

Deep Risk Model: A Deep Learning Solution for Mining Latent Risk Factors to Improve Covariance Matrix Estimation

Hengxu Lin^{*†}
Columbia Business School
New York, United States
helin23@gsb.columbia.edu

Weiying Liu
Microsoft Research
Beijing, China
Weiying.Liu@microsoft.com

Dong Zhou^{*}
Microsoft Research
Beijing, China
Zhou.Dong@microsoft.com

Jiang Bian
Microsoft Research
Beijing, China
Jiang.Bian@microsoft.com

ABSTRACT

Modeling and managing portfolio risk is perhaps the most important step to achieve growing and preserving investment performance. Within the modern portfolio construction framework that built on Markowitz’s theory, the covariance matrix of stock returns is required to model the portfolio risk. Traditional approaches to estimate the covariance matrix are based on human designed risk factors, which often requires tremendous time and effort to design better risk factors to improve the covariance estimation. In this work, we formulate the quest of mining risk factors as a learning problem and propose a deep learning solution to effectively “design” risk factors with neural networks. The learning objective is carefully set to ensure the learned risk factors are effective in explaining stock returns as well as have desired orthogonality and stability. Our experiments on the stock market data demonstrate the effectiveness of the proposed method: our method can obtain 1.9% higher explained variance measured by R^2 and also reduce the risk of a global minimum variance portfolio. Incremental analysis further supports our design of both the architecture and the learning objective.

KEYWORDS

risk management, covariance estimation, neural network, machine learning

1 INTRODUCTION

Modern portfolio theory has witnessed increasing significance of risk management in stock investment in past decades. Successful risk management recognizes the equity market risk that the stocks have exposure. Thereby enables investors to adjust their strategies according to estimation of return and arising uncertainty while constructing their portfolio. The most widely adopted framework to select the best stock portfolio is built on Markowitz’s mean-variance theory [25]. Within this framework, the optimal portfolio is obtained by solving a constrained utility-maximization problem, with the utility defined as the expected portfolio return minus a cost of portfolio risk. To precisely determine the risk of any portfolio, having good estimation of the covariance matrix of stock returns is the fundamental question.

Nevertheless, the empirical covariance matrix can be extremely ill-conditioned as the dimension of stocks grows and even becomes rank deficient such that cannot be used for portfolio optimization. To facilitate the estimation of the covariance matrix under such high-dimension setting, [22, 23] proposed to shrink the off-diagonal elements in the empirical covariance matrix towards zero, based on the assumption that the target covariance matrix is sparse. However, stock returns are often highly related to each other and thus the covariance matrix of stock returns cannot sparse. A more realistic assumption is that the high-dimensional stock returns are driven by a set of low-dimensional *risk factors* [31]. For example, it is often observed that stocks with similar market capitalization tend to have similar market behaviour, thus the market capitalization can be such a risk factor, which is also widely known as size [13]. By introducing multiple risk factors, the estimation of a high-dimensional covariance matrix of stock returns can be decomposed to the estimation of a low-dimensional covariance matrix of risk factor returns¹. Apparently, good estimation of the covariance matrix requires good design of the risk factors.

Risk factors have been carefully studied by investment professionals. Fundamental risk factors are proposed first since they are highly interpretable on an ongoing basis. Capital assets pricing model (CAPM) [30] first formulated systematic risk factor as “beta”. Multiple factors are further included into this fundamental risk factor family: size and value [13], momentum [8] and non-linear size (BARRA) [31], to name a few. Despite their success, the design of these risk factors was extremely slow: it took almost half a century from CAPM to BARRA, with only roughly ten new factors added in the evolution. This is because the discovery of these risk factors usually rely on tremendous human efforts to verify hypotheses based on the long-term observation. Statistical risk factors [1, 3] are thereby proposed to address such limitation, which applies matrix decomposition algorithms like principal component analysis (PCA) or Factor Analysis [19] to obtain the latent risk factors. However, such decomposition methods are linear, thus hard to capture idiosyncratic components and discover non-linear patterns. Besides, unlike fundamental risk factors that have been tested in explaining forward returns, statistical risk factors are simply the principle components of history returns and often struggle in out-of-sample generalization. Therefore, a more efficient and effective risk factor

^{*}The first two authors have equal contribution.

