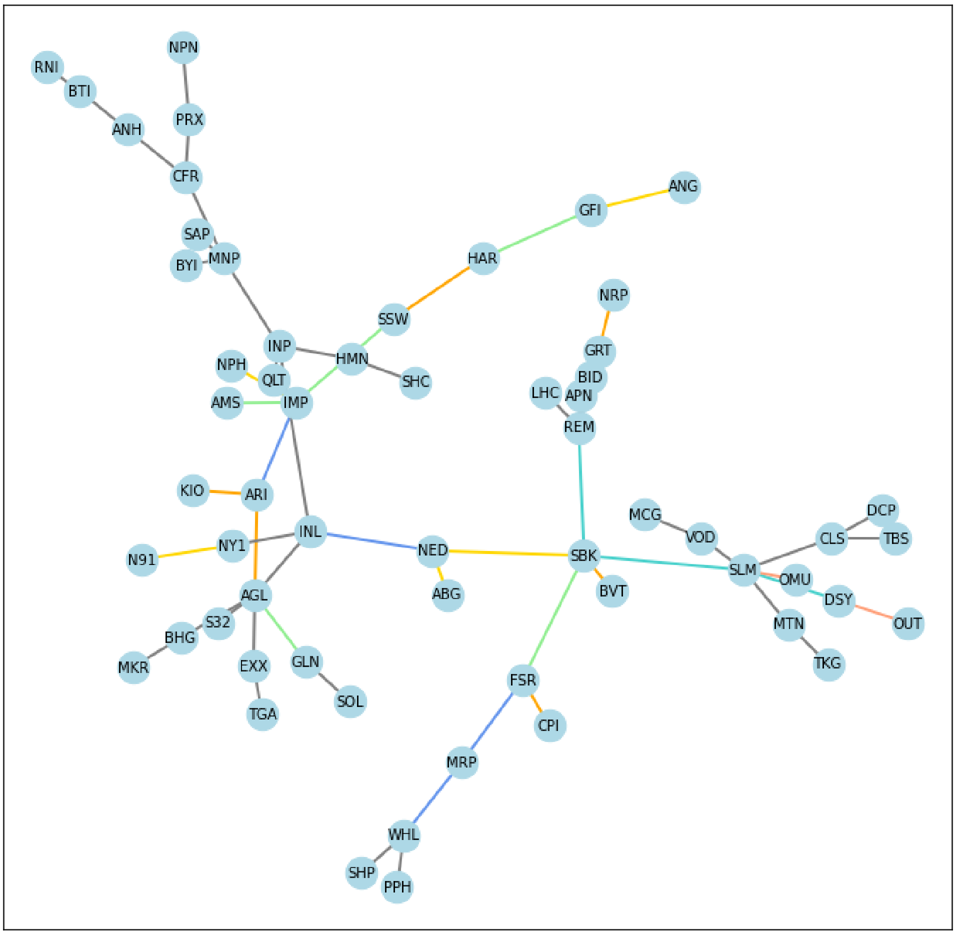


**Figure 5:** We show the minimum spanning tree of a group of 60 strongly-connected stocks traded on the JSE, ranked by market capitalisation. The stocks are labeled by their trading tickers. Data is sourced from Bloomberg (BBG) JALSH close prices over the period: January 2000 - December 2023. The colours of the edges of the MST are proportional to the distance between the stocks $i$ and $j$: gold $0.5 < D_{(i,j)} \leq 0.60$; light green $0.60 < D_{(i,j)} \leq 0.70$; turquoise $0.70 < D_{(i,j)} \leq 0.80$; pink $0.80 < D_{(i,j)} \leq 0.85$; orange $0.85 < D_{(i,j)} \leq 0.95$; dark blue $0.90 < D_{(i,j)} \leq 0.95$, where distances are computed following the metric $D = \sqrt{2(1-C)}$ where $C$ is the observed correlation matrix, as proposed by Mantegna (1999a). This tree comprises of industry groups: precious metal miners, industrials, retailers, energy and oil, financial services (banks and insurers), pharmaceuticals, food and beverages, telecommunications, raw materials and packaging.

The Single Linkage Clustering algorithm is applied to the minimum spanning tree as in the work of Mantegna (1999b) to reveal a subdominant ultrametric space characterised by a hierarchical economic taxonomy present in the stocks comprising the JSE Top 60 data. The strong connections seen in Figure 5 are further reflected in the associated dendrogram plotted in Figure 6. Several main groups and sub-groups of stocks with economic meaning are uncovered comprising: precious metal miners, raw materials, energy and oil, pharmaceuticals and healthcare, retailers, financial services, telecommunications, banks, industrials and asset managers.

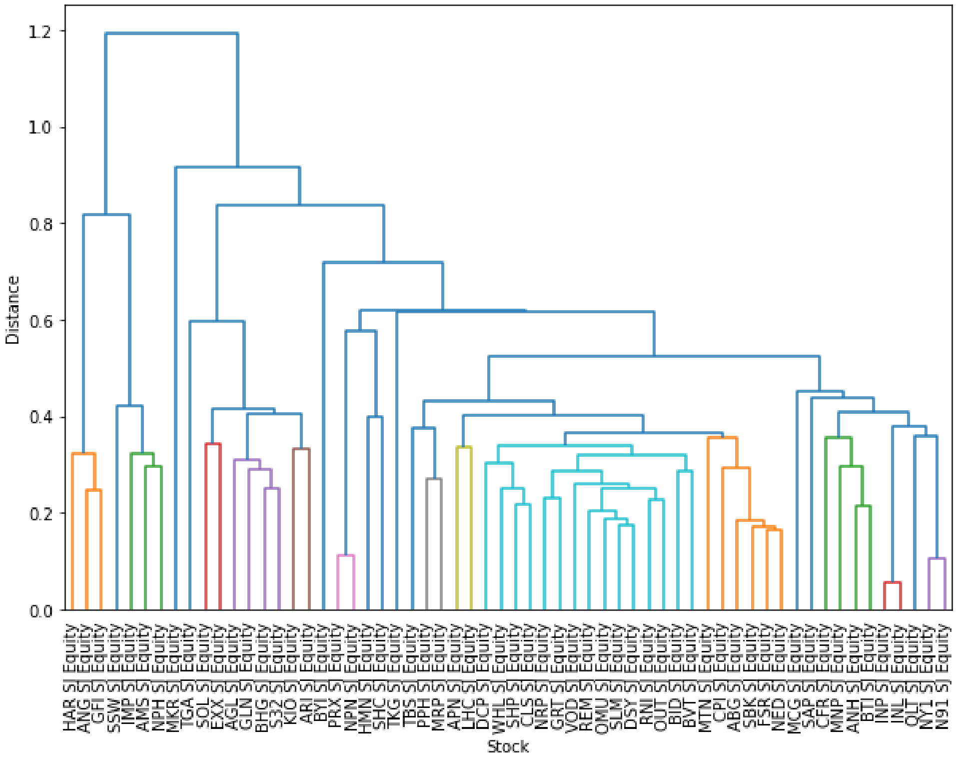


**Figure 6:** Hierarchical dendrogram associated with the MST depicted in Figure 5. This structure is found via the Single Linkage Clustering algorithm (Ward, with distance $D = \sqrt{2(1-C)}$) as in the work of Papenbrock et al. (Spring 2021)). Clusters are grouped by colour according to a distance measure being $< 0.3$. The uncovered clusters show a meaningful economic taxonomy which closely follows that shown in the MST above.

Figure 7 shows a clear block hierarchical structure through points in time. This metacorrelation heatmap is computed by flattening the matrices of the JSE Top 60 asset correlations found in a moving window of 252 days between January 2000 and December 2023, following the techniques implemented in Miori and Cucuringu (2022)'s methodology. The Pearson correlation is calculated for each pair of data points in time. The associated heatmap plot depicts a minimum of three hierarchies in time.

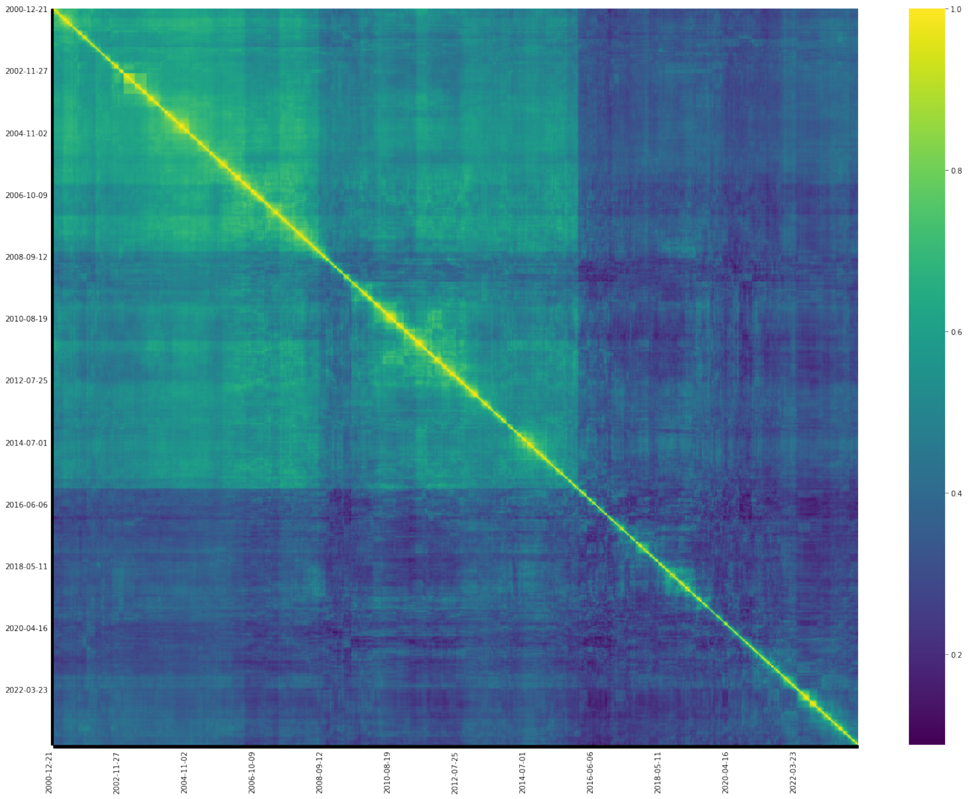


**Figure 7:** Similarity of points in time is measured by the metacorrelation computed from the Pearson correlation found in a sliding window of 252 days between January 2000 and December 2023. There is a clear block hierarchical structure present, with there being diversification in the level of similarity between the different considered points in time (Miori and Cucuringu, 2022).

This empirical validation of the presence of a consistent hierarchical structure providing the under-girding of market evolution through different regimes provides further motivation for deep representation learning respecting this structure in improving regime estimation.

### 4.1.2 Data Labeling

> Researchers need to ponder very carefully how they define labels, because labels determine the task that the algorithm is going to learn.
>
> *Marcos Lopez de Prado*

Three classes of correlation matrices are identified by augmenting the original data which correspond to ’stressed’, ’normal’ and ’rally’ market regimes. A superficial method is established which marries the findings of Miori and Cucuringu (2022) and Ang and Bekaert (2004), and emulates the ’rule-of-thumb’ technique used by Marti (2021). By virtue of considering that three broad market regime categories have been detected in equity markets, and that the annualised SR of indices belonging to multiple asset classes differ significantly amongst six data-informed regimes, a categorisation system emerges for correlation matrix labelling.

The *ex ante* SR is found as:

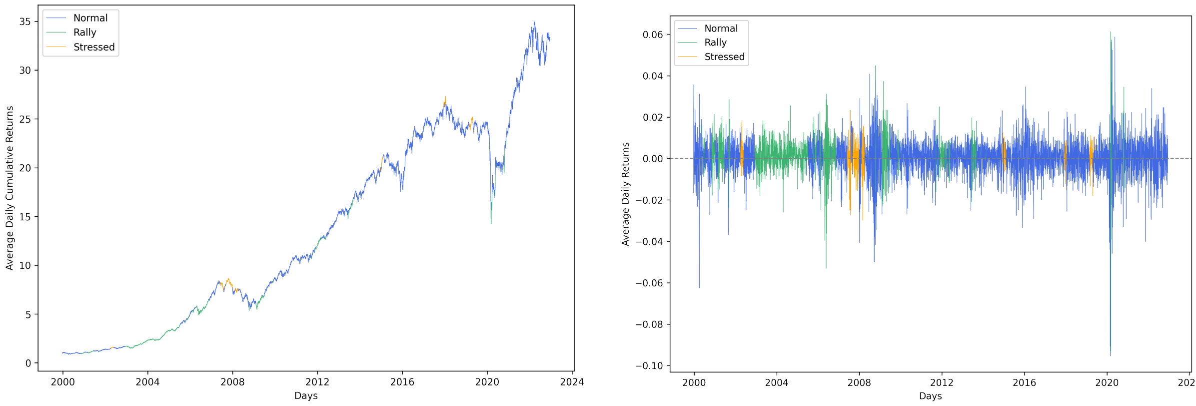
$$SR = \frac{\mathbb{E}[R_X - R_{rf}]}{\sigma_X}, \tag{40}$$

where $R_X$ and $R_{rf}$ are the asset $X$ and risk-free returns time series respectively, and $\sigma_X$ is the contemporaneous volatility of feature $X$.

The 60×60 correlation matrices computed on the relevant equity returns window are classified according to the SR as follows:

1. *Stressed market*: A market is 'stressed' whenever the market capitalisation-weighted basket of 60 stocks under consideration has an SR below -0.5;
2. *Normal market:* A market is 'normal' whenever the market capitalisation-weighted basket of 60 stocks under consideration has an SR falling between -0.5 and 2; and
3. *Rally market:* A market is 'rallying' whenever the market capitalisation-weighted basket of 60 stocks under consideration has an SR above 2.

A total of 5995 correlation matrices are computed on a 252-day rolling window basis throughout the 24-year period under consideration. The average returns and average cumulative returns are shown in Figure 8. The time series have been colour-mapped to align with stressed, normal and rally market phases using the labeling methodology described here, where the corresponding correlation matrices are categorised according to their respective Sharpe Ratio (SR).



**(a)** JSE Top 60 average daily cumulative returns.

**(b)** JSE Top 60 average daily returns.

**Figure 8:** The average JSE Top 60 returns time series are colour-encoded according to the regime labels created following the methodology as explained in Section 4.1.2. This provides a visualisation of the correlation matrix categorisation learning problem to be addressed by the SPDNet and U-SPDNet models.

Following the SR-informed labeling methodology gives rise to three regime classes characterised by large representation imbalance when applied to the JSE Top 60 returns data chronologically. Specifically, 311, 1318 and 4366 stressed, rally and normal correlation matrices are discovered. The sampled correlation coefficient distributions are depicted in Figure 9. It is identified that each class possesses a distinctive coefficient distribution, albeit with closely-spaced mean values. The class imbalance may lead to corner solutions being achieved by the implemented deep learning models. To mitigate against this, the methodology proposed by Marti

(2021) is invoked such that a random sampling algorithm produces equally-sized representative groups of data points for each class.

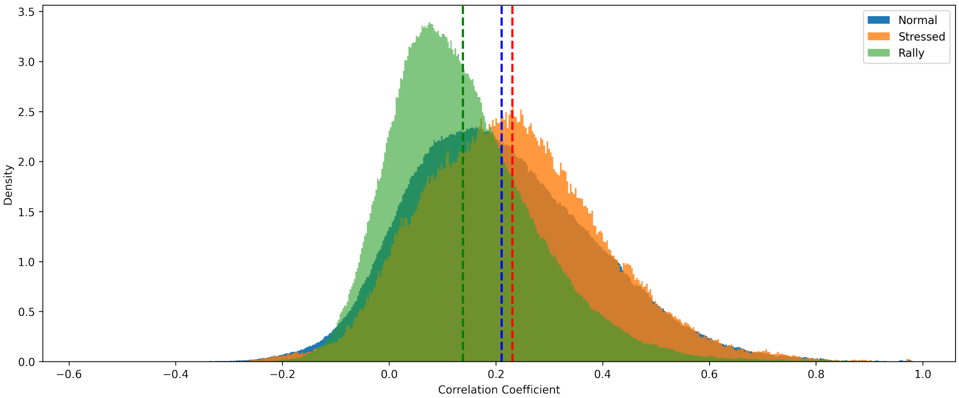
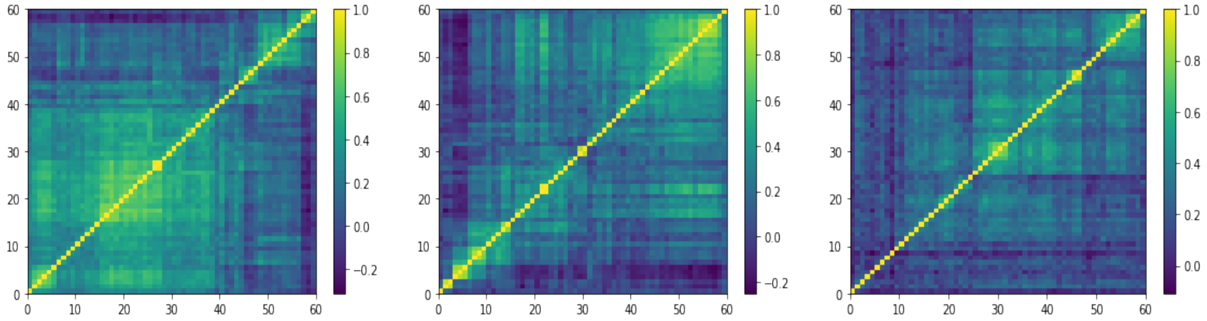


**Figure 9:** JSE Top 60 correlation coefficient density per class without resampling or regime representation imbalance correction. The 'stressed', 'normal' and 'rally' regimes have mean correlations of 0.24, 0.205 and 0.17, respectively. Average correlation and standard deviation vary per regime, supporting the findings of Miori and Cucuringu (2022) in the South African market of traded stocks, where varying underlying distributions characterise different macroeconomic regimes. Interrelatedness of assets is greater in times of market stress with greater volatility, while 'normal' and 'rally' regimes are characterised by lower average correlation and volatility. The regime correlation asymmetries identified by Ang and Bekaert (2004) prevail in the JSE top 60 data.

Each class of correlation matrices is constructed by iteratively, randomly sampling a window of 252 trading days from a market capitalisation-weighted basket of 60 stocks traded on the JSE, with labels derived from the SR computed on the same time series window. This method is used widely in data science modeling contexts for data preparation and is initially conducted to provide a comparative basis for maintaining data chronology in the context of financial markets data where information signal possesses spatio-temporal properties which should be learnt by representation learnign models to produce improved predictions. Survivorship bias is prevented by re-examining the constituents on an annual basis within the same year of study.

Finally, the correlation matrix rows and columns are hierarchically reordered according to permutations of a linkage algorithm with the Ward method and distance measure $\sqrt{2(1-C)}$, where $C$ is the correlation matrix (Miori and Cucuringu, 2022; Papenbrock et al., Spring 2021). The resulting dataset comprises 18,000 $60 \times 60$ matrices evenly divided amongst the regime categories with representative heatmaps as in Figure 10. The aim of creating 6000 matrices per regime class is motivated by the work of Marti (2020a) such that model performance is not biased by under-representation of the stressed and rally regimes categories.



**(a)** Stressed class. **(b)** Normal class. **(c)** Rally class.

**Figure 10:** We show average correlation matrices for each regime identified from the JSE Top 60 data. Distance metric: $\sqrt{2(1-C)}$, where $C$ is the sample correlation matrix. Here the stressed class is for markets with Sharpe ratios $< -0.5$ (Figure 10a), the normal market considers Sharpe ratios in $[-0.5, 2.0]$ (Figure 10b), and the rally market has Sharpe ratios $> 2.0$ (Figure 10c). In the stressed class we can see a giant cluster in the lower left as many assets have synchronised movements, as compared to the normal market condition where the block diagonal structure suggests improved diversification. The correlation density plot for each market class is shown in Figure 11.

This balanced dataset, *ceteris paribus*, exhibits an anti-correlation relationship between the average SPD matrix correlation coefficient and associated SR value (Marti, 2021), where a greater degree of variation is associated with a higher average correlation and lower SR. It is also observable in Figure 11 that, despite the semblance of three separately-sampled distributions, the three classes do not display a high degree of distributional separability, as is observed in Figure 9. It is observed that the 'stressed', 'normal' and 'rally' high-, average- and low-mean correlation matrices respectively, have average correlations of 0.24, 0.205 and 0.17. This underlying distribution system contributes to creating an interesting learning challenge for the implemented models, with the class imbalance being rectified whilst maintaining the original regime-dependent sample distribution structure.