[†]This work was done when the first author was an intern at Microsoft Research Asia.

¹Factor returns are the cross-sectional regression coefficients. We will use these two terms interchangeably in the rest of the paper.

mining solution is still desired for improving the covariance matrix estimation of stock returns.

In this work, we propose a deep learning solution, *Deep Risk Model*, to facilitate the design of risk factors. From the perspective of machine learning, mining risk factors could be effectively solved given a well designed architecture to fully utilize the available information and an appropriate learning objective to match the desired properties of the risk factors. First of all, the architecture should not only be compatible with the transformations used in human designed risk factors, but also have the capacity to represent more complex transformation. To this end, we design a hybrid neural network with Gated Recurrent Units (GRU) [10] to model the temporal information (e.g., the standard deviation of historical stock returns), and a Graph Attention Networks (GAT) [37] layer to represent the cross-sectional transformation (e.g., the excess return over the sector average). Furthermore, we design a learning objective to pursue the desired properties of the learned risk factors on three dimensions: the ability of explaining the variance of stock returns, orthogonality and stability.

To demonstrate the effectiveness of proposed method, we further conduct extensive experiments on the stock market data. Specifically, we first compare the variance explanation ability of risk factors from the proposed method with the ones from the fundamental risk model and statistical risk model. Experiment results show our method can consistently outperform the other two baselines with 1.9% higher R^2 . We further use the learned risk factors to estimate the covariance with which we construct a Global Minimum Variance (GMV) portfolio. We find the proposed deep risk model's factors are more reliable and achieve lowest volatility for the GMV portfolio. Furthermore, we analyze characteristics of the learned factors and receive excellent accomplishment of controlling multi-collinearity and stability. Ablation studies on our model also manifest the necessity and superiority of our architecture and learning objective design.

The main contributions are summarized as follows:

- To the best of our knowledge, we are the first to formulate the quest of risk factor mining as a supervised learning task.
- We propose *Deep Risk Model* as an effective and adaptive solution for learning risk factors with a comprehensive design of both the architecture and the learning objective.
- We conduct extensive experiments with real-world stock market data and compare with existing state-of-the-art baselines. Experiment results demonstrate the effectiveness of the proposed method.

2 RELATED WORKS

Covariance Matrix Estimation. Large scale covariance matrix estimation is fundamental in multivariate analysis and ubiquitous in financial economics panel data. The main challenge of this field remains on the singular matrix arising from an insufficient sample size compared to the large sample dimension. Related solutions mainly focus on two aspects: rank-based estimation and factor-based estimation. Rank-based estimation relies on the sparsity assumption of the matrix: a majority of the matrix elements is nearly zero, thus various thresholds [6, 21, 29] could be designed to control these elements and the parameters to be estimated are reduced significantly.

Factor-based estimation is widely adopted in the scenario where the sparsity assumption is not applicable when some variables are highly correlated (e.g. stocks). By introducing factors and the factor covariance matrix to the decomposition, the residual covariance could be expressed as a conditional sparse matrix: conditional to common factors, the covariance matrix of remaining components are sparse. The factor model is more reasonable and commonly used in economics forecasting for its capacity to perform well in out-of-sample test [15, 16, 18].

Matrix decomposition. Matrix Decomposition or Matrix Factorization, which is fundamental in linear algebra [14, 33], factorizing the matrix into a product of matrices. Matrix decomposition algorithms are designed for specific purpose. LU decomposition, QR decomposition, Cholesky decomposition, rank factorization are used to solve systems of linear equations. Eigendecomposition, or spectral decomposition is used to attain eigenvectors and eigenvalues. Jordan decomposition, Schur decomposition, scale invariant decomposition [34], singular value decomposition (SVD) [35] are further implementations derived from eigendecomposition. In addition, principal component analysis (PCA) [38] is an application of SVD.