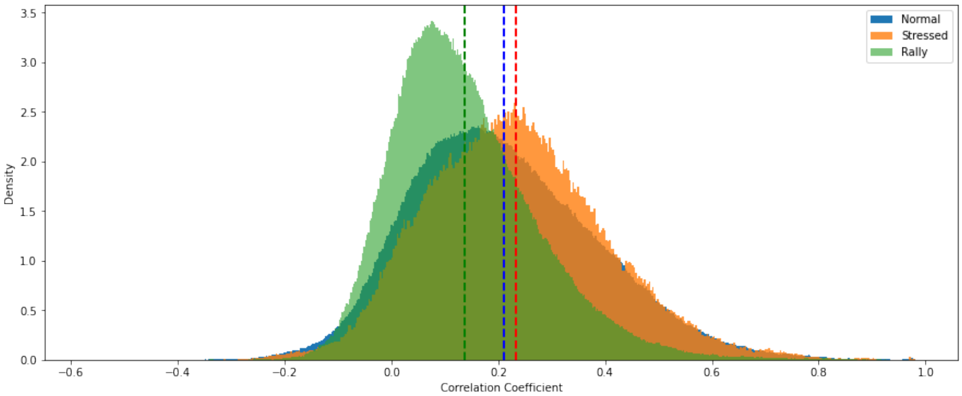


**Figure 11:** JSE Top 60 correlation coefficient density per class once class imbalance has been rectified via random sampling. The distribution profiles present in Figure 9 are maintained with the implementation of the random sampling algorithm. The average correlation matrices for each market regime are shown in Figure 10.

## 4.2 Synthetic Data

### 4.2.1 Hierarchical Nested Factor Model

In the work of Tumminello et al. (2007), the HNFM is proposed as an appropriate model for hierarchically-clustered time series generation. The time series model for the price path action of stock $i$ at time $t$ may be written as:

$$x_i(t) = \sum_{\alpha_h \in G(i)} \gamma_{\alpha_h} f^{(\alpha_h)}(t) + \eta_i \epsilon_i(t), \tag{41}$$

which is understood as a linear factor model which describes the sensitivity $\gamma_{\alpha_h}$ of stock $i$ to the factor $f^{(\alpha_h)}$ specific to hierarchy $\alpha_h$ for hierarchies as members of the overall market 'geneology' $G(i)$. This linear factor sensitivity is further perturbed by the stock's idiosyncratic sensitivity $\eta_i$ to random variation $\epsilon_i$.

Here, $i \in \{1, \ldots, N\}$, $\eta_i = \left[1 - \sum_{\alpha_h \in G(i)} \gamma^2_{\alpha_h}\right]^{1/2}$, the $h$-th factor $f^{(\alpha_h)}(t)$ and $\epsilon_i$ are i.i.d. random variables with distribution $\sim \mathcal{N}(0,1)$. The $\gamma$ parameter may be fixed such that:

$$\gamma_{\alpha_1} = \sqrt{\rho_{\alpha_1}} \tag{42}$$

$$\gamma_{\alpha_h} = \sqrt{\rho_{\alpha_h} - \rho_{G(\alpha_h)}} \qquad \forall h = 2, ..., N-1, \tag{43}$$

where $\rho_{\alpha_h}$ is the empirical correlation value for each hierarchy $\alpha_h$ and asset, which ensures that the factor model is indeed characterised by the correlation matrix given by the data.

By assuming that $\rho_{\alpha_1} \geq 0$, all the coefficients $\gamma_{\alpha_h}$ are non-negative real numbers, it is further ensured that the implied correlation matrix is always positive definite, which is indeed the necessary training ground for an SPD matrix deep learning model. Yelibi and Gebbie (2021) leverage the general model of Tumminello et al. (2007) to construct a nested linear model of $N$ stock price returns processes $x_i$:

$$h_1 : x_i = \beta_1 \eta_1 + \alpha_1 \epsilon_i, \tag{44}$$

$$\vdots$$

$$h_k : x_i = \beta_1 \eta_1 + \ldots + \beta_k \eta_k + \alpha_k \epsilon_i, \tag{45}$$

$$\vdots$$

$$h_c : x_i = \beta_1 \eta_1 + \ldots + \beta_k \eta_k + \ldots + \beta_c \eta_c + \alpha_c \epsilon_i. \tag{46}$$

The price returns $x_i$ are affected by both $c$ hierarchical factors $\eta$ and the idiosyncratic stock effects $\epsilon$. These factors are present at each level in the hierarchy, with index $k$, such that the correlation matrix is organised in nested clusters $h_c \subset ... \subset h_k \subset ... \subset h_1$ where factor influence is inherited for each sub-group. The factors $\eta_1, ..., \eta_c$ are iteratively stacked onto the stock price process signal such that at each hierarchy $h_1, ..., h_c$ all stocks are correlated and all factors are inherited from parent hierarchies. Further, $\alpha_k = 1 - \sum_j^k \beta_j$, where $\alpha$ determines the cluster strength or correlation density which may be selected by enforcing that $\sum_j \beta_j \in [0, 1]$. The $\beta$ values are selected such that there is a stronger relationship between hierarchies $h_k$ and associated factors $\eta_k$, meaning $\beta_j < \beta_{j+1}$.

### 4.2.2 The Student's t-Distribution Case

It is widely-accepted that the Student's t-distribution provides an improved fit to empirical returns than its Gaussian counterpart (Hu and Kercheval, 2010). To this end, a multivariate data set $X$ of uncorrelated, t-distributed random variables are generated to aid the model's attempt at emulating the heavy-tailedness identified in empirical data studies (Bongiorno et al., 2022). The probability density function of t-distributed random variables is

$$f(x) = \frac{\Gamma\left(\frac{v+1}{2}\right)}{\sqrt{v\pi}\Gamma\left(\frac{v}{2}\right)}\left(1+\frac{x^2}{v}\right)^{-\frac{v+1}{2}}, \tag{47}$$

where the degree of freedom (d.o.f) parameter $v$ determines moment finiteness, specifically, the variance is undefined for $1 < v \leq 2$, finite for $2 < v \leq 4$ with infinite kurtosis, and both are finite for $v > 4$. For large d.o.f values, it is known that the t-distribution converges to the Gaussian as $v \rightarrow \inf$ (Bongiorno et al., 2022). Marti et al. (2016) generate Student's t-distributed time series increments by drawing from an $N$-variate distribution with $v = 3$ such that $X_i \sim t_v\left(0, \frac{v-2}{v}\Sigma\right)$.

### 4.2.3 Correlation Coefficient

The Pearson correlation coefficient is most commonly used in quantitative finance to measure the inter-relatedness of traded assets. The coefficient is defined as:

$$\rho(X,Y) = \frac{\mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]}{\sqrt{\mathbb{E}[X^2]-\mathbb{E}[X]^2}\sqrt{\mathbb{E}[Y^2]-\mathbb{E}[Y]^2}}, \tag{48}$$

which may be estimated by:

$$\hat{\rho}(X,Y) = \frac{\sum_{t=1}^{T}(X^t-\bar{X})(Y^t-\bar{Y})}{\sqrt{\sum_{t=1}^{T}(X^t-\bar{X})^2}\sqrt{\sum_{t=1}^{T}(Y^t-\bar{Y})^2}}, \tag{49}$$

where $\bar{X} = \frac{1}{T}\sum_{t=1}^{T} X^t$ for any pair of traded asset features $X$ and $Y$ empirically observed as $\hat{X}$ and $\hat{Y}$. This correlation coefficient, although widely used, unfortunately only measures the linear relationship between variables and fails in being robust to both data and measurement noise.

The estimate of Spearman's Rho is favoured due to its robustness to noise, and invariance in the face of monotonous transforms of random variables, such as the probability integral transform (Marti et al., 2016). Alternative studies have utilised the Kendall rank correlation to achieve reliable correlations computed from asset returns time series (Miori and Cucuringu, 2022).

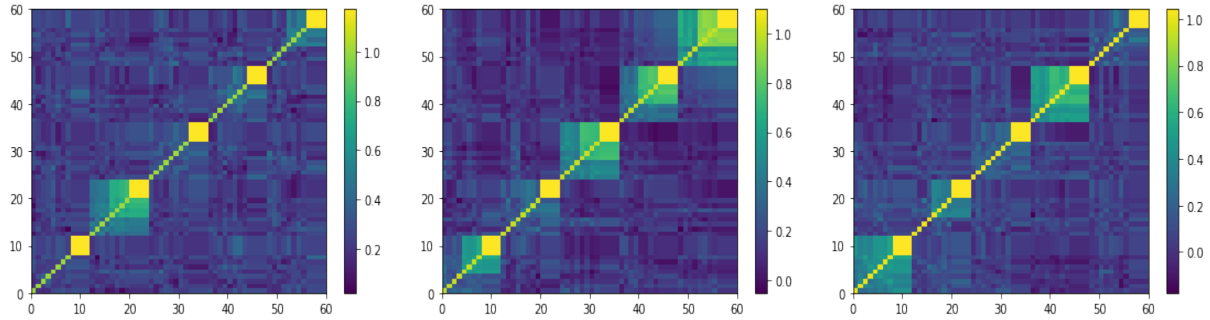
### 4.2.4 Implementation

Realistic correlation structures are generated by employing techniques as implemented by Yelibi and Gebbie (2021) where multiple factors are incorporated in a nested fashion into the price time series model, or into the benchmark HNFM originally proposed by Tumminello et al. (2007). No time factor is incorporated in the generation of these time series or associated correlation matrices. Thus these data provide a set of counterfactual scenarios characterised only by a hierarchically nested structure, where no information lag effects are present in determining the market regime.

Price returns processes are modelled with random variation in the $\beta$ factor sensitivity value in conjunction with additive perturbations in $\eta$ and $\epsilon$ values to construct 18,000 time series for 60 data features in order to mirror the real-world price action data set of BBG JSE top 60 equities for $T = 252$ per computation window. Refer to Section 4.2 for the particular methodology, and see code at Yelibi (2021) and Orton and Gebbie (2024).

The correlation matrices computed on these synthetic time series using Equation 49 preserve SPD qualities, and thus may be used for a Cholesky factor decomposition-motivated Monte Carlo simulation to conduct the SR-guided market regime labelling (Papenbrock et al., Spring 2021). In order to respect the empirically-determined and well-documented fat-tailed nature of financial asset returns time series, this simulation is conducted following Equation 47 by drawing from an $N$-variate distribution with $v = 3$ d.o.f such that $X \sim t_v\left(0, \frac{v-2}{v}\Sigma\right)$ (Marti et al., 2016). Having generated these time series, the correlation matrices are allocated to their relevant 'stressed', 'normal' or 'rally' category in accordance with the method identified by Marti (2021) as above using the contemporaneous stock basket SR as in Equation 40.

Figures 12a, 12b and 12c display correlation matrices from each category comprising systems of 5 clusters with 3 hierarchies each.



**(a)** Stressed class. **(b)** Normal class. **(c)** Rally class.

**Figure 12:** We show representative (simulated) block hierarchical SPD correlation matrices per regime. The synthetic construction uses techniques for nested factor time series generation as proposed by Yelibi and Gebbie (2021) in equations 44 - 45 to create 5 clusters with 3 hierarchies each, sampled from Student's t-distribution with $v = 3$ d.o.f as in Equation 47. The SPD correlation matrices computed on these datasets are permuted to introduce realistic noise with seed = 27. The simulated correlation matrices can be visually compared with the empirical average correlation matrices for each market regime as shown in Figure 10. Refer to Orton and Gebbie (2024) for replicable code implementation.

By construction, the synthetic SPD matrix correlation coefficients display three separate distributions characterised by differing modal values in congruence with the regime switching models introduced by Ang and Bekaert (2004) which identify different macroeconomic regimes being modeled via making draws from two or more possible underlying distributions. This set of distributions is fabricated in the interest of preserving real-world regime correlation dynamics, and is evidenced by the closeness in market regime distributional patterns observed between Figure 13 and Figure 11, using the synthetic and BBG-derived matrices and corresponding labels, respectively. Figures 12 and 13 confirm stylised facts 1 and 4; the synthetic matrices possess a hierarchical structure and their pairwise distributions are indeed positively-shifted.

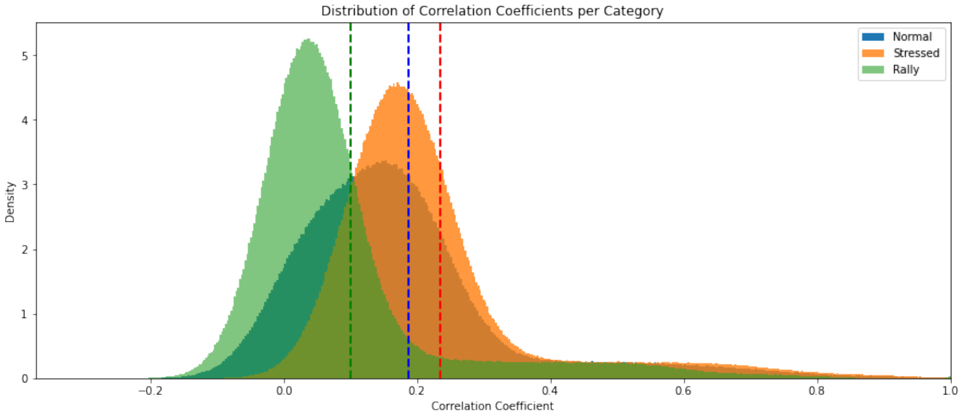


**Figure 13:** Synthetic block hierarchical SPD matrix correlation coefficient density per regime. By construction, the 'stressed', 'normal' and 'rally' regimes have average correlations of 0.24, 0.18 and 0.10 respectively, with descending standard deviation values. The separability of underlying distributions per regime class is preserved to emulate Figure 11.

# 5 Initial Experimentation

Note that these comparative models are trained and tested using open-source code with hyper parameter tuning to achieve optimal market regime classification and hierarchical information representation extraction.

The data sets used for both training and testing are constructed as described in Section 4; correlation matrices are randomly shuffled and allocated to the in-sample and out-of-sample sets meaning that chronology is not respected in these initial experiments. Their purpose is to develop an understanding of their effectiveness in learning deep hierarchical information geometry structures for regime detection in the context of both synthetic and real-world traded asset data without consideration for spatio-temporal correlation information feedback. This provides a comparative benchmark for the importance of causal links in the effectiveness of the models in learning stable feature representation of financial markets.

## 5.1 Model Implementation Details

**Table 1:** Model Configurations. The models acronyms are shown in the model column. The table then provides the Loss Function (LF), the Transformation Matrix Dimension (TMD) that defines the architecure of the model learning systems, the learning rate $\lambda$, and the 'Momentum' (Mom.) parameter. It should be noted that U-SPDNET-6BiRe model configuration is annealed over a range of learning rate parameters, and its TMD is mirrored across the architecture (**). The SPDNet model architecture is visualised in Figure 1, and the U-SPDNet is visualised in Figure 2. BN indicates the use of Riemannian Batch Normalisation (RBN), and BiRe, the implementation of alternating BiMap and ReEig layers. The simulation accuracy results for the configurations are summarised in Table 2.

| **Model** | **Loss Function** | **TMD** | $\lambda$ | **Mom.** |
|---|---|---|---|---|
| SPDNet | Cross-entropy | {60, 20, 3} | $1 \times e^{-3}$ | 0.9 |
| SPDNetBN | Cross-entropy | {60, 20, 3} | $1 \times e^{-3}$ | 0.9 |
| SPDNet-3BiRe | Cross-entropy | {60, 40, 20, 10, 3} | $1 \times e^{-4}$ | 0.9 |
| SPDNetBN-3BiRe | Cross-entropy | {60, 40, 20, 10, 3} | $1 \times e^{-4}$ | 0.9 |
| U-SPDNet-6BiRe | RET + Cross-entropy | {60, 40, 20, 10, 3} (**) | $[1 \times e^{-2}, 1 \times e^{-5}]$ | 0.9 |

In order to ascertain a baseline in market regime detection accuracy, the SPDNet and SPDNetBN models are evaluated on both the BBG JSE top 60 stock price returns and synthetic block hierarchical correlation matrix data sets. In each instance, four configurations are established, namely SPDNet-0BiRe, SPDNetBN-0BiRe, SPDNet-3BiRe and SPDNetBN-3BiRe. The notation $i$BiRe is interpreted as there existing $i$ modules comprising coupled BiMap and ReEig layers (either with or without RBN).

Each of these backbone models is trained over in-sample data set sizes $\in [100, 1000]$ in increments of 100 using SGD with momentum set at 0.9. The learning rate $\lambda$ is selected as $1e^{-2}$ with batch size 30, rectification threshold $\eta = 1e^{-4}$ and random semi-orthogonal weight matrix initialisation for the BiMap layers. It is noted that Singular Value Decomposition (SVD) is used in learning the ReEig layers. [2]

The U-SPDNet is trained and validated employing a 75-25% division, respectively, of both the 18,000 JSE Top 60 and synthetic randomly shuffled SPD correlation matrix data sets. The encoder module of the network structure comprises five operation layers:

$$\mathbf{X_0} \longrightarrow f_b^{(1)} \longrightarrow f_r^{(2)} \longrightarrow f_b^{(3)} \longrightarrow f_r^{(4)} \longrightarrow f_b^{(5)}, \tag{50}$$

where $f_b$ and $f_r$ represent BiMap and ReEig operations, respectively, and $\mathbf{X_0}$ is the initial input SPD correlation matrix. The decoder module mirrors this construction for reconstruction learning. This configuration of six BiRe modules is constructed with two LFE modules. Furthermore, classification is realised by feeding the outputs of the encoder module to LogEig, $f_l$ and fully-connected layers. The resulting model has a total of 26 layers with transformation matrices configured as in Table 1.

The learning rate $\lambda$ is annealed as a function of the training epoch, starting at 0.01 and declining to approach $1e^{-5}$. The model is trained over 600 epochs in batches of size 30 with weight decay set at 0.005 and SGD momentum selected as 0.9 as in Wang et al. (2022b).

## 5.2 Results

The results of experiments conducted OOS are shown in Table 2.

**Table 2:** Regime Detection Accuracy Out-of-Sample. U-SPDNet outperforms all SPDNet benchmark models in the context of the empirical real-world data (JSE Top 60 data). The U-SPDNet OOS accuracy of when trained on the synthetic dataset does not improve on the initial SPDNet model configurations. The various Transformation Matrix (TM) Configurations are shown in the Configuration column. This may suggest that the richer non-linear structure found in the real-world data is better leveraged by more complex architectures.

| **Model** | **Configurations** | **Empirical Data** | **Synthetic Data** |
|---|---|---|---|
| SPDNet | {60, 20, 3} | 68.95% | 66.80% |
| SPDNetBN | {60, 20, 3} | 72.46% | 65.92% |
| SPDNet-3BiRe | {60, 40, 20, 10, 3} | 69.50% | 64.94% |
| SPDNetBN-3BiRe | {60, 40, 20, 10, 3} | 70.71% | 66.20% |
| U-SPDNet-6BiRe | {60, 40, 20, 10, 3} | 98.90% | 65.11% |

### 5.2.1 SPDNet & SPDNetBN

We evaluate the SPDNet-based models using both the ordered JSE top 60 stock returns, and synthetic SPD block hierarchical correlation matrix datasets. Much is revealed by these ini-

---

[2] When practically implementing the Brooks et al. (2019b) SPDNetBN PyTorch library SVD is applied to find the eigendecomposition $\mathbf{X}^{(l-1)} = \mathbf{U}^{(l-1)}\mathbf{\Sigma}^{(l-1)}{\mathbf{U}^{(l-1)}}^T$, where the columns of $\mathbf{U}$ and $\mathbf{\Sigma}$ are orthonormal and $\mathbf{\Sigma}$ is SPD.

tial experiments which provide a baseline specifically for market regime detection given the information contained in these datasets alone.

Firstly, in comparing the 2-layer 60, 20, 3 and 4-layer 60, 40, 20, 10, 3 configurations, it is clear that the introduction of Riemannian batch normalisation layers improves the market state classification accuracy, irrespective of the dataset used for training, which confirms the findings of Brooks et al. (2019a) where the integration of Riemannian batch normalisation layers with the original SPDNet yielded greater radar image classification accuracy.

Furthermore, as identified by Wang et al. (2022b), increased depth holds no relationship with improved classification accuracy since the performance measure convergence occurs within a narrow band around 71% and 66% for the real-world and synthetic data scenarios, respectively, irrespective of model depth. Model performance is weaker when the synthetic block hierarchical correlation matrices are used as training input when compared to average validation accuracies achieved for their real-world counterparts. This may point to the invariant and contrived ordering and underlying distributions of the synthetically simulated matrices providing lower class separability and thus trivialised patterns available for the SPD-Net based models to learn (Yelibi and Gebbie, 2021).

Figures 14 - 17 depict all SPDNet benchmark models converging on training accuracy values nearing 70% which resemble, and even improve on, those achieved when the self-same models are applied in the context of allocating correlation matrices computed from S&P 500 stock price returns to the 'stressed', 'normal' and 'rally' categories as in the work of Marti (2021).

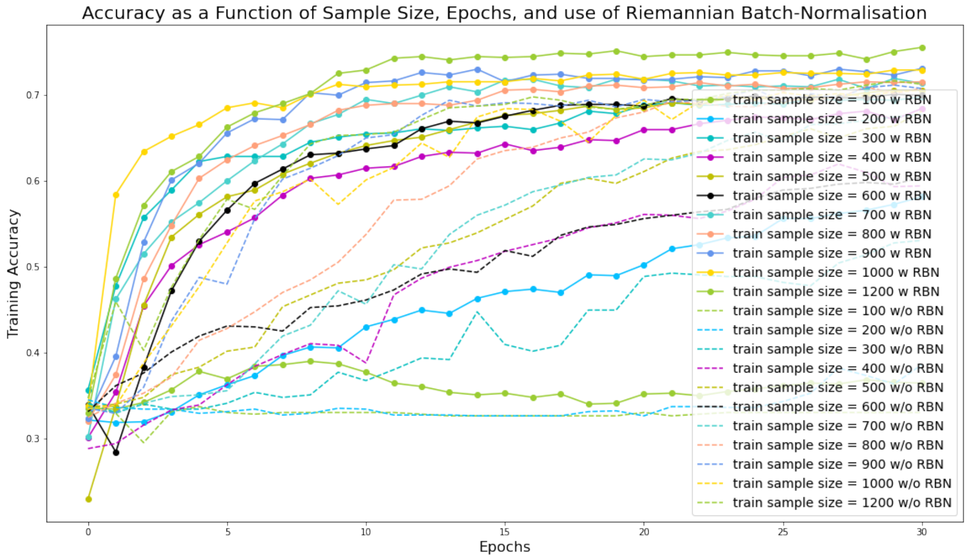


**Figure 14:** Accuracy of SPDNet trained on hierarchically-ordered JSE Top 60 correlation matrices. Distance metric: $\sqrt{2(1-\mathbf{C})}$ where $\mathbf{C}$ is the correlation matrix. Training performed both with (w) and without (w/o) Riemannian batch-normalisation (RBN). In both instances, training accuracy is greater for larger training sample sizes. We note that the introduction of RBN layers improves the market state classification, and that this effect is enhanced for larger sample sizes. This can be compared to the SPDNet trained on the simulated dataset set as shown in Figure 16 and suggests that training on empirical data is more effective.

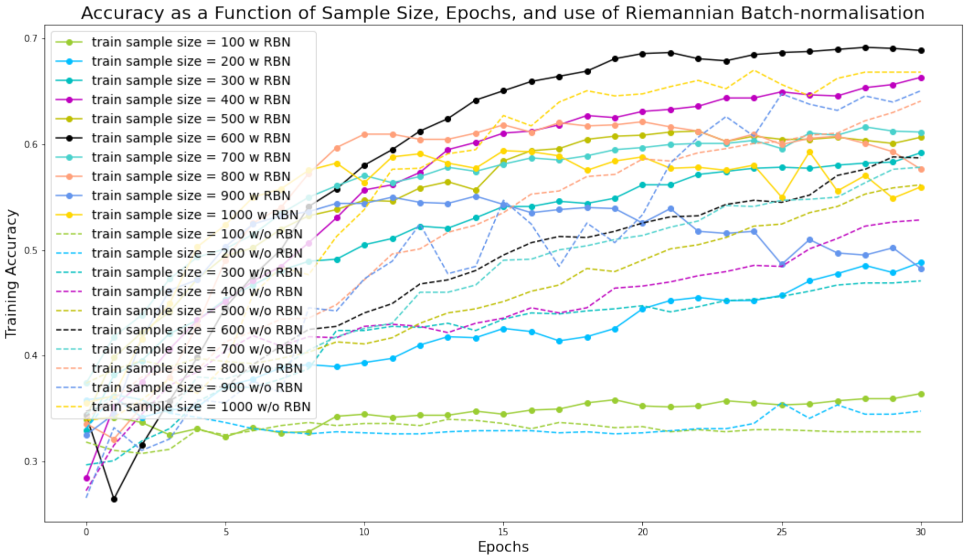


**Figure 15:** Accuracy of SPDNet-3BiRe trained on hierarchically-ordered JSE Top 60 correlation matrices. Distance metric: $\sqrt{2(1-\mathbf{C})}$ where $\mathbf{C}$ is the correlation matrix. Training performed both with (w) and without (w/o) Riemannian batch-normalisation (RBN). Similarly as found for SPDNet in Figure 14, we note that the introduction of RBN layers improves the market state classification. This can be compared to the SPDNet-3BiRe trained on the simulated dataset set as shown in Figure 17, and suggests that training on empirical data is more effective. Perhaps because of the enhanced existence of non-linear patterns in real world data.

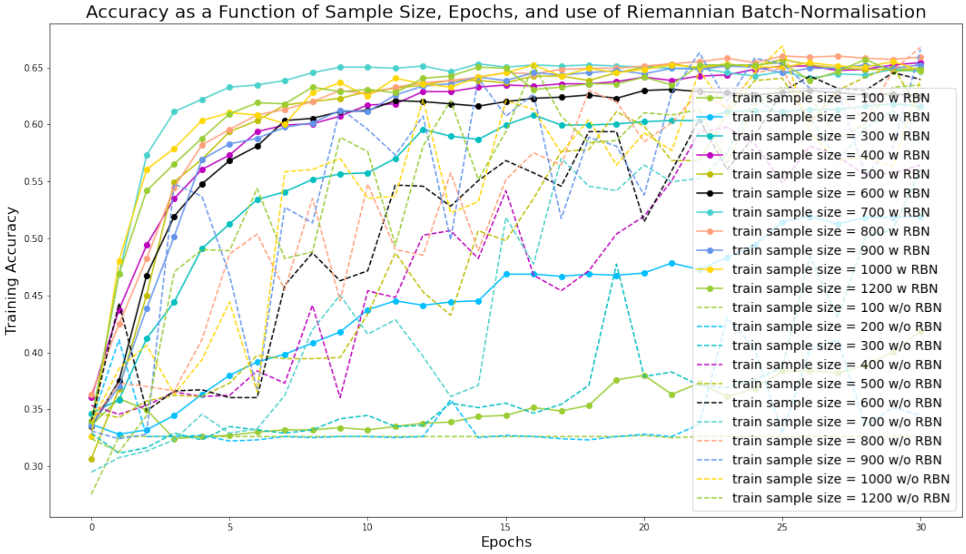


**Figure 16:** Accuracy of SPDNet trained on synthetic block hierarchical SPD correlation matrices. Training performed both with (w) and without (w/o) Riemannian batch-normalisation (RBN). This can be compared to the SPDNet performance in Figure 14 using the empirical dataset, and we again note that the introduction of RBN layers improves the market state classification.

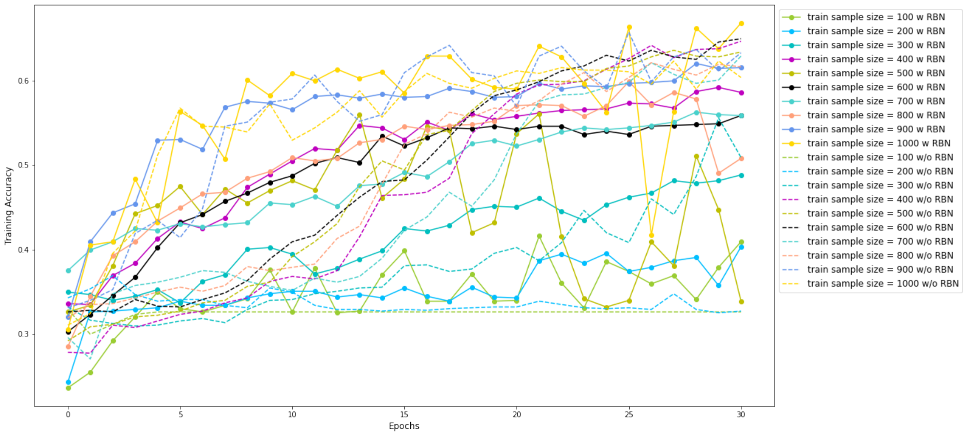


**Figure 17:** Accuracy of SPDNet-3BiRe trained on synthetic block hierarchical SPD correlation matrices. Training performed both with (w) and without (w/o) Riemannian batch-normalisation (RBN). This can be compared to SPDNet-3BiRe trained on empirical correlation matrices in Figure 15. The introduction of RBN layers can improve the market state classification, but the effect seems moderate as compared to that seen in SPDNet trained models, but still improves of relatively larger training datasets.

### 5.2.2 U-SPDNet

Information degradation with time is identified as hampering the potential for SPD manifold statistical feature embedding to deliver improved reconstruction accuracy when extended to unseen input data spaces, as is evident when comparing SPDNet-based architectures irrespective of type of data or whether RBN is utilised. Thus the work of Wang et al. (2022b) in developing the U-SPDNet will be used to investigate whether improving information up-sampling through the latent factor-based feature mapping of the model leads to ameliorated market regime detection, or a preservation of the block-hierarchical structure of the correlation matrices in the model embeddings.

Table 2 shows that the U-SPDNet significantly outperforms all SPDNet benchmark models in the context of the JSE Top 60 data. This suggests that increasing the representational capacity of the model for extraction of structured features to learn a subspace characterised by low intra-class variability and inter-class similarity yields improved regime detection capability of the deep learning model. However, the U-SPDNet OOS accuracy of 65.11% when trained on the synthetic dataset does not improve on the results of the initial SPDNet model configurations. This may suggest that empirical, or measured, financial market data has structured features and non-linear patterns that can be extracted, and that because these types of repeating non-linear patterns are not represented in synthetic data, the algorithm capabilities are not fully leveraged therein. This may suggest that learning statistical features may be more important than learning causal relationships between different market regimes. However, this may be because of data leakage – and thus merely be a form of data over fitting.

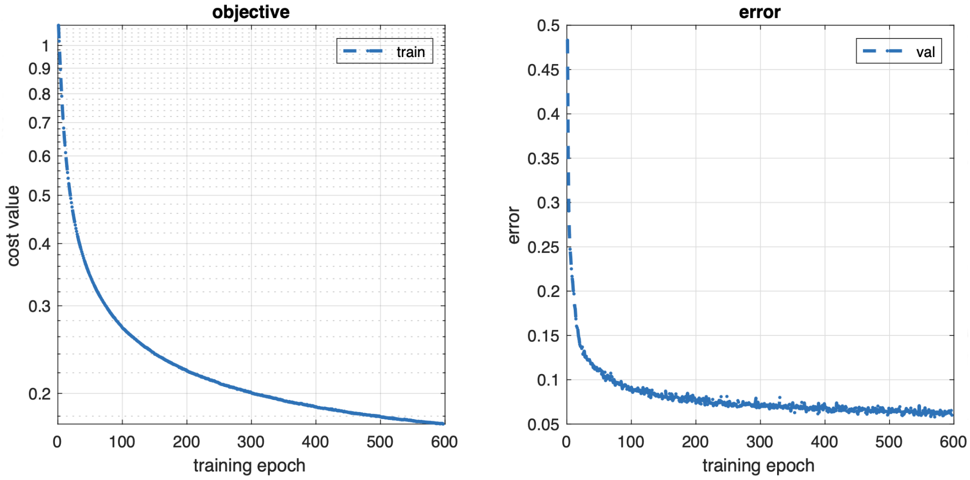


**Figure 18:** Convergence behaviour of U-SPDNet-6BiRe on the hierarchically-ordered JSE Top 60 dataset. Here we show the objective function as CEL + RET, and error as the accumulated average regime classification error. Training performed without Riemannian batch-normalisation (RBN) with parameters configured as in Table 1.

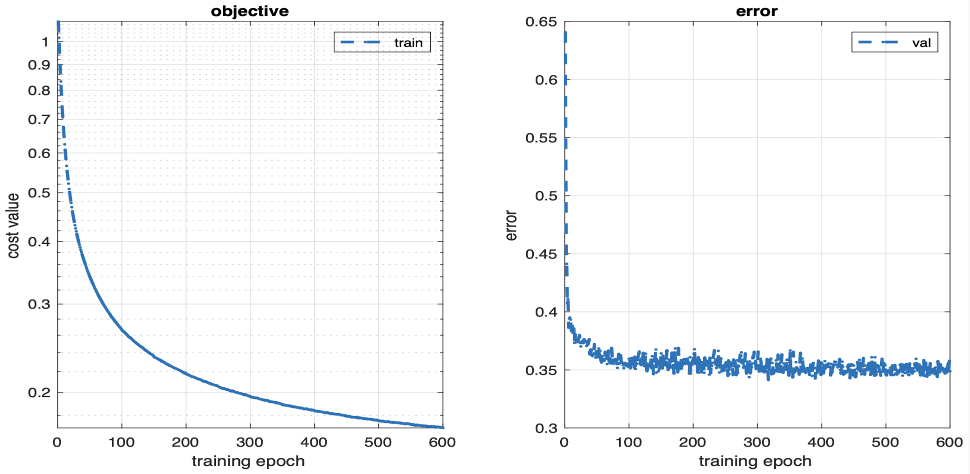
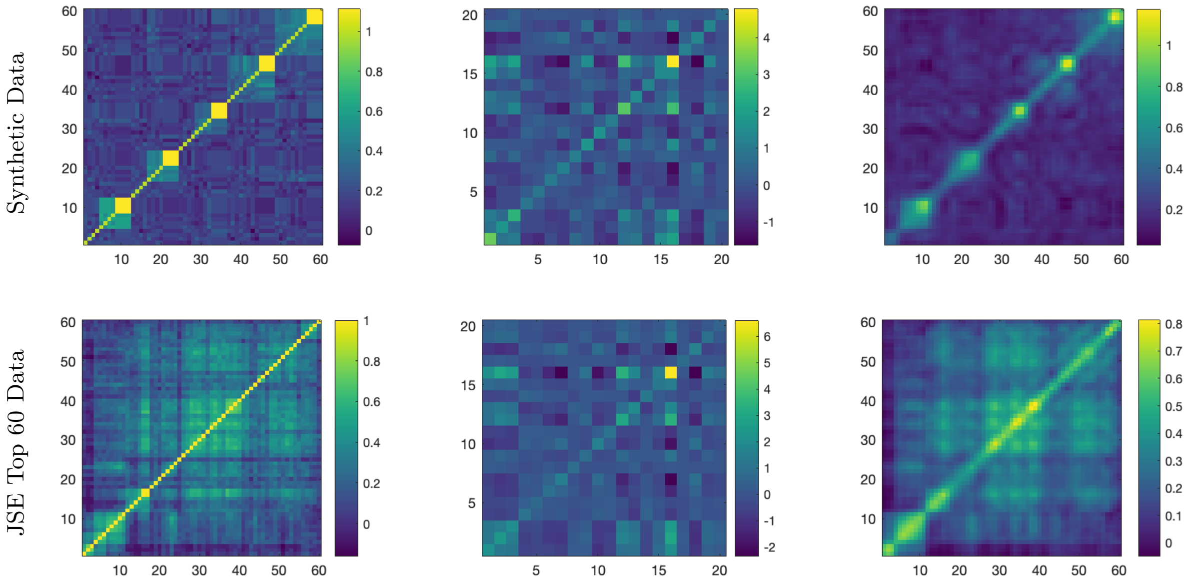


**Figure 19:** Convergence behaviour of U-SPDNet-6BiRe on the synthetic block hierarchical SPD correlation matrix dataset. Here we show the objective function as CEL + RET and error as the accumulated average regime classification error per epoch. Training performed without Riemannian batch-normalisation (RBN) in line with parameter configuration shown in Table 1.

Figures 18 and 19 show the convergence of accumulated average regime classification error toward values of 0.05 and 0.35 on the JSE Top-60 and synthetically-generated block hierarchical validation data sets, respectively. These values may point to information signal available in the data from which the U-SPDNet-6BiRe may learn, where shallow and non-reflexive patterns will be present in the synthetic data by construction meaning that predictions will not be information-informed to the same degree as in the case of the JSE Top 60 data.



**(a)** Input layer (60 × 60). **(b)** Latent space layer (20 × 20). **(c)** Reconstructed layer (60 × 60).

**Figure 20:** We show example input, latent space and reconstructed layer matrices for U-SPDNet-6BiRe trained and validated on both synthetic nested block hierarchical correlation matrices and their JSE Top 60 data counterparts, configured as in Table 1. Here, a representational example drawn from the $600^{th}$ epoch. The block hierarchical structure is maintained in the latent layer in both instances of validation. The reconstructed synthetic output matrix fails to perfectly replicate the 5-block structure with 3 hierarchies each, or the unit diagonal correlation characteristic of traded stocks. In the JSE Top 60 data case, the reconstructed matrix 'smooths' the input feature space, meaning that the block hierarchical structure is preserved, but a one-to-one information mapping is not perfectly achieved OOS. The extracted latent 20-feature layers show preservation of the block-hierarchical structure and meaningful statistical factor identification. The latent space matrices have similar structures with subtle differences, pointing to the U-SPDNet-6BiRe model capturing a tractable underlying correlation information set on the SPD manifold which incorporates the differentiating time factor feature of the real world data in comparison to their synthetic counterpart.

The RET encourages the SPD network to approach an identity mapping, which encodes the inter- and intra-class distributions during training. Given that the JSE top 60 stock price returns correlation matrices exhibit little distributional separability this approach is hoped to deliver improved regime detection classification by virtue of constructing information geometry-dense deep features characterising the original block hierarchical financial data.

It is the case that the RET component of the objective function approaches a value of 0.05, which indicates fairly accurate, though imperfect, reconstruction of the input SPD matrix, meaning that the block hierarchical structure is preserved in both instances of model validation on the benchmark and BBG datasets. Of course, as discussed above, this high replication accuracy value may stem from information leakage induced by the randomisation of training and validation data points. The sampled reconstructed matrices, selected from the final epoch, depict nested block hierarchical structures on the diagonal with a smoothing effect being observed in that the maxima of the correlation values are smaller than those of their corresponding input matrices. Furthermore, as in the application of the U-SPDNet to the AFEW data in the work of Wang et al. (2022b), we identify increased diagonal values in the reconstructed SPD matrices.

These findings point to limitations of the U-SPDNet model being attributable to the following factors:

1. The U-SPDNet places emphasis on learning statistical features as opposed to the encoded temporal changes between different market regimes, which results in poor pattern-related representational capacity of the learned features.

2. The SPD correlation matrices computed in both the original JSE Top 60 (empirical data) and synthetic data cases are learned without consideration being made for the sequence of the data themselves, meaning that error and implicit data volatility information is not eliminated in the end-to-end learning process.

3. Since the data are randomly shuffled prior to model evaluation, there is a high risk of information leakage amidst the training and validation correlation matrix datasets. This would contribute to the significantly greater accuracy achieved by the U-SPDNet when trained on the real-world JSE data. This challenge is addressed using the *purging* and *embargo* techniques introduced in Section 7.3 below (Lopez De Prado Lopez, 2022).

4. The data 'scrambling' would induce the loss of all dependence information (Politis, 2003). Thus block sampling techniques which preserve the chronology of OOS data are invoked in Section 6 below to test the extent to which this impacts on model performance.

5. Finally, by construction no temporal relationship exists between correlation matrices in the synthetic dataset. The performance of U-SPDNet being similar to that of the SPDNet indicates that without the spatio-temporal information feedback loops present in real-world data, the model's estimation accuracy is negatively affected by the lack of this non-linear causal dynamic, despite the encoded presence of hierarchical structures whose underlying distributions vary per regime.

# 6 Block Resampling Experimentation

## 6.1 Methodology

Financial time series data exhibit temporal dependence patterns. It is recognised that the simple application of i.i.d. bootstrap techniques for data sampling results in all dependence information being lost. Indeed, statistical estimators become inconsistent if this technique is applied, as evidenced by results in sections 5.2.1 and 5.2.2 above. To provide a contrasting experimental design which may be compared to that of the above initial experimentation using the randomisation of training and validation data, we consider block resampling in the case of the SPDNet.

Following the work of Politis (2003), a block resampling approach is implemented here in order to maintain the chronology of the financial time series data. The methodology is as follows:

1. **Correlation Matrix Computation**: As in the work of Rodriguez Dominguez (2023), SPD correlation matrices are computed in rolling 6-month (126-day) windows, the selection of which is motivated by experiments confirming that a shorter window allows improved adaptation to changing market conditions regimes.