Representation Learning. Representation Learning, or Feature Learning, is a technique used to extract effective representations needed for specific downstream task (e.g. classification, regression, feature detection) from raw feature. Unsupervised and supervised representation learning has been well studied for decades: matrix factorization as discussed above, unsupervised dictionary learning, principal components analysis, independent components analysis [36], clustering techniques like K-Means, linear discriminant analysis [4]. Recently, the success of deep learning brings multiple supervised representation learning techniques into prosperity. Neural networks have demonstrated their distinguish competency and accomplished state-of-the-art in representation learning even for various kinds of tasks: vision computing [2, 9], natural language processing [11, 27, 32].

3 PRELIMINARIES

The main interest of this section is to give an overview of modern multi-factor model theories in *factor-based risk models*, and to manifest the necessity of adopting an advanced solution to discover and capture intrinsic and adaptive risk factors.

3.1 Factor-Based Risk Models

Driven by the assumption that the intrinsic dependency of stock returns could be explained by some common latent factors, a large volume of academic and industrial research discuss and study the design of factors. Mathematically, the multi-factor model is a multi-variable linear model with risk factors as the independent variables and stock returns as the dependent variable. Let $y_{i,t}$ denote the observable return of stock i at time t , the multi-factor model takes the form of

$$y_{i,t} = \mathbf{f}_i^T \mathbf{b}_t + u_{i,t} \quad (1)$$

where $\mathbf{f}_i \in \mathbb{R}^K$ represents a vector of K risk factors for the i -th stock, $\mathbf{u}_{i,t}$ is the idiosyncratic error term, and $\mathbf{b}_t \in \mathbb{R}^K$ is the vector of regression coefficients, which is also called factor returns.

Moreover, considering forward returns of a specific stock over some horizon H , we can further decompose the stock returns into factor returns by

$$\begin{bmatrix} y_{i,t} \\ y_{i,t+1} \\ \dots \\ y_{i,t+H-1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_i^\top \mathbf{b}_t + \mathbf{u}_{i,t} \\ \mathbf{f}_i^\top \mathbf{b}_{t+1} + \mathbf{u}_{i,t+1} \\ \dots \\ \mathbf{f}_i^\top \mathbf{b}_{t+H-1} + \mathbf{u}_{i,t+H-1} \end{bmatrix}, \quad (2)$$

or equivalently in its matrix form

$$\mathbf{y}_i = \mathbf{B} \mathbf{f}_i + \mathbf{u}_i, \quad (3)$$

where $\mathbf{B} \in \mathbb{R}^{H \times K}$ contains the factor returns from time t to $t+H-1$. Without loss of generality, we assume that $\mathbf{E}(\mathbf{y}_i) = \mathbf{E}(\mathbf{u}_i) = \mathbf{0}$, hereby, the covariance $\Sigma_{i,j}$ of any two stocks i, j can be derived as:

$$\Sigma_{i,j} = \mathbf{f}_i^\top \Sigma_{\mathbf{B}} \mathbf{f}_j + \sigma_i \rho_{ij} \sigma_j \quad (4)$$

where $\sigma_i = \text{std}(\mathbf{u}_i)$ is standard deviation of the idiosyncratic errors, and ρ_{ij} denotes their correlation. A common assumption is that the idiosyncratic errors are independent (otherwise they can be explained by certain risk factors) and thus we have $\rho_{ij} = \mathbb{1}(i = j)$. Additionally, let $\mathbf{F} := [\mathbf{f}_1, \dots, \mathbf{f}_N]^\top \in \mathbb{R}^{N \times K}$ denote the risk factor matrix for all N stocks, the entire covariance matrix of stock returns can be derived as

$$\Sigma = \mathbf{F} \Sigma_{\mathbf{B}} \mathbf{F}^\top + \Delta, \quad (5)$$

where Δ is a diagonal matrix with $\Delta_{i,i} = \sigma_i^2$. Thereby we can decompose the covariance of stock returns $\Sigma \in \mathbb{R}^{N \times N}$ into the covariance of factor returns² $\Sigma_{\mathbf{B}} \in \mathbb{R}^{K \times K}$ which has lower dimension. Apparently, the design of risk factors \mathbf{F} is crucial to give accurate covariance estimation as both \mathbf{b} and \mathbf{u} are determined when \mathbf{F} is specified in Equation 1.

3.2 Risk Factor Design

The most popular risk factors are designed by human experts [31], which have been grounded in academic research and historically recognized to explanations of stock return: Size (measures a company's scale of market capital), Value (measures a stock's intrinsic value), Beta (explanation to market index return), Momentum (uses price trend to forecast future return), to name a few. Despite the achievement from the human designed risk factors, it actually took a long time for human specialists to draw these factors based on the long-term performance. Besides, evidence also exemplifies that these factors during unpredictable market movements. Statistical risk factors are thereby proposed to address such limitations of human designed factors, which usually applies principal component analysis (PCA) on historical stock returns to obtain the first K components as the risk factors. Although efficient, such a method often has poor out-of-sample performance in risk forecasting as the obtained risk factors simply overfit the history returns but not explaining the forward-looking returns. The limitations of both these two methods stimulate us to design a more efficient and effective solution for the design of risk factors.

²As we cannot access all forward returns like $y_{i,t+1}$ at time t , a common practice is using historical stock returns to estimate $\Sigma_{\mathbf{B}}$.

4 DEEP RISK MODEL

From the perspective of machine learning, the design of risk factors can be seen as the learning of a transformation function g_θ , which maps raw data \mathbf{X} to risk factors $\mathbf{F} = g_\theta(\mathbf{X})$. From this perspective, human designed factors are equivalent to specifying the transformation g_θ as explicit formulas. In this work, we consider adopting a deep neural network to learn such transformation in a data-driven approach. Apparently, a well-designed neural network architecture should not only be compatible with human designed transformations but also have the capacity to capture more complex transformations. Therefore, we design a hybrid neural network to represent both temporal transformation and cross-section transformation, besides the neural network can also ensure capture the complex non-linear transformations. Moreover, we also prepare our learning objective carefully to ensure the ability of learned risk factors to explain variance of stock returns, while satisfying the requirements of both orthogonality and stability.

4.1 Architecture Design

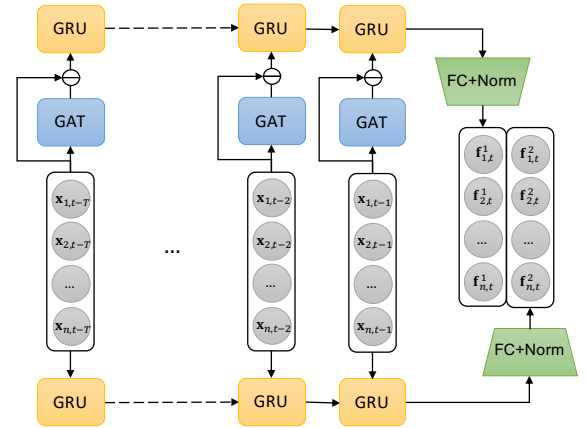


Figure 1: The proposed architecture for risk factors mining consists of two branches of Gated Recurrent Units (GRU) modules to learn temporal transformations. One of the two branches has a Graph Attention Network (GAT) layer to learn cross-sectional transformation.