2. **Block Resampling**: Blocks are sampled with replacement in a rolling 252-day window and split into training and validation sets in a 75:25 ratio. This window length is chosen given that the returns time series is computed by differencing the daily closing price and is thus assumed to be stationary, implying that the block size $b$ may be selected as $b = 2\hat{m}$, where $\hat{m}$ is the smallest integer such that the tolerance $\hat{\gamma}(k) \approx 0$ for $k > \hat{m}$ and where each block $k$ has estimated covariance $\hat{\gamma}(k)$. The size of $\hat{\gamma}(k)$ is selected based on the work of Lopez De Prado Lopez (2022) as this results in an empirical tolerance of 0 using S&P 500 data. This process creates 5871 windows spanning June 2000 - December 2023.

3. **Tapering/Purging**: The price returns data points found at the beginning, end and train-test border are removed from the data in order to combat data leakage following the recommendations of López de Prado (2018) and to implement a version of the tapering approach discussed by Politis (2003) for improved accuracy of variance estimators computed on the window, with replacement.

## 6.2 Results

Unsurprisingly, when trained and validated on the block-sampled rolling data windows of JSE Top 60 hierarchically-ordered correlation matrices, the SPDNet delivers a corner solution OOS as in the initial experiments.

Despite the block resampling technique maintaining the data chronology both in- and out-of-sample, the SPDNet fails to capture the temporal nature of the correlation matrix information on its respective causal manifold. When considered in tandem with the results observed in Section 5, viewed through the lens of reflexivity, it becomes clear that the SPDNet is only capable of learning a stable latent representation of the market which holds in the majority of the timeframe under consideration, but degrades in terms of information particular to stressed and rally market phases (Soros, 2013).

Figures 21 and 22 clearly depict a breakdown in the representational capacity of the SPDNet model in times of market stress and rally. Accuracy drops as low as 38% during the timeframe of the GFC, while increased levels of volatility are seen to correspond to lower model performance.

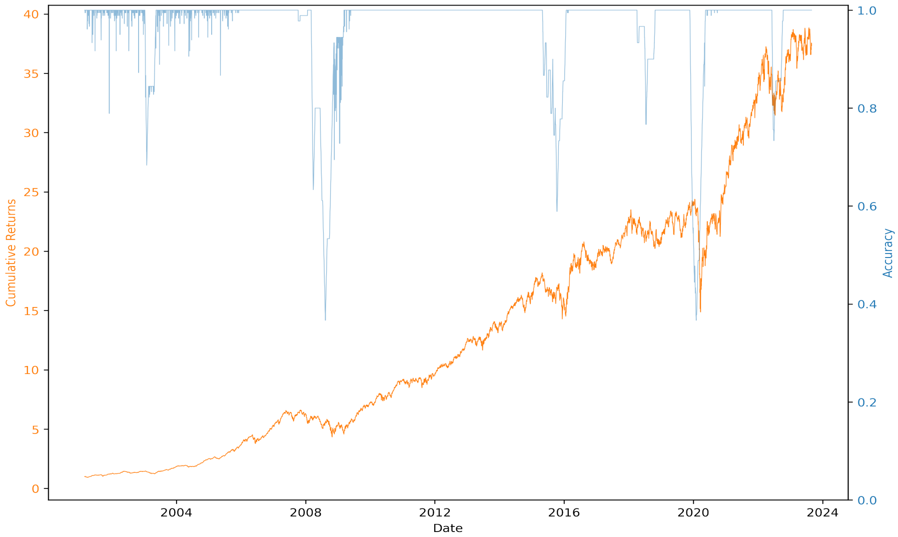


**Figure 21:** We train and validate the SPDNet using the block resampling technique which generates 5871 data windows. The model delivers perfect regime detection accuracy in phases of market action which may be identified as normal, i.e. outside of rally or stressed periods. Accuracy drops as low as 24% OOS during rally and stressed regimes which are characterised by increased volatility in average daily returns. Here, the decrease in accuracy of the SPDNet in detecting stressed and rally regimes is confirmed. A significant drop in regime detection accuracy is observed when the cumulative average daily returns pass through bull and bear market phases as opposed to that of normal phases characterised by lower volatility levels.

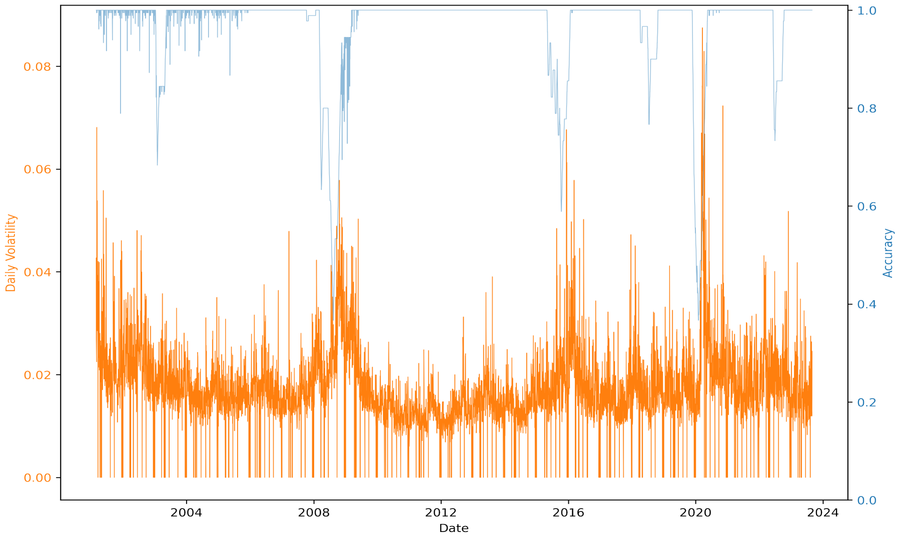


**Figure 22:** We depict an anecdotal relationship between SPDNet regime detection accuracy and market-level volatility as measured by asset return standard deviation. An increase in volatility which is associated with both stressed and rally regimes may be linked to a decrease in model classification accuracy OOS. This observation points to the SPDNet unsuccessfully learning temporal feedback loops which characterise and drive the hierarchical market structure through different regimes and volatility cycles.

A further investigation of this relationship is considered in Figure 23. When the correlation between model accuracy and market volatility as measured by the returns standard deviation, and lagged by + and -252 days, there is a positive relationship over 200 days in the past, with a minimum being reached within a 50-day positive lag. This indicated that the model is failing to learn the causal temporal links driving the market through regime-specific hierarchical structure.

This lagged correlation structure points to there being an alternative explanation for the evolution of financial markets through varying regimes which does not depend on a set of initial conditions, but rather, on one that considers that no single 'snapshot' of the system is satisfactory in explaining the rest of the hierarchical correlation structure as it shifts with time with market phases.

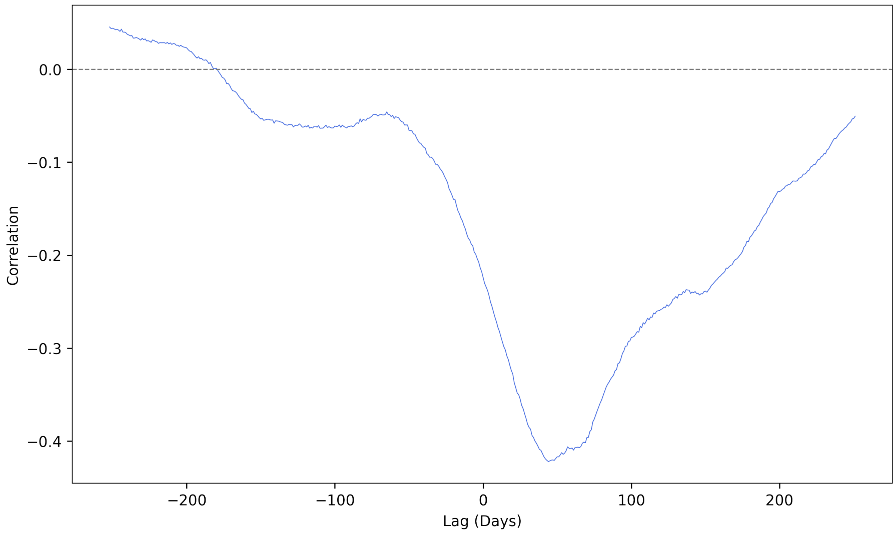


**Figure 23:** We show the correlation between SPDNet regime detection accuracy and market returns data with positive and negative lag of 252 days. A positively-correlated relationship exists between SPDNet regime detection accuracy and market return volatility when lagged by over 200 days. This is an indication that an increase in volatility precedes accurate model state identification by almost one financial year. This indicates that the model may learn lagged temporal correlation dependencies which characterise hierarchical structure evolution through differing regimes should the OOS timeframe lag the training dataset by over 200 days.

In the realm of counterfactual thinking, and considering the work of Soros (2013), it becomes even more likely that reflexive feedback loops need to be considered within the deep Riemannian manifold learning context for predictions to be understood from a standpoint which does not result in infinite regress (Marletto, 2021).

# 7 Real-world Application: Asset Allocation

Given the importance of the co-evolution of hierarchical asset relationships and varying financial market regimes in the context of investment framework development, we develop an initial framework for optimising asset allocation leveraging the deep-learned underlying system structure as executed by the models previously described in the task of effective regime detection for JSE Top 60 traded assets.

A benchmark optimisation strategy is first considered: Markowitz's seminal work supporting diversification by applying the Mean-Variance rule which maximises an investment objective whilst minimising undesirable overall portfolio risk. This approach ignores market regime switching.

A second comparative regime-switching optimisation approach is proposed which leverages the market regimes detected OOS by the deep learning frameworks implemented in this dissertation. The performance of the SPDNet and U-SPDNet on both real-world JSE Top 60 data and synthetic block hierarchical data points to their effectively learning the information feature structures driving the hierarchical evolution of asset dynamics through differing macroeconomic regimes for accurate regime detection to the order of circa 70% OOS.

A tailored regime-dependent strategy is thus formulated to determine whether additional value may be derived from the implementation of these deep learning algorithms in conjunction with widely-used portfolio optimisation techniques used in the quantitative finance industry.

## 7.1 Mean-Variance Portfolio Optimisation

Markowitz's original proposition in his paper Portfolio Selection (Markowitz, 1952) is that diversified portfolios which satisfy an investor's risk-reward preference frontier will outperform an equally-weighted portfolio. Crucially, his idea argued for forward varying means and covariances, as opposed to backward looking historic means and covariances.

Consider a universe of $n$ risky stocks as characterised by an expected return vector $\mathbb{E}[\mathbf{R}]$ and covariance matrix $\boldsymbol{\Sigma}$, where the investor's capital allocation amongst these assets is governed by a set of weights $\mathbf{w}$, with a vector transpose denotated as $\mathbf{w}'$. The covariance matrix is comprised of components $\Sigma_{ij} = \sigma_i \sigma_j \rho_{ij}$ for stocks $i$ and $j$ with standard deviation and correlation values $\sigma$ and $\rho$, respectively. The expected portfolio return is therefore computed as:

$$\mathbb{E}[R_P] = \mathbf{w}'\mathbb{E}[\mathbf{R}] = \sum_{i=1}^{n} w_i \mathbb{E}[R_i]. \tag{51}$$

The empirical portfolio risk is:

$$\sigma_P^2 = \mathbf{w}'\boldsymbol{\Sigma}\mathbf{w} = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \Sigma_{ij} = \sum_{i=1}^{n} w_i^2 \sigma_i^2 + 2\sum_{i \neq j} w_i w_j \sigma_i \sigma_j \rho_{ij}. \tag{52}$$

The optimisation objective may be described as minimising the overall portfolio risk subject to two constraints, the first fixing a particular portfolio expected target return on the efficient frontier, and the second, ensuring that the available wealth is fully invested:

$$\text{argmin}_{\mathbf{w}} \left\{ \mathbf{w}'\boldsymbol{\Sigma}\mathbf{w} \right\}, \text{ s. t. } \mathbb{E}[R_P] = \mathbf{w}'\mathbb{E}[\mathbf{R}], \text{ and } \mathbf{w}'\mathbf{1} = 1. \tag{53}$$

Here $\mathbf{1}$ is a vector of ones. The Kuhn-Tucker method using the Lagrangian multiplier vector $\boldsymbol{\lambda}$ may be employed to solve for this optimisation problem as follows:

$$L(\mathbf{w}, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{w}'\boldsymbol{\Sigma}\mathbf{w} - \boldsymbol{\lambda}' \begin{pmatrix} \mathbf{w}'\mathbb{E}[\mathbf{R}] - \mathbb{E}[R_P] \\ \mathbf{w}'\mathbf{1} - 1 \end{pmatrix}. \tag{54}$$

The $\frac{1}{2}$ is included in order to simplify subsequent operations. The optimal investment policy weights may be uncovered by first setting the first-order partial derivatives of the Lagrangian to 0:

$$\frac{\partial L}{\partial \mathbf{w}} = \boldsymbol{\Sigma}\mathbf{w} - \boldsymbol{\lambda}' \begin{pmatrix} \mathbb{E}[\mathbf{R}] \\ \mathbf{1} \end{pmatrix} = \mathbf{0}, \tag{55}$$

$$\frac{\partial L}{\partial \lambda} = \begin{pmatrix} \mathbf{w}'\mathbb{E}[\mathbf{R}] - \mathbb{E}[R_P] \\ \mathbf{w}'\mathbf{1} - 1 \end{pmatrix} = \mathbf{0}. \tag{56}$$

The optimal investment policy in terms of the two components of the Lagrangian multiplier vector, $\lambda_1$ and $\lambda_2$ derived from Equation 55 is:

$$\mathbf{w}^* = \boldsymbol{\Sigma}^{-1}(\lambda_1\mathbb{E}[\mathbf{R}] + \lambda_2\mathbf{1}). \tag{57}$$

The optimal weight vector $\mathbf{w}^*$ is substituted into Equation 56 to produce the following derived equation:

$$\begin{pmatrix} [\boldsymbol{\Sigma}^{-1}(\lambda_1\mathbb{E}[\mathbf{R}] + \lambda_2\mathbf{1})]'\mathbb{E}[\mathbf{R}] \\ [\boldsymbol{\Sigma}^{-1}(\lambda_1\mathbb{E}[\mathbf{R}] + \lambda_2\mathbf{1})]'\mathbf{1} \end{pmatrix} = \begin{pmatrix} \mathbb{E}[R_p] \\ 1 \end{pmatrix}, \tag{58}$$

$$\begin{pmatrix} \mathbb{E}[\mathbf{R}]'\boldsymbol{\Sigma}^{-1}\mathbb{E}[\mathbf{R}] & \mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbb{E}[\mathbf{R}] \\ E[\mathbf{R}]'\boldsymbol{\Sigma}^{-1}\mathbf{1} & \mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} \mathbb{E}[R_P] \\ 1 \end{pmatrix}. \tag{59}$$

For simplifying purposes, the following values are defined:

$$\begin{aligned} A &= \mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbb{E}[\mathbf{R}], & B &= \mathbb{E}[\mathbf{R}]'\boldsymbol{\Sigma}^{-1}\mathbb{E}[\mathbf{R}], \\ C &= \mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1}, & D &= BC - A^2. \end{aligned}$$

By inverting the matrix pre-multiplied by $\boldsymbol{\lambda}$ in Equation 59, the optimal Lagrangian multipliers may be determined in terms of the target portfolio return:

$$\lambda_1^* = \frac{\mathbb{E}[R_P]C - A}{D}, \text{ and } \lambda_2^* = \frac{B - \mathbb{E}[R_P]A}{D}. \tag{60}$$

By substitution, the optimal weights are obtained as follows:

$$\mathbf{w}^* = \frac{1}{D}(B\mathbf{\Sigma}^{-1}\mathbf{1} - A\mathbf{\Sigma}^{-1}\mathbb{E}[\mathbf{R}]) + \frac{1}{D}\mathbb{E}[R_P](C\mathbf{\Sigma}^{-1}\mathbb{E}[\mathbf{R}] - A\mathbf{\Sigma}^{-1}\mathbf{1}). \tag{61}$$

This gives the celebrated mutual fund separation theorem, which describe the efficient frontier as the linear combination of a low risk minimum variance portfolio with a unique, higher-risk Sharpe optimal portfolio. A portfolio manager may choose to invest on a daily basis, and would implement this Mean-Variance optimisation strategy by selecting a target portfolio return $\mathbb{E}[R_P]$ and computing the empirical expected returns and covariance matrix, $\mathbb{E}[\mathbf{R}]$ and $\mathbf{\Sigma}$, using the returns time series of the assets under consideration at the beginning of the investment period. Subsequently, the optimal weights may be solved for and the portfolio re-balanced on a daily basis to enforce these originally-identified weights. This is referred to as a Constant Mix (CM) strategy.

## 7.2 Regime-Dependent Mean-Variance Optimisation

As depicted in Figure 10, the JSE Top 60 market microstructure varies in space and time, both in terms of its hierarchical structure and volatility dynamics specific to different regimes. Analysis of sub-figures 10a, 10b and 10c reveals an average greater correlation and decreased hierarchical separability present in stressed market regimes, as compared to lower average volatility and a greater degree of economic taxonomy separability in normal and rally markets. Given these quantified phenomena, it is likely that different investment decisions would yield optimal results for differing states, as is shown in the works of Singo (2022) and Miori and Cucuringu (2022).

To account for the non-stationarity of market dynamics between regimes, the Mean-Variance portfolio optimisation problem described above may be adapted as follows:

$$\text{argmin}_{\mathbf{w}} \left\{\mathbf{w}'_{r_i}\mathbf{\Sigma}_{r_i}\mathbf{w}_{r_i}\right\}, \text{ s. t. } \mathbf{w}'_{r_i}\mathbb{E}[\mathbf{R}_{r_i}] = \mathbb{E}[R_P], \text{ and } \mathbf{w}'_{r_i}\mathbf{1} = 1. \tag{62}$$

Here $r_i$ represents the market regime for $i \in \{0, 1, 2\}$.

Practically, this approach is implemented by using the SPDNet and U-SPDNet-6BiRe models to estimate 'stressed', 'normal' and 'rally' market regimes in chronological OOS data.

## 7.3 Purging & Embargo

Conducting portfolio strategy backtesting using financial stock price data requires construction of training and testing datasets where chronology of the time series is respected OOS for interpretative ease, and information leakage is mitigated.

Information leakage occurs when data included in the test dataset is also used in the training dataset. Should a serially-correlated stock price correlation matrix $X$ with regime label $Y$ be formed using overlapping data, information may be leaked because:

(i) $X_t \approx X_{t+1}$; and

(ii) $Y_t \approx Y_{t+1}$, since labels are derived from these overlapping data points.

When the deep SPD matrix classifiers are first trained on $(X_t, Y_t)$ and then used to estimate $\mathbb{E}[Y_{t+1}|X_{t+1}]$ based on the empirically-observed $X_{t+1}$, these models have increased likelihood of predicting regime labels $Y_{t+1} = \mathbb{E}[Y_{t+1}|X_{t+1}]$ even if $X$ is irrelevant. Leakage takes place only if $(X_{\text{train}}, Y_{\text{train}}) \approx (X_{\text{test}}, Y_{\text{test}})$.

Two methods are proposed by López de Prado (2018) to counter information leakage, which we implement in backtesting of comparative portfolio optimisation strategies, and are visualised in Figure 24:

1. **Purging**: All returns observations whose labels overlap in time with those which are also present in the test set are removed. These regime labels $Y_j$ which are functions of observations at times $t \in [t_{j,0}, t_{j,1}]$ such that $Y_j = f([t_{j,0}, t_{j,1}])$ overlap with labels $Y_i = f([t_{j,0}, t_{j,1}])$ if any of three sufficient conditions are met; namely: $t_{j,0} \leq t_{i,0}$; $t_{j,0} \leq t_{i,1} \leq t_{j,1}$; $t_{i,0} \leq t_{j,0} \leq t_{j,1} \leq t_{i,1}$.

2. **Embargo**: The presence of serial correlation in the time series of correlation matrices motivates the elimination of observations in the training set which immediately follow those in the testing set. The embargo period $h$ is implemented by setting $h = 0.1T$, where $T$ is the number of observations in the test dataset, prior to purging, such that $Y_j = f([t_{j,0}, t_{j,1} + h])$.