4.1.1 Temporal Transformation. In order to learn the temporal transformation, e.g., the standard deviation of historical returns, to a full extent from the panel data, we employ Gated Recurrent Units (GRU) [10] as the backbone model. GRU has been widely used in natural language processing and receive remarkable achievements in modeling sequential information. Denote the input feature as $\mathbf{x}_{i,t} \in \mathbb{R}^P$ for stock i at time t with P as the number of features. The general idea of GRU is to recurrently project the input time series into the hidden representation distribution space. At each time step, the GRU learns the hidden representation $\mathbf{h}_{i,t}$ by jointly leveraging current input $\mathbf{x}_{i,t}$ and previous hidden representation $\mathbf{h}_{i,t-1}$ in a recursive manner:

$$\mathbf{h}_{i,t} = \text{GRU}(\mathbf{x}_{i,t}, \mathbf{h}_{i,t}). \quad (6)$$

Besides, in order to adaptively select the most important information from different time steps, we also use the attention mechanism [17, 24] to aggregate all hidden representations.

4.1.2 Cross-Sectional Transformation. The aforementioned temporal transformation simply treats each stock independently, which, however, is able to represent the transformation that involves relations among different stocks. For example, a common transformation in human designed factors is subtracting stock return from the sector return it belongs to get the residual momentum factor [7]. Such a transformation requires the model to be aware of the relations of individual stocks in addition to the temporal transformation. To fulfill this goal, we introduce another branch of GRU with a Graph Attention Networks (GAT) [37] layer to model the cross-sectional transformation as illustrated in Future 1. The GAT layer not only can represent the traditional stock relations like sector, but can also dynamically capture latent relations with learnable attention weights.

Mathematically, given the raw features $\mathbf{X} \in \mathbb{R}^{N \times P}$ for N stocks at time t , the GAT layer first projects the input features into another space with a shared linear transformation $\mathbf{W} \in \mathbb{R}^{P \times P}$, and then performs self-attention $a: \mathbb{R}^P \times \mathbb{R}^P \rightarrow \mathbb{R}$ on the nodes (i.e., stocks) to compute the attention coefficients:

$$e_{ij} = \text{LeakyReLU}(a(\mathbf{W}\mathbf{x}_i, \mathbf{W}\mathbf{x}_j)). \quad (7)$$

The attention coefficients can indicate the similarity of stock i and stock j . A prior graph structure, e.g., sector or business relation, can also be injected into GAT by performing masked attention [37].

The second step in GAT is aggregating the information from stocks with high attention coefficients, which can be accomplished by normalizing the attention coefficients by the softmax function:

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}, \quad (8)$$

where \mathcal{N}_i contains all neighborhoods of the i -th stock. The normalized attention coefficients α_{ij} can thereby be used to compute a linear combination of the features by

$$\tilde{\mathbf{x}}_i = \text{LeakyReLU}\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}\mathbf{x}_j\right). \quad (9)$$

A shared GAT layer will be applied at all time stamps to get the aggregated information $\tilde{\mathbf{x}}_{i,t}$. As we are interested in learning the relative information compared to the group of the related stocks for cross-sectional transformation, we will subtract the aggregated information from neighborhoods from the current feature and pass the residuals to the GRU module:

$$\tilde{\mathbf{h}}_{i,t} = \text{GRU}(\mathbf{x}_{i,t} - \tilde{\mathbf{x}}_{i,t}, \tilde{\mathbf{h}}_{i,t-1}), \quad (10)$$

where we use another GRU to further accomplish the temporal transformation from the residual features. The output hidden representations from both branches will be project into the risk factors by two separate linear projections $\mathbf{Q} \in \mathbb{R}^{P \times K_1}$ and $\tilde{\mathbf{Q}} \in \mathbb{R}^{P \times (K-K_1)}$:

$$\mathbf{f}_{i,t} = [\text{Norm}(\mathbf{Q}\mathbf{h}_{i,t}) \parallel \text{Norm}(\tilde{\mathbf{Q}}\tilde{\mathbf{h}}_{i,t})], \quad (11)$$

where \parallel represents the concatenation operation, and Norm is a normalization function that transforms the learned risk factors to have capitalization weighted zero mean and equal-weighted unit standard deviation [4].

4.2 Loss Design

Now we ponder the design of loss function (i.e. the learning objective) to fulfill three desired properties of the mined risk factors.

4.2.1 Explained Variance. The most important metric to evaluate the quality of a given set of risk factors is the proportion of the variance in the stock returns predicted with the risk factors. In statistics, the coefficient of determination, i.e., R^2 , is used to measure the ratio of explained variance. Let $\mathbf{F}_t = g_\theta(\mathbf{X}_t)$ denote the risk factors produced by our model at time t , then R^2 can be calculated as

$$R_t^2 = 1 - \frac{\|\mathbf{y}_t - \mathbf{F}_t(\mathbf{F}_t^\top \mathbf{F}_t)^{-1} \mathbf{F}_t^\top \mathbf{y}_t\|_2^2}{\|\mathbf{y}_t\|_2^2}, \quad (12)$$

where $\|\cdot\|_2^2$ represents the L^2 distance. Thence, we can use the below empirical loss function to train our model

$$\max \frac{1}{T} \sum_{t=1}^T R_t^2 \triangleq \min_{\theta} \frac{1}{T} \sum_{t=1}^T \frac{\|\mathbf{y}_t - \mathbf{F}_t(\mathbf{F}_t^\top \mathbf{F}_t)^{-1} \mathbf{F}_t^\top \mathbf{y}_t\|_2^2}{\|\mathbf{y}_t\|_2^2}. \quad (13)$$

4.2.2 Multi-collinearity Regularization. Directly training the model towards minimizing the objective defined in Equation 13 often ends up with highly correlated risk factors, which will make the regression coefficients in Equation 1 unstable and thus harm the estimation of the covariance matrix. A practical statistical ratio used for diagnosing multi-collinearity is variance inflation factor (VIF) [5], which quantifies the extent of correlation between i -th variate \mathbf{F}_i and the remaining covariates:

$$\text{VIF}_i = \frac{1}{1 - R_i^2} \quad (14)$$

where R_i^2 is the coefficient of determination for the regression of the i -th risk factors \mathbf{F}_i on the other covariates. A straight forward idea to reduce the multi-collinearity is using $\sum_{i=1}^K \text{VIF}_i$ as the regularization term, which however, requires K separate regressions to obtain all VIF ratios and is time consuming.

In this work, we introduce a computationally efficient alternative to obtain $\sum_{i=1}^K \text{VIF}_i$. In the remainder of this section, we use \mathbf{F}_{-i} to denote all the remaining factors excluding the i -th factor. Let \mathbf{F}_{-i} be the independent variables and \mathbf{F}_i as the dependent variable and through linear regression we can obtain the regression coefficients $\boldsymbol{\beta}_{-i} \in \mathbb{R}^{K-1}$. Then we have

$$\begin{aligned} \sum_{i=1}^K \text{VIF}_i &= \sum_{i=1}^K (1 - R_i^2)^{-1} \\ &= \sum_{i=1}^K \mathbf{F}_i^\top \mathbf{F}_i [(\mathbf{F}_i - \mathbf{F}_{-i} \boldsymbol{\beta}_{-i})^\top (\mathbf{F}_i - \mathbf{F}_{-i} \boldsymbol{\beta}_{-i})]^{-1} \\ &= \sum_{i=1}^K \mathbf{F}_i^\top \mathbf{F}_i [\mathbf{F}_i^\top \mathbf{F}_i - 2\boldsymbol{\beta}_{-i}^\top \mathbf{F}_{-i}^\top \mathbf{F}_i + \boldsymbol{\beta}_{-i}^\top \mathbf{F}_{-i}^\top \mathbf{F}_{-i} \boldsymbol{\beta}_{-i}]^{-1}. \end{aligned} \quad (15)$$