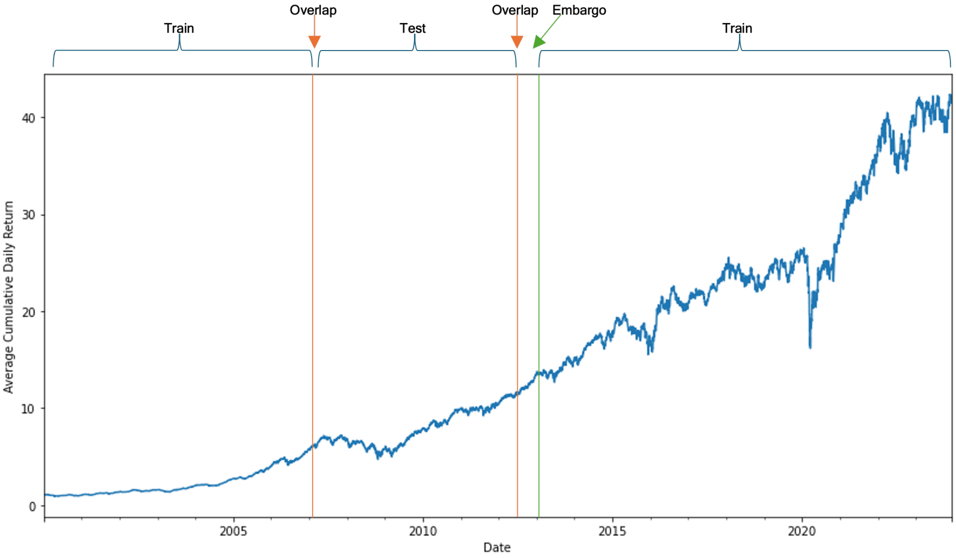


**Figure 24:** One partition of the BBG JSE Top 60 stock price returns data being considered for model training and portfolio allocation strategy backtesting. The subjectively-selected test set spanning January 2007 - December 2012 is surrounded by two train sets, generating two overlaps that must be purged to prevent leakage. To further prevent leakage, the train observations immediately after the testing set are also embargoed. This is performed by removing the last 10% of test data, resulting in a test set available for correlation matrix calculation existing between January 2007 and mid-July 2012 (López de Prado, 2018).

## 7.4 Investment Strategy Comparison

Whilst backtesting may not be regarded as a useful research tool, for the purposes of illustrating potential uses for the regime estimation output of our information geometry-motivated deep learning models in the context of quantitative finance, we compare regime switching strategy performance OOS. Note that no trading costs are considered in this backtest implementation.

Specifically, the following steps are followed:

1. The BBG JSE Top 60 stock price returns data are split into test and training sets, with the test data falling between January 2007 and December 2012. This selection is made subjectively in order to include data corresponding to all three identified regimes for testing. The remainder of the data which falls to either side of the testing period is allocated to the training dataset.

2. As in Figure 24, the overlap dates 2006-12-31 and 2013-01-01 are purged, and the training data is embargoed prior to this by removing the final 10% of the test dataset. Thus the test dataset comprises information extending from January 2007 to mid-July 2012 only and data leakage is mitigated.

3. The training data are augmented such that the stressed, normal and rally regimes are equally-represented in number of correlation matrices. This is executed by randomly sampling 252-day windows from the training data, computing the Pearson correlation between assets, and applying a Single Linkage algorithm with Ward distance to re-order the matrices in order to reveal hierarchical relationships. 18,000 matrices are constructed with 6,000 belonging to each regime.

4. The same process is applied to the test data period, where a moving window of 252 days is employed to compute, and then hierarchically reorder, correlation matrices for the market capitalisation-ordered top 60 stocks traded on the JSE. This results in chronological correlation matrices being available between January 2008 and mid-July 2012 for model testing purposes.

5. SPDNet and U-SPDNet-6BiRe are trained using the augmented dataset comprising 18,000 matrices and tested on the unshuffled, chronological, test data correlation matrices. Thus two sets of regime labels are estimated OOS which are shown in Figure 29 and Figure 30 below.

6. In order to construct the Constant-Mix Mean-Variance portfolio optimisation backtest where there is no change in investment decision over the entire test data window, the covariance matrix and expected returns, $\mathbf{\Sigma}$ and $\mathbb{E}[\mathbf{R}]$, are estimated from the training data. Daily rebalancing is implemented to maintain the optimal weights determined for allocation amongst the 60 stocks in the primary period as described in Section 7.1, with an equal-weighted portfolio being used as the initial optimal guess.

7. Two additional regime switching portfolio optimisation approaches are backtested, corresponding to the regimes estimated by the SPDNet and U-SPDNet-6BiRe models OOS. Specifically, the methodology described in Section 7.2 is applied in both instances, where covariance matrices and expected returns are estimated per regime in the training data, according to the SR-labeled data, which are then used as input parameters for the daily-rebalancing regime-dependent optimisation performed OOS on the asset returns comprising the test dataset.

### 7.4.1 SPDNet & U-SPDNet Regime Detection

We return to the baseline SPDNet and U-SPDNet models described in Section 3 to illustrate the comparison of the constant-mix and regime-switching strategies. In order to marry portfolio optimisation with market regime detection, an illustrative approach will be adopted, where we employ the SPDNet to establish OOS regime labels. The SPDNet and U-SPDNet-6BiRe models are parameterised as in Table 1.

Figures 25 and 26 depict model convergence in both instances, with the SPDNet achieving an accuracy of 70.44% OOS and the U-SPDNet predicting test sample market regimes with 64.9% accuracy.

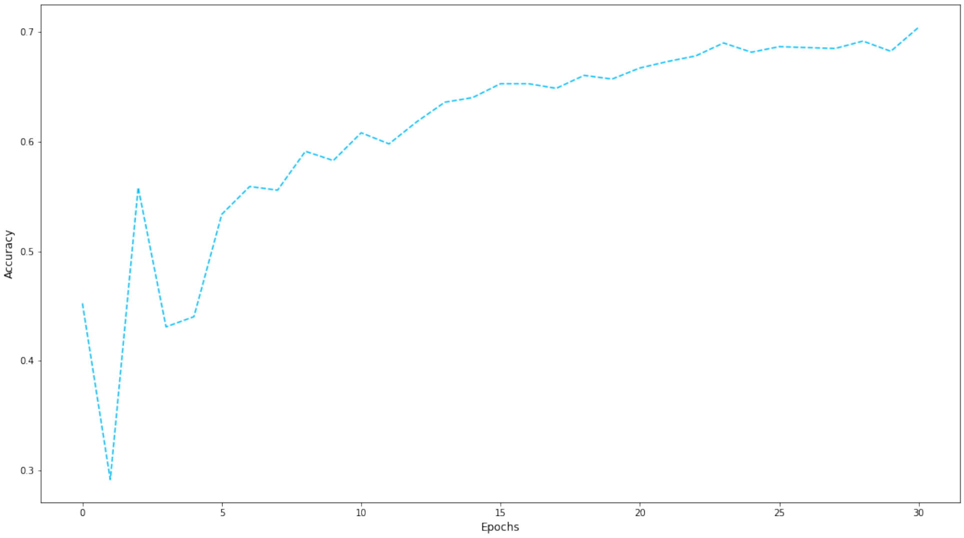


**Figure 25:** A baseline SPDNet model trained and tested on JSE Top 60 data partitioned as in Figure 24 converges on an OOS regime label accuracy of 70.44%. We parameterise this baseline model as in Table 1.

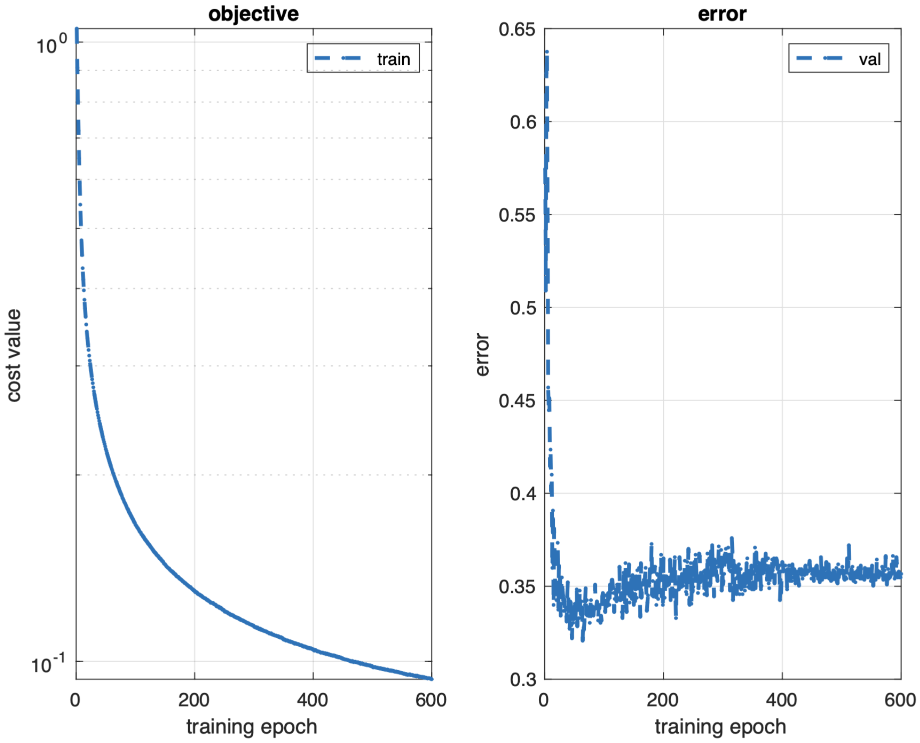


**Figure 26:** A U-SPDNet-6BiRe model trained and tested on JSE Top 60 data partitioned as in Figure 24 over 600 epochs converges on an OOS regime label accuracy of 65.01%. We parameterise this model as in Table 1.

Interestingly, upon closer inspection of the confusion matrices as in Figure 27 and Figure 28, it is clarified that the superior prediction accuracy of the SPDNet OOS is a product of simply converging on a corner solution. Despite generating many additional training samples to mitigate class imbalance resulting in inaccurate predictions, unfortunately the SPDNet fails in learning a representation of the hierarchical structure underpinning market evolution in different regimes, as evidenced by its learning to place most test observations into the over-sampled normal market regime. Thus, despite the U-SPDNet achieving lower regime detection accuracy, it is likely that it has learned an improved information-geometric representation of the market's state-dependent hierarchical structure, which leads to greater detection accuracy in the stressed and rally class instances.

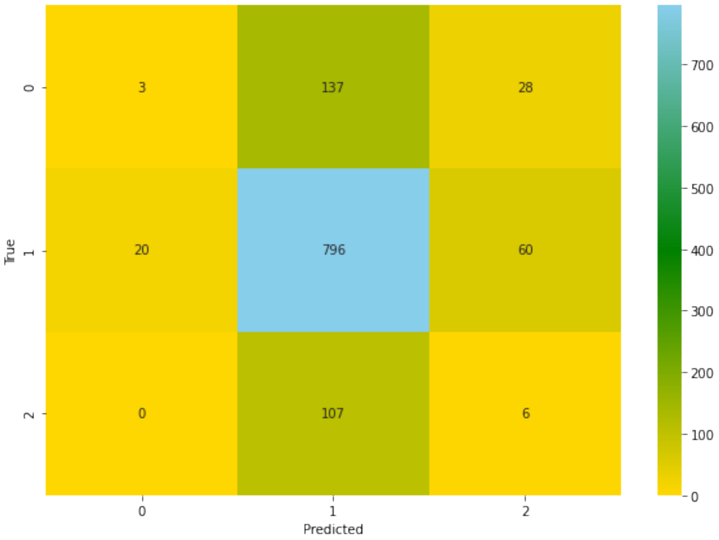


**Figure 27:** The SPDNet confusion matrix depicts the model's estimated regime labels as compared to the SR-labeled correlation matrices computed on the test data set of JSE Top 60 returns over the period January 2008 - mid-July 2012. Proportionally, the model makes most accurate predictions of normal regime labels, and inaccurately classifies stressed and rally labels as normal regimes in the majority of cases.

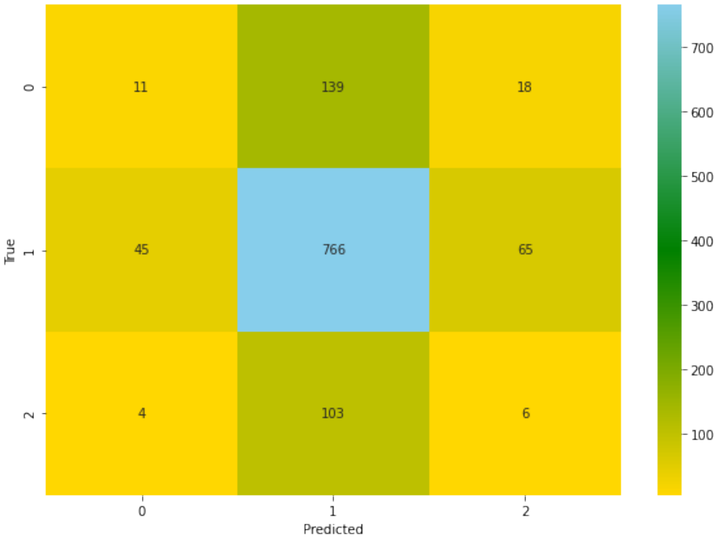


**Figure 28:** The U-SPDNet confusion matrix depicts the model's estimated regime labels as compared to the SR-labeled correlation matrices computed on the test data set of JSE Top 60 returns over the period January 2008 - mid-July 2012. Proportionally, the model makes most accurate predictions of normal regime labels.Relative to the SPDNet-identified regimes, the stressed and rally regimes show improved predictive accuracy at the expense of marginally lower accuracy for normal regime detection when comparing the model's predictions to the true SR-determined labels OOS.

More frequent regime switching is observed in the market regime estimations made by the U-SPDNet, as learned in comparison of Figure 29 and Figure 30. Over the January 2008 - mid-July 2012 test period the SPDNet detects more protracted periods characterised by normal regime labels, whereas U-SPDNet identifies a greater interspersion of regimes. Both models are consistent in identifying greater concentration of stressed regimes during the 2008 GFC, but the U-SPDNet is able to capture improved information which translates into the identification of both more stressed and rally classes during time frames otherwise classified as simply normal by the SPDNet.

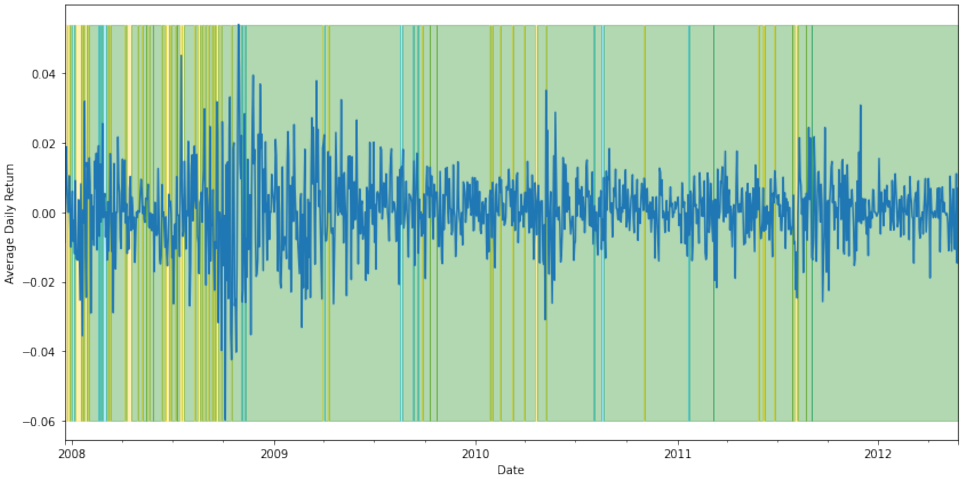


**Figure 29:** The market states identified by the SPDNet are super-imposed on the test data JSE Top 60 average daily returns. The following associations are made: yellow 'stressed', green 'normal'; blue 'rally'. It is observed that these colour bars do indeed correspond with the market's price returns over the test period. Yellow bars exist predominantly during the higher-volatility period of the 2008 GFC, whilst green and blue bars align with data points with lower volatility and greater average returns.

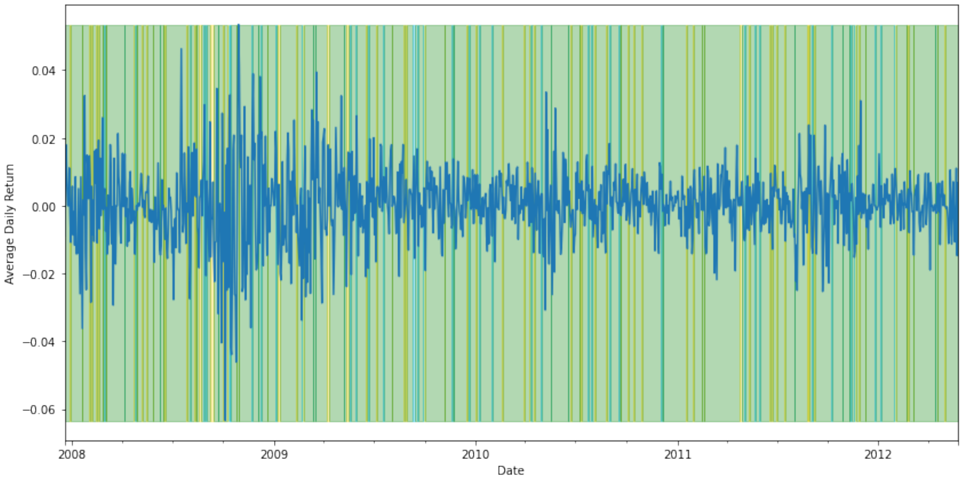


**Figure 30:** The market states identified by the U-SPDNet are super-imposed on the test data JSE Top 60 average daily returns. The following associations are made: yellow 'stressed', green 'normal'; blue 'rally'. In contrast to the regime predictions made by the SPDNet, the U-SPDNet identifies a greater interspersion of classes. More switching is identified with shorter periods of only 'normal' regimes being detected.

### 7.4.2 Backtesting & Performance

The backtest methodology outlined at the outset of Section 7.4 is implemented and the cumulative returns associated with each of the three optimisation strategies depicted in Figure 31 below.

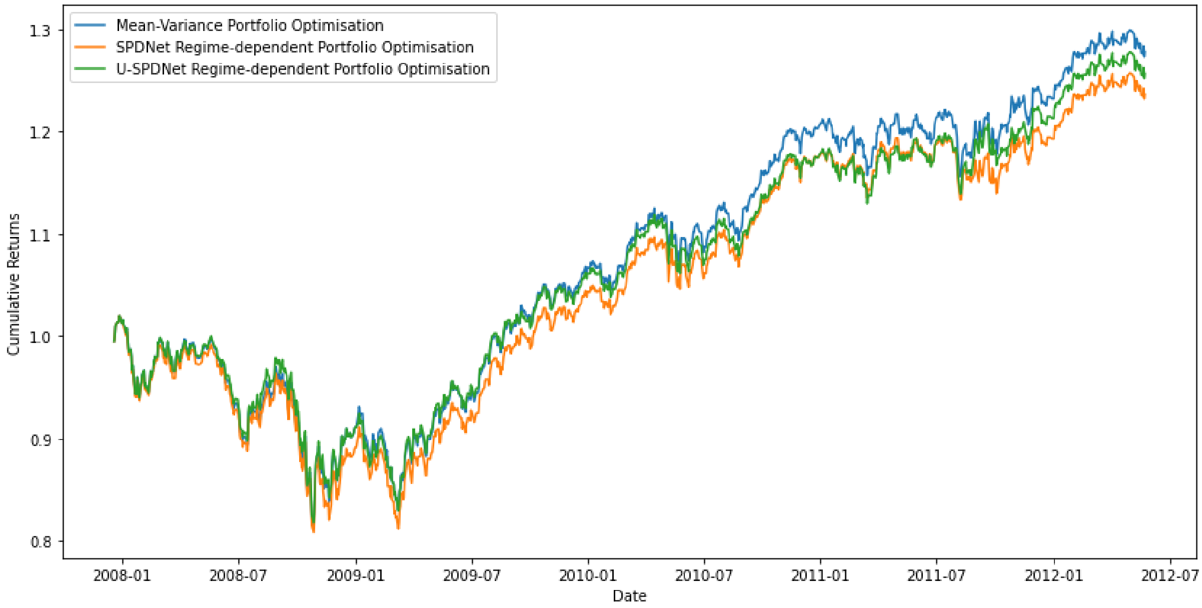


**Figure 31:** We perform daily-rebalanced portfolio optimisation backesting on the OOS JSE Top 60 test data set from January 2008 - mid-July 2012 without considering trading costs. Before 2009, the U-SPDNet regime-dependent strategy outperforms both the benchmark mean-variance and SPDNet regime-dependent strategies during the stressed market periods associated with the GFC. Thereafter, the U-SPDNet regime-dependent strategy outperforms its SPDNet counterpart, whilst the mean-variance portfolio delivers greatest cumulative returns overall. The portfolio performance metrics are shown in Table 3.