Since the factors are normalized to have unit deviation in Equation 11, we substitute $\mathbf{F}_i^\top \mathbf{F}_i = N$. And from standardized regression characteristics we also have $\mathbf{F}_{-i}^\top \mathbf{F}_{-i} \boldsymbol{\beta}_{-i} = \mathbf{F}_{-i}^\top \mathbf{F}_i$, then:

$$\begin{aligned}
\sum_{i=1}^K \text{VIF}_i &= N \sum_{i=1}^K (\mathbf{F}_i^\top \mathbf{F}_i - \boldsymbol{\beta}_{-i}^\top \mathbf{F}_{-i}^\top \mathbf{F}_i)^{-1} \\
&= N \sum_{i=1}^K (\mathbf{F}_i^\top \mathbf{F}_i - \mathbf{F}_i^\top \mathbf{F}_{-i} (\mathbf{F}_{-i}^\top \mathbf{F}_{-i})^{-1} \mathbf{F}_{-i}^\top \mathbf{F}_i)^{-1} \quad (16) \\
&= N \sum_{i=1}^K (\mathbf{F}^\top \mathbf{F})_{ii}^{-1} = N * \text{tr}((\mathbf{F}^\top \mathbf{F})^{-1}).
\end{aligned}$$

The second to last equation is supported by Schur complement without loss of generality. Compared with regressing each variate i on the remaining covariates to obtain $\sum_{i=1}^K \text{VIF}_i$, the trace of matrix $(\mathbf{F}^\top \mathbf{F})^{-1}$ is more feasible and efficient. Consequently, we can regularize multi-collinearity by minimizing $\text{tr}((\mathbf{F}^\top \mathbf{F})^{-1})$.

4.2.3 Multi-Task Learning. Another desired property of the risk factors is their stability, i.e., the value of risk factors should be continuous along the time. This is an important property as in Equation 3 and Equation 5 we assume the risk factors are constant in the estimation period. To ensure the stability of the risk factors, we design a multi-task learning objective that requires the learned risk factors \mathbf{F}_t can be used to explain the variance of stock returns in multiple forward periods:

$$\min_{\theta} \frac{1}{H} \sum_{h=1}^H \frac{\|\mathbf{y}_{t+h} - \mathbf{F}_t (\mathbf{F}_t^\top \mathbf{F}_t)^{-1} \mathbf{F}_t^\top \mathbf{y}_t\|_2^2}{\|\mathbf{y}_{t+h}\|_2^2}, \quad (17)$$

where H is the number of forward periods.

Finally, we put together all the loss terms from the above discussion and propose the following learning objective:

$$\min_{\theta} \frac{1}{T} \sum_{t=1}^T \left[\frac{1}{H} \sum_{h=1}^H \frac{\|\mathbf{y}_{t+h} - \mathbf{F}_t (\mathbf{F}_t^\top \mathbf{F}_t)^{-1} \mathbf{F}_t^\top \mathbf{y}_t\|_2^2}{\|\mathbf{y}_{t+h}\|_2^2} + \lambda * \text{tr}((\mathbf{F}_t^\top \mathbf{F}_t)^{-1}) \right], \quad (18)$$

where λ is a hyper-parameter that control the strength for the multi-collinearity regularization.

5 EXPERIMENTS

5.1 Experiment Settings

5.1.1 Data. The proposed framework can accept all common structured data like price, financial data, or even text, as input features to train the risk model. In this work, we use 10 common style factors in [28] as the input features, including *Size*, *Beta*, *Momentum*, *Residual Volatility*, *Non-linear Size*, *Book-to-Price*, *Liquidity*, *Earnings Yield*, *Growth* and *Leverage*. All these style factors will also serve as risk factors in the compared fundamental risk model to make sure our method have no advantage of introducing more information in the experiments. In fact, this is a more challenging experiment setting as these risk factors are summarized from decades of research efforts and our model must be able to learn better risk factors to beat human experts. These factors are implemented for China stock market and the studied universe covers all listed stocks. We will follow the temporal order to split data into training (2009/04/30~2014/12/31), validation (2015/01/01~2016/12/31) and testing (2017/01/01~2020/02/10).