Figure 31 depicts the comparative cumulative performance of the three portfolio optimisation strategies implemented and tested on the real-world JSE Top 60 dataset. Initially, the U-SPDNet delivers greatest returns; this regime-dependent daily rebalancing approach marginally outperforms the Mean-Variance portfolio until 2010. Thereafter, the Mean-Variance benchmark optimisation consistently outperforms both regime-dependent strategies. The SPDNet-informed regime-dependent optimisation performs worse particularly during market rally periods, which corresponds to this category being significantly under-detected by the model in the OOS period as shown in Figure 27. It is likely that, despite the U-SPDNet achieving lower classification accuracy OOS, that it is superior to the SPDNet in identifying regimes other than solely 'normal'. The SPDNet, in providing a corner solution, is in fact less accurate in the market regimes which are in fact of greater relevance to regime-dependent portfolio optimisation. The greater sensitivity of the U-SPDNet to market dynamic information in different regimes is reflected in the U-SPDNet regime-dependent strategy delivering a return of 0.25698 with associated annualised risk of 0.09542 being the lowest of the three strategies, as shown in Table 3. Furthermore, Table 3 shows the annualised risk-adjusted and downside risk-adjusted returns of the U-SPDNet strategy improve on those associated with the SPDNet strategy, although they do not surpass those of the benchmark Mean-Variance portfolio optimisation.

**Table 3:** Portfolio Performance Metrics. Here the different mean-variance backtests are compared using annualised Sharpe ratios and Sortino ratios. The Sortino ratio computes the risk using the shortfall return standard deviation only. The Sharpe ratio uses both positive and negative returns in the standard deviation (risk) computation. The risk is the annualised standard deviation, and the returns are cumulative (compounded) annual returns. The Naïve Mean-Variance portfolio construction is compared to the Regime-Dependent (RD) Mean-Variance (MV) simulations. The backtest time series are presented in Figure 31. This initial simulation work suggests that there is little meaningful value in the correlation matrix predication relative to a naïve mean-variance portfolio.

| **Portfolio** | **Return** | **Risk** | **Sharpe Ratio** | **Sortino Ratio** |
|---|---|---|---|---|
| Mean-Variance MV | 0.27815 | 0.09782 | 0.59543 | 0.81252 |
| SPDNet RD-MV | 0.23682 | 0.09709 | 0.52544 | 0.70958 |
| U-SPDNet RD-MV | 0.25698 | 0.09542 | 0.56986 | 0.77137 |

# 8 Conclusions

Financial markets are complex systems characterised by dynamics present in both time and space dimensions. Their emergent, self-organising properties give rise to hierarchies which co-evolve with different macroeconomic regimes. The correlation matrices encoding reflexive relationships within traded asset networks are of particular interest due to their representation of the information geometry underpinning these multi-scale temporal dynamics.

Families of evolving correlation matrices can act as a useful tool in further developing our understanding of the dynamics at work in real financial markets, particularly in the presence of bottom-up and top-down noise, and self-organising effects corresponding to differing regimes. Correlation matrices computed from traded asset returns possess interesting empirical properties, specifically: i.) a nested block hierarchical structure (Marti et al., 2016); ii.) an information geometry which appears to allow a Riemannian manifold representation (Huang and Gool, 2016); and iii.) asymmetries in correlations amidst inter- and intra-cluster asset groups in different market regimes (Miori and Cucuringu, 2022).

This suggests that nested block hierarchies characterise the structure of traded financial assets, and that this stylised fact can captured by their SPD correlation matrices; but, crucially, that these matrices do not reside in a Euclidean space. This has suggested two broad research aims for representation learning algorithm development in finance (Marti et al., 2016): 1.) representation learning of stable features underpinning market dynamics such that the hierarchical SPD manifold information structure is preserved, and 2.) the construction of realistic synthetic correlation matrices.

Here, these matrices are the focal point for our implementation of methods aimed at understanding hierarchical dynamics underlying asset returns indicative of differing macroeconomic regimes. The project investigates the robustness of deep learning models borrowed from the realm of Computer Vision which specifically account for the SPD Riemannian manifold upon which the SPD correlation matrices exist; namely the SPDNet, SPDNetBN and U-SPDNet, which belong to the broad classes of Riemannian NNs and AEs respectively (Wang et al., 2022b; Brooks et al., 2019a; Huang and Gool, 2016). Specifically, we assess these models from the perspectives of causal link embedding, counterfactual data and standard sampling practices used in data science as compared to the quantitative finance realm.

Practical tests reveal that the SPDNet algorithm, despite achieving greater OOS accuracy, converges on a corner solution involving estimating the regime with the greatest representation, 'normal', which implies the underlying hierarchical latent feature structure is in fact not being learnt, as is the case for the U-SPDNet. Despite the regimes identified by the U-SPDNet being less accurate in the 'normal' class), it does perform better in the instances of 'stressed' and 'rally' regimes thus resulting in superior returns.

This is an important insight into the efficacy of the algorithms being tested and suggests that greater value may be realised by considering market regimes as characterised by fragility; specifically, expanding on techniques which aim to learn the deep hierarchical structure underpinning market dynamics that can replace explicit analysis of the correlation matrices themselves.

## 8.1 Exploratory Data Analysis

Section 4.1 corroborates previous works describing stochastic asset market structural dynamics. Indeed, the JSE Top 60 correlation matrices organised in accordance with a single-linkage Ward-based algorithm using the distance measure of $\sqrt{2(1-\mathbf{C})}$ as in Papenbrock et al. (Spring 2021) display a clear block correlation structure as presented by Mantegna (1999b). Furthermore, the data, when allocated to a regime according to the contemporaneous SR, display a trimodal correlation coefficient distribution, shown in Figure 11.

Figure 10 supports findings of the block hierarchical structure of financial market correlations maintaining their structure but displaying stronger average inter- and intra-cluster cluster strengths in stressed regimes. Finally, Figure 7 provides anecdotal evidence for the presence of causal feedback spatio-temporal dynamics at work in the JSE Top 60 data. A clear block hierarchical structure is present indicating different regimes' presence within the January 2000 - December 2023 time frame, which aligns with the results of Miori and Cucuringu (2022)'s work.

## 8.2 Experimental Outcomes

Drawing on the work of Marti (2020a), the effectiveness of the deep Riemannian representation learning models in accurately classifying correlation matrices into three different regimes - 'normal', 'stressed' and 'rally' - is first evaluated using BBG JSE Top 60 price returns data alongside synthetically-generated nested block hierarchical correlation matrices constructed by virtue of the Yelibi and Gebbie (2021)-adapted technique first proposed by Tumminello et al. (2007). These data are randomly divided between training and validation sets.

The objective of performing model training and testing using synthetic data is two-fold. Firstly, to provide 'counterfactual' scenarios to check model robustness to unseen events, as in the work of Papenbrock et al. (Spring 2021); and secondly, as the synthetic data possess no lagged inter-matrix relationships, to provide a benchmark for identifying the extent to which temporal reflexive correlation dynamics contribute to model accuracy in regime identification.

The comparison of Figure 14 and Figure 15 with Figure 18 confirms that the classification accuracies achieved by the SPDNet and SPDNetBN models when applied to the JSE Top 60 data converge at a significantly lower level than that of the U-SPDNet, despite their conducting SPD matrix learning preserving the Riemannian geometry of the input correlation matrices. Little to no improvement in accuracy of market detection is seen with an increase in depth of the original SPDNet, agnostic of whether used in conjunction with RBN. Table 1 thus confirms the findings of Wang et al. (2022b).

Thus block resampling techniques which preserve the chronology of OOS data are invoked in Section 6 to test the extent to which the deep learning of temporal correlation relationships impacts on model classification performance for the benchmark SPDNet model using 5871 data windows. Despite chronology being maintained, it becomes clear that the SPDNet is only capable of learning a stable latent representation of the market which holds in the majority of the timeframe under consideration, but degrades in terms of information particular to stressed and rally market phases (Soros, 2013). Figure 23 points to the importance of lead-lag dynamics in deep learning model accuracy which are not captured by the SPDNet.

The U-SPDNet with six BiRe modules out-performs the SPDNet and SPDNetBN models in terms of classification accuracy of the BBG JSE Top 60 correlation matrices into either stressed, normal or rally regimes. Table 1 shows that detection accuracy of the U-SPDNet baseline model on unseen real-world data converges on 98.90% averaged over the validation epochs.

The fundamental reason for this significant U-SPDNet improvement in regime detection performance is quoted as being due to the design enhancements made to the baseline SPDNet architecture by Huang and Gool (2016); specifically, the introduction of the LFE modules as shortcut connections to increase information pass-through from the descending to ascending layers of the model, coupled with an end-to-end deep non-linear architecture. However, it is important that the case of financial markets data is different to that of MRI or radar data where causal temporal links are of little significance in correlation dynamics.

Comparatively, the U-SPDNet does not significantly improve on the market regime detection performance of the baseline SPDNet or SPDNet with RBN when used in the context of the synthetically-generated block hierarchical SPD correlation matrices. Indeed, considering Figure 19 reveals convergence of the validation set classification error to a value of 65.2% whilst the SPDNet and SPDNetBN achieve 66.8% and 65.92% respectively.

These findings and discrepancies may be explained by the following factors:

1. The U-SPDNet places emphasis on learning statistical features as opposed to the encoded temporal changes between different market regimes, which results in poor pattern-related representational capacity of the learned features.
2. The SPD correlation matrices computed in both the original JSE Top 60 and synthetic data cases are learned without consideration being made for the sequence of the data themselves, meaning that error and implicit data volatility information is not eliminated in the end-to-end learning process.
3. Since the data are randomly shuffled prior to model evaluation, there is a high risk of information leakage amidst the training and validation correlation matrix datasets. This would contribute to the significantly greater accuracy achieved by the U-SPDNet when trained on the real-world JSE data. This challenge is addressed using the *purging* and *embargo* techniques introduced in Section 7.3 above (Lopez De Prado Lopez, 2022).
4. The data 'scrambling' would induce the loss of all dependence information (Politis, 2003).
5. Finally, by construction no temporal relationship exists between correlation matrices in the synthetic dataset. The performance of U-SPDNet being similar to that of the SPDNet indicates that without the spatio-temporal information feedback loops present in real-world data, the model's estimation accuracy is negatively affected by the lack of this non-linear causal dynamic, despite the encoded presence of hierarchical structures whose underlying distributions vary per regime.

We aim to further dissect the competing influences of data leakage and effective deep learning of causal feedback loops in market participant behaviour driving emerging block hierarchical correlation matrix structures through market phases in the robustness of the U-SPDNet model. We conduct this by implementing simple MVPO and compare the performance of Mean-Variance SPDNet and U-SPDNet regime prediction dependent portfolio construction techniques with OOS backtesting.

## 8.3 Regime-Dependent Asset Allocation

Given the promising state prediction accuracies recovered by the deep learning models, namely the SPDNet, SPDNet-3BiRe, SPDNetBN, SPDNetBN-3BiRe and the U-SPDNet-6BiRe models trained and tested on randomised input SPD matrices possessing inherent block hierarchical structure, an additional application is justified in the context of quantitative finance: state-dependent asset allocation.

While feature importance analysis is a research tool of far greater use, backtesting is implemented here for illustrative purposes in order to determine whether superior value is delivered by state-dependent asset allocation strategies using the regimes estimated by SPDNet and U-SPDNet-6BiRe models as compared to a simple Mean-Variance asset allocation procedure.

A baseline SPDNet model and best-performing U-SPDNet model with six BiRe modules were trained on JSE Top 60 data which first underwent purging and embargo techniques to prevent information leakage as in Figure 24. The models were then implemented in estimating market regimes in the chronological OOS test dataset using correlation matrices computed from data over the time-frame of January 2007 - mid-July 2012.

Importantly, this practical test reveals that the SPDNet, despite achieving greater OOS accuracy, is in fact converging on a corner solution involving estimating the regime with the greatest representation, 'normal', which implies the underlying hierachical latent feature structure is in fact not being learnt, as is the case for the U-SPDNet. Despite the regimes identified by the U-SPDNet being less accurate in the 'normal' class, it does perform better in the instances of 'stressed' and 'rally' regimes, thus resulting in superior returns. These class-based accuracies are described in Figures 27 and 28. The superiority of the U-SPDNet in preventing the degradation of structural information in the learning process with a superior representational capacity is confirmed in this regime-dependent optimisation's greater return profile.

Figure 31 depicts the U-SPDNet regime-dependent strategy resulting in greater returns than the SPDNet case. Both strategies under-perform the daily-rebalanced benchmark Mean-Variance portfolio implementation which provides motivation for improving on the rudimentary labeling strategy adopted and rather employing unsupervised techniques in this regard where the data provides the regime labels itself.

## 8.4 Concluding Remarks

Some important conclusions may be drawn from the set of experiments and model interrogation techniques we have implemented:

1. Experimental outcomes point to the importance of financial markets data context-specific model robustness evaluation. Accuracy fails in this respect, as is confirmed in Section 7, where we show that despite the SPDNet making OOS regime predictions with greater accuracy that its U-SPDNet counterpart, this is in fact due to rampant overfitting. When performance is considered from the perspective of financial markets being complex systems where regimes are influenced by causal feedback loops in participant behaviour and associated correlation structures, it may be concluded that the U-SPDNet is in fact performing superior learning of spatio-temporal dynamics as it has greater accuracy when making predictions during stressed and rally regimes.

2. A significant lead-lag relationship exists between market volatility and model regime detection accuracy OOS as shown in Figure 23. It is widely-known that volatility increases during bull and bear markets, which implies that temporal correlation structure relationships are indeed important in robust latent market structure learning. This provides probable explanation of the superiority of the U-SPDNet over the SPDNet due to its attempt to solve the information degradation problem in the context of financial markets data (Huang and Gool, 2016). In this respect the DreamNet proposed by Wang et al.

(2022a) could also to be considered due to its hierarchical architecture and information flow adaptations better-reflecting latent market structure.

3. The counterfactual dataset, which possesses no lead-lag correlation matrix evolution relationship, provides the benchmark for understanding the improved U-SPDNet performance OOS on JSE Top 60 data as being attributable to increased sensitivity to information-geometric dynamics present in the real-world data, as corroborated by similar performance to the SPDNet when validated on synthetic data, as opposed to an accuracy increase on correlation matrices constructed from traded price returns.
4. Financial markets correlation matrix data in semi-equilibrium necessitates processing techniques for model learning such as those proposed by Lopez De Prado Lopez (2022) (Purgeing and Embargo) to prevent overfitting and information leakage, which is not present in computer vision or image classification contexts.
5. Increased SPDNet model depth does not correlate with improved latent feature extraction or market regime classification accuracy, agnostic of the introduction of RBN, confirming the findings of Brooks et al. (2019a).
6. Regime dependent MVPO delivers greater OOS returns when predictions made by the U-SPDNet are used in favour of those made by the SPDNet model. This simple experiment confirms the greater deep representation learning efficacy of the U-SPDNet in the practical context of portfolio construction strategy selection.

Further research areas are identified as follows, which would extend on this work:

1. The importance of the data labelling process is highlighted by the somewhat arbitrary use of the SR in our experiments. Specifically, this research may be enhanced by unsupervised clustering techniques being employed for a more intuitive market regime labeling scheme as in the work of Miori and Cucuringu (2022).
2. The generation of realistic synthetic data is imperative in model robustness testing in counterfactual scenarios. Techniques proposed by Papenbrock et al. (Spring 2021) involving matrix evolutions, or those of Marti (2020b) utilising the CorrGAN approach may be considered to augment the data set.
3. From the perspective of causal modelling, the incorporation of alternative datasets which provide additional information quantifying the reflexive behavioural mechanisms of market participants influencing the volatility and block hierarchical correlation structures emerging in different market regimes may be used to guide model architecture to better-reflect latent market structure for improved representation learning. For example, financial newsfeeds and participant commentary.
4. In terms of MVPO: Approaching the problem space from the perspective that risk is yet to be quantified in tractable ways, greater value may be realised by considering market regimes as characterised by *fragility*; specifically, expanding on techniques which aim to learn the deep hierarchical structure underpinning market dynamics can replace explicit analysis of the correlation matrix on a sector-by-sector basis within the index to reveal which sectors are contributing most to overall market fragility. Sectors with high internal

correlations and high correlations with the broader market could be more sensitive to systemic risk. This knowledge can be incorporated into portfolio optimisation aiming to cap downside volatility sensitivity and contemporaneously maximise on upside volatility sensitivity in machine learning-driven investment strategy implementation (Taleb, 2022). This further points to the use of market regime detection based on the hierarchical information encoded in the correlation matrices themselves as in Miori and Cucuringu (2022).

# 9 Bibliography

Absil, P.A., Mahony, R., Sepulchre, R., 2010. Optimization algorithms on matrix manifolds. Foundations of computational mathematics 10, 241–245. URL: `https://www.jstor.org/stable/j.ctt7smmk`.

Akioyamen, P., Tang, Y.Z., Hussien, H., 2020. A hybrid learning approach to detecting regime switches in financial markets. Proceedings of the First ACM International Conference on AI in Finance URL: `http://dx.doi.org/10.1145/3383455.3422521`, doi:`10.1145/3383455.3422521`.

Amari, S., Tsuchiya, N., Oizumi, M., 2018. Information Geometry and Its Applications: On the Occasion of Shun-Ichi Amari's 80th Birthday, IGAIA IV Liblice, Czech Republic, June 2016. volume 252 of *Springer proceedings in mathematics & statistics*. Springer International Publishing AG, Cham.

Ando, T., Li, R., Mathias, C.K., 2004. Geometric means. Linear algebra and its applications 385, 305–334. doi:`https://doi.org/10.1016/j.laa.2003.11.019`.

Ang, A., Bekaert, G., 2004. How regimes affect asset allocation. Financial analysts journal 60, 86–99. doi:`10.2469/faj.v60.n2.2612`.

Arsigny, V., Fillard, P., Pennec, X., Ayache, N., 2007. Geometric means in a novel vector space structure on symmetric positive-definite matrices. SIAM journal on matrix analysis and applications 29, 328–347. doi:`10.1137/050637996`.

Atkinson, C., Mitchell, A.F.S., 1981. Rao's distance measure. Sankhyā: The Indian Journal of Statistics, Series A 43, 345–365. URL: `https://www.jstor.org/stable/25050283`.

Balakrishnan, S., Xu, M., Krishnamurthy, A., Singh, A., 2011. Noise Thresholds for Spectral Clustering. volume 24. Curran Associates, Inc. URL: `https://proceedings.neurips.cc/paper_files/paper/2011/file/dbe272bab69f8e13f14b405e038deb64-Paper.pdf`.

Barbaresco, F., 2019. Jean-Louis Koszul and the Elementary Structures of Information Geometry. Signals and Communication Technology, Springer International Publishing, Cambridge. doi:`10.1007/978-3-030-02520-5_12`.

Bhatia, R., Holbrook, J., 2006. Riemannian geometry and matrix geometric means. Linear algebra and its applications 413, 594–618. doi:`10.1016/j.laa.2005.08.025`.

Bollerslev, T., Ole Mikkelsen, H., 1996. Modeling and pricing long memory in stock market volatility. Journal of econometrics 73, 151–184. doi:`h10.1016/0304-4076(95)01736-4`.

Bonanno, G., Vandewalle, N., Mantegna, R.N., 2000. Taxonomy of stock market indices. Physical Review. E 62, 7615–7618. doi:`10.1103/PhysRevE.62.R7615`.

Bongiorno, C., Miccichè, S., Mantegna, R.N., 2022. Statistically validated hierarchical clustering: Nested partitions in hierarchical trees. Physica A: Statistical Mechanics and its Applications 593, 126933. doi:`10.1016/j.physa.2022.126933ff`.

Bonnabel, S., Sepulchre, R., 2010. Riemannian Metric and Geometric Mean for Positive Semidefinite Matrices of Fixed Rank. SIAM Journal on Matrix Analysis and Applications 31, 1055–1070. doi:10.1137/080731347.

Borghesi, C., Marsili, M., Miccichè, S., 2007. Emergence of time-horizon invariant correlation structure in financial returns by subtraction of the market mode. Phys. Rev. E 76, 26–104. URL: https://link.aps.org/doi/10.1103/PhysRevE.76.026104, doi:10.1103/PhysRevE.76.026104.

Bouchaud, J.P., Potters, M., 2011. Financial applications of random matrix theory: a short review. The oxford handbook of random matrix theory , 824–850URL: https://doi.org/10.48550/arXiv.0910.1205.

Brooks, D.A., Schwander, O., Barbaresco, F., Schneider, J., Cord, M., 2019a. Riemannian batch normalization for SPD neural networks. CoRR abs/1909.02414. URL: http://arxiv.org/abs/1909.02414, arXiv:1909.02414.

Brooks, D.A., Schwander, O., Barbaresco, F., Schneider, J., Cord, M., 2019b. Second-order networks in pytorch. Geometric Science of Information 11712. URL: https://doi.org/10.1007/978-3-030-26980-7_78.

Cybenko, G.V., 1989. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems 2, 303–314. URL: https://api.semanticscholar.org/CorpusID:3958369.

Dolphin, R., Smyth, B., Dong, R., 2023. Stock embeddings: Representation learning for financial time series. Engineering Proceedings 39. URL: https://www.mdpi.com/2673-4591/39/1/30, doi:10.3390/engproc2023039030.

Edelman, A., Arias, T.A., Smith, S.T., 2006. The geometry of algorithms with orthogonality constraints. SIAM Journal on Matrix Analysis and Applications 20, 303–353. URL: https://doi.org/10.1137/S0895479895290954, doi:10.1137/S0895479895290954.

Engin, M., Wang, L., Zhou, L., Liu, X., 2018. Deepkspd: Learning kernel-matrix-based spd representation for fine-grained image recognition 11206, 629–645. URL: https://doi.org/10.48550/arXiv.1711.04047.

Fletcher, P.T., Venkatasubramanian, S., Joshi, S., 2009. The geometric median on Riemannian manifolds with application to robust atlas estimation. NeuroImage (Orlando, Fla.) 45, 143–152. doi:10.1016/j.neuroimage.2008.10.052.

Hendricks, D., Gebbie, T., Wilcox, D., 2016. Detecting intraday financial market states using temporal clustering. Quantitative Finance 16, 1657–1678. URL: https://doi.org/10.1080/14697688.2016.1171378, doi:10.1080/14697688.2016.1171378, arXiv:https://doi.org/10.1080/14697688.2016.1171378.

Hu, W., Kercheval, A.N., 2010. Portfolio optimization for student t and skewed t returns. Quantitative Finance 10, 91–105. doi:10.1080/14697680902814225.

Huang, Z., Gool, L.V., 2016. A Riemannian Network for SPD Matrix Learning. CoRR 1608.04233. arXiv:1608.04233.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. doi:10.48550/arXiv.1502.03167, arXiv:1502.03167.

Ionescu, C., Vantzos, O., Sminchisescu, C., 2016. Training deep networks with structured layers by matrix backpropagation. doi:doi.org/10.48550/arXiv.1509.07838, arXiv:1509.07838.

Laloux, L., Cizeau, P., Bouchaud, J.P., Potters, M., 1999. Noise dressing of financial correlation matrices. Phys. Rev. Lett. 83, 1467–1470. URL: https://link.aps.org/doi/10.1103/PhysRevLett.83.1467, doi:10.1103/PhysRevLett.83.1467.

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324. doi:10.1109/5.726791.

Lopez De Prado Lopez, M.M., 2022. Tactical investment algorithms through Monte Carlo backtesting. USPTO US20210082046A1. URL: https://patents.google.com/patent/US20210082046A1/en.

Mantegna, R., 1999a. Information and hierarchical structure in financial markets. Computer Physics Communications 121-122, 153–156. doi:10.1016/S0010-4655(99)00302-1. proceedings of the Europhysics Conference on Computational Physics CCP 1998.

Mantegna, R.N., 1999b. Hierarchical structure in financial markets. The European physical journal. B, Condensed matter physics 11, 193–197. doi:10.1007/s100510050929.

Markowitz, H., 1952. Portfolio selection. The Journal of Finance 7, 77–91. doi:10.2307/2975974.

Marletto, C., 2021. The Science of Can and Can't: A Physicist's Journey through the Land of Counterfactuals. Viking, Oxford.

Marsili, M., 2002. Dissecting financial markets: sectors and states. Quantitative Finance 2, 297–302. doi:10.1088/1469-7688/2/4/305.

Marti, G., 2020a. Can we predict a market regime from correlation matrix features? URL: https://gmarti.gitlab.io/qfin/2020/09/04/correlation-matrix-features-market-regimes.html. accessed on: May 20, 2024.

Marti, G., 2020b. Corrgan: Sampling realistic financial correlation matrices using generative adversarial networks , URL: http://dx.doi.org/10.1109/ICASSP40776.2020.9053276, doi:10.1109/icassp40776.2020.9053276.

Marti, G., 2021. Classification of correlation matrices using spdnet with riemannian batch normalization. URL: https://marti.ai/ml/2021/01/22/spdtorchnet.html. accessed on Aug 8, 2024.

Marti, G., Nielsen, F., Bińkowski, M., Donnat, P., 2021. A Review of Two Decades of Correlations, Hierarchies, Networks and Clustering in Financial Markets. Springer International Publishing, Cham. chapter 10. pp. 245–274. URL: https://doi.org/10.1007/978-3-030-65459-7_10, doi:10.1007/978-3-030-65459-7_10.

Marti, G., Nielsen, F., Donnat, P., Andler, S., 2016. On clustering financial time series: a need for distances between dependent random variables. arXiv:1603.07822.

Miori, D., Cucuringu, M., 2022. Returns-driven macro regimes and characteristic lead-lag behaviour between asset classes. arXiv:2209.00268.

Nystrup, P., Madsen, H., Lindström, E., 2018. Dynamic portfolio optimization across hidden market regimes. Quantitative Finance 18, 83–95. doi:10.1080/14697688.2017.1342857.

Orton, A., Gebbie, T., 2024. 'bh-spdnet'. URL: https://github.com/AlexaOrton/BH-SPDNet. accessed on Apr 1, 2024.

Papenbrock, J., Schwendner, P., Jaeger, M., Krügel, S., Spring 2021. Matrix evolutions: Synthetic correlations and explainable machine learning for constructing robust investment portfolios. The Journal of Financial Data Science 3, 51–69. doi:10.3905/jfds.2021.1.056.

Pennec, X., Fillard, P., Ayache, N., 2006. A Riemannian Framework for Tensor Computing. International journal of computer vision 66, 41–66. doi:10.1007/s11263-005-3222-z.

Pinele, J., Costa, S.I.R., Strapasson, J.E., 2019. On the fisher-rao information metric in the space of normal distributions, in: Nielsen, F., Barbaresco, F. (Eds.), Geometric Science of Information, Springer International Publishing, Cham. pp. 676–684. doi:10.1007/978-3-030-26980-7_70.

Polakow, D., Gebbie, T., Flint, E., 2023. Epistemic limits of empirical finance: Causal reductionism and self-reference. SSRN 33. URL: https://ssrn.com/abstract=4646664, doi:http://dx.doi.org/10.2139/ssrn.4646664.

Politis, D.N., 2003. The Impact of Bootstrap Methods on Time Series Analysis. Statistical Science 18, 219–230. URL: http://www.jstor.org/stable/3182852.

Popper, K., 1944. The Poverty of Historicism. Economica 11, 86–103. doi:10.2307/2549642.

López de Prado, M., 2018. Advances in financial machine learning. volume 33. Wiley, New York. URL: http://dx.doi.org/10.2139/ssrn.3257420.

Rodriguez Dominguez, A., 2023. Portfolio optimization based on neural networks sensitivities from assets dynamics respect common drivers. Machine Learning with Applications 11, 100447. doi:10.1016/j.mlwa.2022.100447.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597.

Rui, C., Jorge, B., Joao, C., Cristian, S., 2015. Semantic segmentation method with second-order pooling. USPTO US20150104102A1. URL: https://patents.google.com/patent/US20150104102A1/en.

Said, S., Bombrun, L., Berthoumieu, Y., Manton, J.H., 2017. Riemannian Gaussian Distributions on the Space of Symmetric Positive Definite Matrices. IEEE Transactions on Information Theory 63, 2153–2170. doi:10.1109/TIT.2017.2653803.

Singo, U., 2022. Market state discovery. Masters dissertation. Department of Statistical Sciences, University of Cape Town. URL: https://open.uct.ac.za/items/47e5a198-ef49-4b0c-be14-071dc747c45a.

Soros, G., 2013. Fallibility, reflexivity, and the human uncertainty principle. Journal of Economic Methodology 20, 309–329. doi:`10.1080/1350178X.2013.859415`.

Sra, S., 2013. Positive definite matrices and the s-divergence. URL: `https://arxiv.org/abs/1110.1773`, `arXiv:1110.1773`.

Taleb, N.N., 2022. Statistical consequences of fat tails: Real world preasymptotics, epistemology, and applications. `arXiv:2001.10488`.

Tumminello, M., Lillo, F., Mantegna, R.N., 2007. Hierarchically nested factor model from multivariate data. Europhysics Letters (EPL) 78, 30006. URL: `https://doi.org/10.1209/0295-5075/78/30006`, doi:`10.1209/0295-5075/78/30006`.

Tumminello, M., Lillo, F., Mantegna, R.N., 2010. Correlation, hierarchies, and networks in financial markets. Journal of Economic Behavior & Organization 75, 40–58. doi:`10.1016/j.jebo.2010.01.004`.

Wang, R., Wu, X.J., Chen, Z., Xu, T., Kittler, J., 2022a. Dreamnet: A deep riemannian network based on spd manifold learning for visual classification. `arXiv:2206.07967`.

Wang, R., Wu, X.J., Xu, T., Hu, C., Kittler, J., 2022b. U-spdnet: An spd manifold learning-based neural network for visual classification. Neural Networks 161, 382–396. URL: `https://www.sciencedirect.com/science/article/pii/S0893608022004713`, doi:`https://doi.org/10.1016/j.neunet.2022.11.030`.

Wilcox, D., Gebbie, T., 2014. Hierarchical causality in financial economics. SSRN Electronic Journal URL: `https://doi.org/10.48550/arXiv.1408.5585`, doi:`10.2139/ssrn.2544327`.

Yair, O., Ben-Chen, M., Talmon, R., 2019. Parallel transport on the cone manifold of spd matrices for domain adaptation. IEEE Transactions on Signal Processing 67, 1797–1811. doi:`10.1109/TSP.2019.2894801`.

Yang, L., 2010. Riemannian median and its estimation. LMS Journal of Computation and Mathematics 13, 461–479. URL: `http://dx.doi.org/10.1112/S1461157020090531`, doi:`10.1112/s1461157020090531`.

Yang, L., Arnaudon, M., Barbaresco, F., 2010. Riemannian median, geometry of covariance matrices and radar target detection, in: The 7th European Radar Conference, IEEE. pp. 415–418. URL: `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5615027`.

Yelibi, L., 2021. timeseries_generator. URL: `https://github.com/lyelibi/timeseries_generator`. accessed on Nov 12, 2023.

Yelibi, L., Gebbie, T., 2021. Agglomerative likelihood clustering. Journal of Statistical Mechanics: Theory and Experiment 2021, 113408. URL: `https://doi.org/10.1088/1742-5468/ac3661`, doi:`10.1088/1742-5468/ac3661`.

Yuan, X., Huang, W., Absil, P.A., Gallivan, K.A., 2020. Averaging Symmetric Positive-Definite Matrices. Springer International Publishing, Cham. URL: `https://www.math.fsu.edu/~whuang2/papers/ASPDM.htm`.

Zanini, P., Congedo, M., Jutten, C., Said, S., Berthoumieu, Y., 2018. Transfer Learning: A Riemannian Geometry Framework With Applications to Brain–Computer Interfaces. IEEE Transactions on Biomedical Engineering 65, 1107–1116. doi:10.1109/TBME.2017.2742541.

# 10 Appendices

## A JSE JALSH BBG Tickers

The BBG tickers of stocks found in the JSE Top 60 by ranked market capitalisation over the period of January 2000 - December 2023 are as follows:

**Table 4:** JSE JALSH BBG tickers and listing dates.

| **Ticker** | **Name** | **Listing Date** |
|---|---|---|
| *Communication Services (5 members)* | | |
| MTN SJ Equity | MTN Group Ltd | 19950701 |
| MCG SJ Equity | MultiChoice Group | 20190227 |
| BLU SJ Equity | Blue Label Telecoms Ltd | 20071114 |
| TKG SJ Equity | Telkom SA SOC Ltd | 20030304 |
| VOD SJ Equity | Vodacom Group Ltd | 20090518 |
| *Consumer Discretionary (19 members)* | | |
| WHL SJ Equity | Woolworths Holdings Ltd/South Africa | 19970701 |
| SPG SJ Equity | Super Group Ltd/South Africa | 19960701 |
| CSB SJ Equity | Cashbuild Ltd | 19860701 |
| PRX SJ Equity | Prosus NV | 20190911 |
| ITE SJ Equity | Italtile Ltd | 19880701 |
| CFR SJ Equity | Cie Financiere Richemont SA | 20230419 |
| MTH SJ Equity | Motus Holdings Ltd | 20181122 |
| TRU SJ Equity | Truworths International Ltd | 19980511 |
| TFG SJ Equity | Foschini Group Ltd/The | 19410602 |
| SUI SJ Equity | Sun International Ltd/South Africa | 19410602 |
| PPH SJ Equity | Pepkor Holdings Ltd | 20170920 |
| TSG SJ Equity | Tsogo Sun Limited | 20190612 |

Continued on next page

**Table 4 – continued from previous page**

| **Ticker** | **Name** | **Listing Date** |
|---|---|---|
| MRP SJ Equity | Mr Price Group Ltd | 19520701 |
| COH SJ Equity | Curro Holdings Ltd | 20110602 |
| SSU SJ Equity | Southern Sun Ltd | 20190612 |
| FBR SJ Equity | Famous Brands Ltd | 19940701 |
| ADH SJ Equity | Advtech Ltd | 19971120 |
| NPN SJ Equity | Naspers Ltd | 19940912 |
| *Consumer Staples (13 members)* | | |
| AVI SJ Equity | AVI Ltd | 19940101 |
| ANH SJ Equity | Anheuser-Busch InBev SA/NV | 20161011 |
| RCL SJ Equity | RCL Foods Ltd/South Africa | 19890615 |
| OCE SJ Equity | Oceana Group Ltd | 19470701 |
| SPP SJ Equity | SPAR Group Ltd/The | 20041018 |
| ARL SJ Equity | Astral Foods Ltd | 20010409 |
| TBS SJ Equity | Tiger Brands Ltd | 19440101 |
| PIK SJ Equity | Pick n Pay Stores Ltd | 19690101 |
| DCP SJ Equity | Dis-Chem Pharmacies Ltd | 20161118 |
| SHP SJ Equity | Shoprite Holdings Ltd | 19360101 |
| CLS SJ Equity | Clicks Group Ltd | 19960101 |
| BID SJ Equity | Bid Corp Ltd | 20160530 |
| BTI SJ Equity | British American Tobacco PLC | 20081028 |
| *Energy (2 members)* | | |
| EXX SJ Equity | Exxaro Resources Ltd | 20011126 |
| TGA SJ Equity | Thungela Resources Ltd | 20210607 |

Continued on next page

**Table 4 – continued from previous page**

| **Ticker** | **Name** | **Listing Date** |
|---|---|---|
| *Financials (22 members)* | | |
| SNT SJ Equity | Santam Ltd | 19640101 |
| AFH SJ Equity | Alexander Forbes Group Holdings Ltd | 20140724 |
| CPI SJ Equity | Capitec Bank Holdings Ltd | 20020218 |
| ABG SJ Equity | Absa Group Ltd | 19860301 |
| NY1 SJ Equity | Ninety One Ltd | 20200316 |
| N91 SJ Equity | Ninety One PLC | 20200316 |
| SLM SJ Equity | Sanlam Ltd | 19981130 |
| REM SJ Equity | Remgro Ltd | 20000926 |
| SBK SJ Equity | Standard Bank Group Ltd | 19670101 |
| INL SJ Equity | Investec Ltd | 19880301 |
| OMU SJ Equity | Old Mutual Ltd | 20180626 |
| JSE SJ Equity | JSE Ltd | 20060605 |
| FSR SJ Equity | FirstRand Ltd | 19980701 |
| MTM SJ Equity | Momentum Metropolitan Holdings | 20010921 |
| RNI SJ Equity | Reinet Investments SCA | 20171129 |
| CML SJ Equity | Coronation Fund Managers Ltd | 20030613 |
| OUT SJ Equity | OUTsurance Group Ltd | 20110307 |
| KST SJ Equity | PSG Financial Services Ltd | 20140618 |
| NED SJ Equity | Nedbank Group Ltd | 19690101 |
| DSY SJ Equity | Discovery Ltd | 19991021 |
| QLT SJ Equity | Quilter PLC | 20180625 |
| INP SJ Equity | Investec PLC | 20020722 |

Continued on next page

**Table 4 – continued from previous page**

| Ticker | Name | Listing Date |
|---|---|---|
| *Health Care (4 members)* | | |
| AIP SJ Equity | Adcock Ingram Holdings Ltd | 20080825 |
| APN SJ Equity | Aspen Pharmacare Holdings Ltd | 19880101 |
| NTC SJ Equity | Netcare Ltd | 19960701 |
| LHC SJ Equity | Life Healthcare Group Holdings Ltd | 20100610 |
| *Industrials (10 members)* | | |
| BVT SJ Equity | Bidvest Group Ltd/The | 19540701 |
| GND SJ Equity | Grindrod Ltd | 19860101 |
| HDC SJ Equity | Hudaco Industries Ltd | 19850101 |
| TCP SJ Equity | Transaction Capital Ltd | 20120607 |
| RBX SJ Equity | Raubex Group Ltd | 20070320 |
| HCI SJ Equity | Hosken Consolidated Investments Ltd | 19730701 |
| WBO SJ Equity | Wilson Bayly Holmes-Ovcon Ltd | 19880101 |
| RLO SJ Equity | Reunert Ltd | 19480101 |
| BAW SJ Equity | Barloworld Ltd | 19410101 |
| KAP SJ Equity | KAP Ltd | 19940701 |
| *Information Technology (4 members)* | | |
| BYI SJ Equity | Bytes Technology Group PLC | 20201217 |
| KRO SJ Equity | Karooooo Ltd | 20210421 |
| AEL SJ Equity | Altron Ltd | 19700101 |
| DTC SJ Equity | DataTec Ltd | 19940101 |
| *Materials (23 members)* | | |
| S32 SJ Equity | South32 Ltd | 20150518 |

Continued on next page

**Table 4 – continued from previous page**

| Ticker | Name | Listing Date |
|---|---|---|
| OMN SJ Equity | Omnia Holdings Ltd | 19800101 |
| GLN SJ Equity | Glencore PLC | 20131113 |
| AFT SJ Equity | Afrimat Ltd | 20061107 |
| BHG SJ Equity | BHP Group Ltd | 20220131 |
| THA SJ Equity | Tharisa PLC | 20140410 |
| HAR SJ Equity | Harmony Gold Mining Co Ltd | 19510701 |
| IMP SJ Equity | Impala Platinum Holdings Ltd | 19730701 |
| NPH SJ Equity | Northam Platinum Holdings Ltd | 20210915 |
| AFE SJ Equity | AECI Ltd | 19660101 |
| ARI SJ Equity | African Rainbow Minerals Ltd | 20071107 |
| AMS SJ Equity | Anglo American Platinum Ltd | 19950101 |
| SAP SJ Equity | Sappi Ltd | 19370101 |
| SOL SJ Equity | Sasol Ltd | 19791001 |
| ANG SJ Equity | Anglogold Ashanti Plc | 20230920 |
| PPC SJ Equity | PPC Ltd | 20010101 |
| MNP SJ Equity | Mondi PLC | 20070702 |
| SSW SJ Equity | Sibanye Stillwater Ltd | 20200219 |
| DRD SJ Equity | DRDGOLD Ltd | 19850101 |
| AGL SJ Equity | Anglo American PLC | 19990524 |
| KIO SJ Equity | Kumba Iron Ore Ltd | 20061120 |
| PAN SJ Equity | Pan African Resources PLC | 20070731 |
| GFI SJ Equity | Gold Fields Ltd | 19680101 |
| *Real Estate (19 members)* | | |

Continued on next page

**Table 4 – continued from previous page**

| **Ticker** | **Name** | **Listing Date** |
|---|---|---|
| SRE SJ Equity | Sirius Real Estate Ltd | 20141205 |
| BTN SJ Equity | Burstone Group Limited | 20110414 |
| GRT SJ Equity | Growthpoint Properties Ltd | 19870701 |
| HYP SJ Equity | Hyprop Investments Ltd | 19880701 |
| NRP SJ Equity | NEPI Rockcastle NV | 20170712 |
| RDF SJ Equity | Redefine Properties Ltd | 20000223 |
| SSS SJ Equity | Stor-Age Property REIT Ltd | 20151116 |
| HMN SJ Equity | Hammerson PLC | 20160901 |
| MSP SJ Equity | MAS P.L.C. | 20090831 |
| FFB SJ Equity | Fortress Real Estate Investments Ltd | 20091022 |
| FTB SJ Equity | Fairvest Ltd | 19700101 |
| ATT SJ Equity | Attacq Ltd | 20131014 |
| LTE SJ Equity | Lighthouse Properties plc | 20151127 |
| RES SJ Equity | Resilient REIT Ltd | 20021206 |
| SHC SJ Equity | Shaftesbury Capital PLC | 20100510 |
| EMI SJ Equity | Emira Property Fund Ltd | 20030701 |
| SAC SJ Equity | SA Corporate Real Estate Ltd | 19950101 |
| EQU SJ Equity | Equites Property Fund Ltd | 20140618 |
| VKE SJ Equity | Vukile Property Fund Ltd | 20040623 |
| *Utilities (1 member)* | | |
| MKR SJ Equity | Montauk Renewables Inc | 20210125 |

# B The Riemannian Manifold Geometry Toolkit

The deep learning layer architecture is underpinned by some important geometric computations on the SPD manifold $\mathcal{S}_*^+$. These results extend to the batch-normalisation (batch normalisation) algorithm and the manifold-constrained backpropagation used in the SPD matrix learning framework (Huang and Gool, 2016; Brooks et al., 2019b; Wang et al., 2022b)

## B.1 Basic Concepts

We begin by introducing the covariance matrices of interest as a subset of the vector space of $n \times n$ symmetric square matrices. In particular, these $n \times n$ SPD covariance matrices form an open convex cone of dimension $\frac{n(n+1)}{2}$.

This tensor space of SPD matrices may be endowed with an affine-invariant Riemannian metric which gives rise to the useful theoretical result that a regular and complete boundaryless manifold may replace the cone of SPD matrices, specifically the open sub-manifold becomes a smooth Riemannian manifold, denoted $\mathcal{S}_*^+$, when endowed with the affine-invariant Fisher-Rao metric (Said et al., 2017). This metric for a point $P \in \mathcal{S}_*^+$ is given by

$$g_P(\xi_P, \eta_P) = \operatorname{tr}\big(\xi_P P^{-1} \eta_P P^{-1}\big). \tag{63}$$

The Riemannian, or Fisher-Rao, metric provides an important framework for the generalisation of statistics to manifolds (Pennec et al., 2006).

## B.2 Riemannian Distances and Tangent Space Mappings

The Affine-Invariant Riemannian Metric (AIRM), or distance, $\delta_{\mathfrak{R}}(P_1, P_2)$ on a manifold is the shortest parameterised curve $\xi_P(t)$, or length of the geodesic $\gamma_{P_1 \to P_2}$, between two points $P_1$ and $P_2$. This natural distance is computed following the Fisher-Rao distance measure. This geodesic distance is calculated as:

$$\delta_{\mathfrak{R}}(P_1, P_2) = \frac{1}{2}\left\|\log\left(P_1^{-\frac{1}{2}} P_2 P_1^{-\frac{1}{2}}\right)\right\|_F = \left(\sum_{i=1}^{n} \log^2 \lambda_i\right)^{\frac{1}{2}}, \tag{64}$$

with $\|\cdot\|_F$ the Frobenius norm and $\lambda_1, \ldots, \lambda_n$ the eigenvalues of $P_1^{-\frac{1}{2}} P_2 P_1^{-\frac{1}{2}}$, following Yuan et al. (2020) and Zanini et al. (2018).

The invariance property of this distance is particularly useful since it means that the distance between two SPD matrices is invariant with respect to a change in the reference point, such as $P_0$ (Zanini et al., 2018). In the space of multivariate normal distributions, where this entropy-related metric allows for statistical inference and optimisation in the context of the particular geometry of the covariance matrix manifold, this proves useful in the formulation of batch normalisation computations (Pinele et al., 2019; Atkinson and Mitchell, 1981). This invariance further allows for diffusion backpropagation learning schemes to be adapted to the Riemannian framework as will be expanded on below.

This metric is, however, computationally intensive. Therefore, Wang et al. (2022b) introduce the LEM, which, when compared with the above Affine-Invariant Riemannian Metric (AIRM), works directly on the SPD matrix logarithmic domain and is defined as:

$$\delta_{LEM}(P_1, P_2) = \|\log(P_1) - \log(P_2)\|_F^2 \,. \tag{65}$$

Huang and Gool (2016) do indeed make use of the LEM, specifically in learning of LogEig layer in the SPDNet by calculation of the Log-Euclidean Riemannian geometric mean with respect to this metric (see Equation 4 above).

Mapping between the tangent Euclidean spaces as a bundle of groupings at each point on the manifold and the manifold itself also crucially depends on the information geometry implied by the Riemannian metric (Arsigny et al., 2007). This is achieved by the logarithmic mapping $Log_{P_0}$ of any point $P \in \mathcal{S}_*^+$ from the position $P_0 \in \mathcal{S}_*^+$ to a corresponding vector $S$ in the tangent space $\mathcal{T}_{P_0}$, or domain of $P_0$. This approach takes information integration of causal dynamical systems into account (Amari et al., 2018). The inverse mapping is denoted as $\mathrm{Exp}_{P_0}$ where both may be computed in closed form as:

$$\mathrm{Exp}_{P_0}(S) = P_0^{\frac{1}{2}} \exp\big(P_0^{-\frac{1}{2}} S P_0^{-\frac{1}{2}}\big) P_0^{\frac{1}{2}} \in \mathcal{S}_*^+, \quad \forall S \in \mathcal{T}_{P_0}, \tag{66}$$

and

$$\mathrm{Log}_{P_0}(P) = P_0^{\frac{1}{2}} \log\big(P_0^{-\frac{1}{2}} P P_0^{-\frac{1}{2}}\big) P_0^{\frac{1}{2}} \in \mathcal{T}_{P_0}, \quad \forall P \in \mathcal{S}_*^+. \tag{67}$$

## B.3 The Riemannian Barycenter

The Riemannian barycenter, also known as the $L^2$-Riemannian mean, Fréchet mean or Riemannian center of mass is defined as the the point on the manifold that minimises inertia with respect to the Riemannian metric (Marti, 2021).

When computing the batch mean for batch normalisation, it may be tempting to average over the batch $\mathcal{B}$ of $N$ SPD matrices $\{P_i\}_{i \leq N}$ using the arithmetic mean. This should be avoided, however, due to the induced *swelling effect* which causes an increase in the matrix determinant post-averaging (Yuan et al., 2020). Additionally, the arithmetic mean does not enforce invariance under inversion which leads to ill-conditioned matrix inputs in a multitude of optimal control optimisation problems (Marti, 2021).

Thus, the Riemannian barycenter $\mathfrak{G}$ is the favoured geometric mean. It satisfies all fundamental properties that characterise a geometrically-appropriate matrix mean, referred to as the ALM list (Ando et al., 2004). However, it is well-established that the barycenter is not a robust estimator and is sensitive to outliers. In order to circumvent these flaws Fletcher et al. (2009) construct the weighted geometric median of a set of discrete sample points lying on a Riemannian manifold and prove its existence and uniqueness. Thus the barycenter may be trivially extended to account for the weight space $\mathbf{w} := \{w_i\}_{i \geq N}$ whilst maintaining the strict convexity of the inertia defined in terms of the Riemannian metric 63, thus ensuring that the barycenter, or median, is a unique minimiser (Brooks et al., 2019a; Bhatia and Holbrook, 2006). It is defined as

$$\mathfrak{G} = \underset{G \in \mathcal{S}_*^+}{\mathrm{argmin}} \sum_{i=1}^{N} w_i \delta_{\mathfrak{R}}^2(G, P_i), \quad \text{with} \quad w_i \geq 0 \quad \text{and} \quad \sum_{i \leq N} w_i = 1, \tag{68}$$

where $G$ is a bias parameter matrix, and $\delta_{\mathfrak{R}}$ is the geodesic distance associated with the metric in Equation 63. The Riemannian barycenter 68 has no closed-form solution for $N > 2$, meaning that the mean is generally computed using iterative methods as employed by the *Karcher flow algorithm*, or the Riemannian optimisation framework. This technique, proposed in the work of Yang et al. (2010), relies on the mappings between the Riemannian manifold $\mathcal{S}_*^+$ and the Euclidean tangent space $\mathcal{T}_{P_0}$ as described above in equations 66 and 67. An interim step of this iterative mapping process is that of the geometric averaging which takes place in the tangent space. Commonly, $\mathfrak{G}$ is initialized by the arithmetic-harmonic mean (Yuan et al., 2020). Following the procedure of Brooks et al. (2019a) which relies on the computation of a unique Riemannian median, the process guarantees convergence (Yang, 2010).

**Computation of the weighted Riemannian barycenter:** Let $(t_k)_k$ be a step sequence in the interval $(0, \epsilon]$ for small $\epsilon$ where

$$\lim_{k\to\infty} t_k = 0 \quad \text{and} \quad \sum_{k=1}^{\infty} t_k = +\infty. \tag{69}$$

For each $P_i \in \mathcal{B}$ we define

$$H(G) = \sum_{\substack{i=1 \\ P_i \neq G}}^{N} w_i \frac{-\exp_G^{-1} P_i}{\delta_{\mathfrak{R}}(G, P_i)}, \tag{70}$$

where $G \in \mathcal{S}_*^+$. Then the sequence of $(G_k)_K$ generated by the algorithm 3 will converge on $\mathfrak{G}$ of the batch $\mathcal{B}$ of SPD matrices $\{P_i\}_{i \leq N}$.

**Algorithm 3** Riemannian median: Karcher flow algorithm

```
Require: $G_1 \in \mathcal{S}_*^+$                                   ▷ initialise $\mathfrak{G}$
  while $H(P_k) \neq 0$ do
      $G_{k+1} = \exp_{G_k}\left(-t_k \frac{H(G_k)}{\|H(G_k)\|}\right)$
  end while
```

## B.4 Extending Batch-Normalisation to the SPD Manifold

In order to expand on previous SPD matrix learning architectures Brooks et al. (2019a) propose that parallel transport be leveraged such that a batch normalisation layer may be introduced. Using parallel transport means that the Euclidean batch normalisation, which involves the centering and biasing of batch $\mathcal{B}$, may be generalised to the curved geometry of the manifold by involving the identity matrix $I_d$ and parametric bias SPD matrix $G$. Analagous operations on the manifold require manipulations of the SPD matrices to either shift the sample Riemannian barycenter $\mathfrak{G}$ around $I_d$, and towards a bias parameter $G$ as featured in Equation 63.

Parallel transport on $\mathcal{S}_*^+$ for *tangent vectors* is defined as

$$\Gamma_{p_1 \to P_2}(S) = (P_2 P_1^{-1})^{\frac{1}{2}} S (P_2 P_1^{-1})^{\frac{1}{2}} \in \mathcal{T}_{P_2}, \quad \forall S \in \mathcal{T}_{P_1}. \tag{71}$$

The parallel transport operator acts on the vector $S$ which exists in the tangent plane $\mathcal{T}_{P_1}$ and defines a path between a corresponding point $P_1 \in \mathcal{S}_*^+$ and another point $P_2 \in \mathcal{S}_*^+$. $S$, as it moves along this particular path, remains parallel to itself in the associated tangent planes.

**SPD transport**: $\Gamma_{P_1 \to P_2}$ draws on this technique in order to transport points on the cone manifold of SPD matrices, as opposed to vectors in the tangent space only (Brooks et al., 2019a). This is executed by first projecting the point from the manifold to the tangent space using the *Log* mapping as in Equation 67, applying parallel transport to the resulting vector using Equation 71, and finally, *Exp*-mapping this back to the manifold by employing Equation 66 (Yair et al., 2019). Interestingly, the operation of SPD transport matches exactly the above formula 71 (Yair et al., 2019). The centering and biasing features of normalisation in the context of Riemannian geometry are defined as the SPD transport from $\mathfrak{G}$ to $I_d$ and subsequently from $I_d$ to $G$ respectively.

**Batch centering and biasing on the manifold:** Centering $\mathfrak{G}$ toward $I_d$:

$$\bar{P}_i = \Gamma_{\mathfrak{G} \to I_d}(P_i) = \mathfrak{G}^{-\frac{1}{2}} P_i \mathfrak{G}^{-\frac{1}{2}} \tag{72}$$

Biasing from $I_d$ towards parametric SPD matrix $G$:

$$\tilde{P}_i = \Gamma_{I_d \to G}(\bar{P}_i) = G^{\frac{1}{2}} \bar{P}_i G^{\frac{1}{2}}, \tag{73}$$

$\forall i \leq N$ and $P_i$ in the batch $\mathcal{B}$ of $N$ SPD matrices $\{P_i\}_{i \leq N}$.

## B.5 Gaussian Distribution on the Riemannian Manifold

In order to generalise the concept of Euclidean batch-normalisation to SPD matrices, the concept of Gaussian density must first be established in the context of the Riemannian manifold. In traditional neural nets, the centering and standardisation of batches of data requires parameterised mean and variance values as calibrated from the learnt Gaussian distribution (Ioffe and Szegedy, 2015). The concepts of a Riemannian mean and variance are currently ill-defined. Viewing the set of covariance matrices as a cone of SPD matrices allows for their information geometry to inform a maximum-entropy approach to deriving the Gaussian density from exponential families (Barbaresco, 2019). Jean-Louis Koszul introduced fundamental tools to characterise the geometry of sharp convex cones, where the associated exponential family is entirely parametrised by, simply, the Riemannian mean as defined above. The notion of Gaussian density on $\mathcal{S}_*^+$ has no associated concept of variance as is the case for its Euclidean counterpart. The density $p$ on the set of SPD matrices $M$ of dimension $n$ may be defined as

$$p(M) \propto \det\left(\alpha \mathfrak{G}^{-1}\right) e^{\operatorname{tr}(\alpha \mathfrak{G}^{-1} M)}, \tag{74}$$

where

$$\alpha = \frac{n+1}{2}. \tag{75}$$

In this equation $\mathfrak{G}$ is the Riemannian mean.

By adopting this specific density definition, the batch-normalisation layer accounting for the Riemannian geometry of the manifold of SPD covariance matrices will involve centering and biasing, while excluding variance multiplication since its notion does not exist in the non-Euclidean space (Brooks et al., 2019a).

# C Notation Discrepancy Notes

While the work of Brooks et al. (2019a), Wang et al. (2022b) and Wang et al. (2022a) build on that of Huang and Gool (2016), there are some notation discrepancies worth clarifying. Their similarities and differences are highlighted in Table 5.

| | **Papers** | | |
|---|---|---|---|
| | Huang and Gool (2016) | Brooks et al. (2019b) | Wang et al. (2022b) |
| **Description** | **Symbol** | | |
| Layer index | $k$ | $l$ | $k,\ \alpha$ |
| Weights | $W$ | $w$ | $W$ |
| SPD manifold | $Sym^+_{d_k}$ | $\mathcal{S}^+_*$ | $\mathcal{S}^{d_k}_{++}$ |
| Tangent projection/retraction operation | $\Gamma$ | $\Pi$ | – |
| Input matrix | $X_k$ | $P, X^{(l+1)}$ | $M$ |
| Output hidden layer matrix | $X_{k-1}$ | $X,\ X^{(l)}$ | $M$ |
| Monotonous non-linear function operating on eigenvalues $\sigma$ | $P$ | $f$ | $f$ |
| Loss function | $L$ | $\mathcal{L}$ | $L$ |
| Loewner matrix | - | $L$ | – |
| Eigenspace | $U$ | $U$ | $U$ |
| Eigenvalues | $\sigma_i, \sigma_j$ | $(\sigma_i)_{i \leq n}$ | |
| Eigenvectors | $\Sigma$ | $\Sigma$ | $\Sigma$ |
| Rectification threshold | $\epsilon$ | $\epsilon$ | $\epsilon$ |
| Euclidean gradient w.r.t. weights matrix **W** or bias parameter $G$ | $\nabla \mathcal{L}^{(k)}_{\mathbf{W}_k}$ | $\frac{\partial \mathcal{L}}{\partial G},\ \delta G_{eucl}$ | – |
| Tangential gradient | $\tilde{\nabla} \mathcal{L}^{(k)}_{\mathbf{W}_k}$ | $\partial G_{riem}$ | – |

| | | | |
|---|---|---|---|
| Manifold | - | $\mathcal{M}$ | $\mathbf{M}$ |
| Bias parameter | - | $G$ | – |
| Riemannian barycenter | - | $\mathfrak{G}$ | $\mathcal{P}$ |
| Riemannian | $\mathbb{R}^{d_k \times d_k}$ | $\mathfrak{R}$ | – |
| BiMap, LogEig, ReEig and ExpEig | $f_b, f_l, f_r$ | - | $f_b, f_l, f_r, f_e$ |
| Max function | $A$ | max | max |
| Compact Stiefel manifold linking layers | $St(d_k, d_{k-1})$ | - | $St(d_k, d_{k-1})$ |
| Tangent space | - | $\mathcal{T}$ | – |
| Partial w.r.t. the LEM | $\delta_{LEM}$ | $\delta_{LEM}$ | $D_{LEM}$ |
| Virtual layer | $k'$ | – | $k^{\triangle}$ |

**Table 5:** Notation variations across the work of Brooks et al. (2019a), Wang et al. (2022b) and Wang et al. (2022a) as built on Huang and Gool (2016).

# D Research Reproducibility

All code written in support of this research may be found at the Orton and Gebbie (2024) GitHub repository BH-SPD-CorrMat-Nets, or on the Deep Learning Block Hierarchical SPD Correlation Matrices for Regime Detection OSF project page `https://osf.io/msrgc/` where data for both SPDNet, SPDNetBH and U-SPDNet model evaluation on both real world and synthetic data is stored.

The implementation of the SPDNet is an extension of the code base produced by Brooks et al. (2019b) using Python, which is replicated and augmented with the RBN approach as found on GitLab (link below). The original U-SPDNet architecture is found on GitHub, written by Wang et al. (2022b) in MATLAB, which has been modified for training on the SPD correlation matrices data sets. Finally, the Tumminello et al. (2007)-based technique leveraged by Yelibi (2021) is extended for synthetic nested block hierarchical correlation matrix generation as found at Orton and Gebbie (2024).

**Repositories requiring download:**

1. `https://gitlab.lip6.fr/schwander/torchspdnet`
2. `https://github.com/GitWR/U-SPDNet`
3. `https://github.com/AlexaOrton/BH-SPD-CorrMat-Nets`

The language used in constructing the SPDNet models, performing exploratory data analysis and synthetic data generation is Python. Specifically, Jupyter notebooks are used in running of of the .ipynb scripts. We recommend installing Anaconda from which Jupyter notebooks may be launched once the GitHub repositories has been downloaded to your local machine. We also recommend creating a virtual project environment.

**Notes for U-SPDNet MATLAB Implementation:**

1. In order to run the U-SPDNet .m scrips, MATLAB version R2024a is required.
2. The matrix data used in training and validating the SPDNet and SPDNetBN models must be converted from `.py` to `.mat` format for U-SPDNet evaluation, and create the files: `sample_for_SPDNet_UAV.mat`. Then run the following jupyter notebooks:
    (a) Synthetic data: `U-SPDNet_synth_data_processing.ipynb`;
    (b) Real-world data: `U-SPDNet_JSETop60_data_processing.ipynb`; and
    (c) Portfolio Backtesting: `backtest_data_conversion.ipynb`.

   These jupyter notebooks complete the actions for the synthetic, real-world and portfolio backtesting scenarios respectively.
3. One would need to transfer the data files found in the Orton and Gebbie (2024) to the `afew` folder within the data folder contained in the Wang et al. (2022b) repository directories. Note that they need to possess the file path structure of `/data/afew/UAV_CONV` in order to be recognised by the `spdnet_afew.m` matlab script.
4. The `spdnet_train_afew.m` file must be replaced with the `spdnet_train_afew_orig.m` in order to save latent space matrices and plot their heatmaps using the viridis colour palette when runnning `spdnet_afew.m`.

Table 6 below links each of the figures in this dissertation as well as data processing and generation, and model evaluation to scripts found in the GitHub repository found at: `https://github.com/AlexaOrton/BH-SPD-CorrMat-Nets`.

| **Script File Name** | **Figures** |
|---|---|
| `Top60_data_processing_dist.ipynb` | 4, 6, 5, 10, 11 |
| `JSE_Top60_Kmeans.ipynb` | 7 |
| `synthetic_data_processing_yelibi_gebbie.ipynb` | 12, 13 |
| `Top60_SPDNet_dist.ipynb` | 14, 15 |
| `Top60_SPDNet_3BiRe_dist.ipynb` | 16, 17 |
| `spdnet_train_afew_orig.m` | 18, 19, 20, 26 |
| `block_sampling_JSE_Top_60_SPDNet.ipynb` | ??, 21, 22, 23 |
| `Top60_U-SPDNet_backtesting.ipynb` | 25, 27, 28, 29, 30, 31 |

**Table 6:** Overview of figures, scripts, and their functions as found at the Orton and Gebbie (2024) GitHub repository: `https://github.com/AlexaOrton/BH-SPD-CorrMat-Nets`

Alternatively, a full description of this project along with supporting data and code scripts may be found at the OSF project page: `https://osf.io/msrgc/`.