5.1.2 Model Implementation. We implement the architecture designed in Section 4.1 with PyTorch. Both the two GRU modules in the designed network have 2 hidden layers and 32 hidden units. We let the number of learned risk factors equal to $K/2$ for both branches. The GAT module has the same hidden size as the input size, which is 10. We also add attention dropout in GAT with dropout probability set to 0.5. We pack all stocks in the same day as a single batch to ensure GAT can well capture stock relations and leverage gradient accumulation with the update frequency set to 64 to ensure the model see as many diverse samples as possible. We use Adam [20] as the optimizer with a fixed learning rate set as 0.0002. To further ensure the stability of the predictions, we also add exponential smoothing to the predictions with a smoothing factor set to 0.99. Unless otherwise specified, the regularization parameter λ is set to 0.01. The number of tasks in Equation 18 is set to 20, i.e., we encourage the learned risk factors can be used in the next 20 days. The model is trained on a single TITAN Xp GPU and it takes 6.5 hours to finish the model training for learning 10 risk factors.

5.2 Evaluation Protocol

5.2.1 Compared Methods. We compare the following methods for designing risk factors:

- **Fundamental Risk Model (FRM):** FRM directly use the 10 style factors described in Section 5.1.1 as risk factors. **FRM** shows the best performance of the risk factors designed by human experts.
- **Statistical Risk Model (SRM):** SRM takes the first 10 components with the largest eigenvalues by applying PCA on stock returns in the last 252 trading days. **SRM** shows the best performance of the traditional data-driven approach for learning latent risk factors.
- **Deep Risk Model (DRM):** DRM is the proposed method in this work, which learns 10 risk factors directly from the data described in Section 5.1.1.

As both **SRM** and **DRM** can learn any number of latent risk factors, we also add the results of **SRM (2x)** and **DRM (2x)** which will learn 20 risk factors.

5.2.2 Covariance Estimation. After the acquisition of the risk factors with different risk models, we will combine the risk factors with the country factor and 29 industry factors for cross-sectional regression with Equation 1. As the country factor introduces an exact collinearity into this equation, we follow [26] to introduce an extract constraint to force the coefficients of industry factors to have zero sum. After obtaining the factor returns, we then follow Equation 5 to derive the covariance matrix. To further ensure the estimated covariance more responsive to the market, we estimate the correlation and variance with an exponential halflife of 240 and 60 days respectively and we also apply Volatility Regime Adjustment [26] with an exponential halflife of 20 days to alleviate the forecasting bias.

5.2.3 Evaluation Metrics. To measure the risk forecasting performances of different risk models, we consider the below metrics:

- R^2 : R^2 measures the proportion of the explained variance of stock returns when using different sets of risk factors. The

calculation of R^2 is defined as Equation 12 and we will report the average R^2 in the test periods. A good risk model should have higher R^2 .

- **GMV**: We further show the performance of the estimated covariance matrix with different risk factors by constructing a Global Minimum Variance (GMV) portfolio and reporting its annualized volatility. Given a covariance matrix Σ , the GMV portfolio can be solved as $\frac{\Sigma^{-1}1}{1^T \Sigma^{-1}1}$ where 1 is a vector of ones. A good risk model should ensure the GMV portfolio has lower out-of-sample volatility.
- **GMV+**: This has the same objective as the GMV portfolio, except an extract no short-selling constraint is added. We use CVXPY [12] to solve the optimized portfolio.

To further show the characteristics of the learned risk factors, we will also report the following statistics:

- **Mean |t|**: This is the average absolute t-statistic of the regression coefficients in Equation 1. A higher absolute t-statistic means the factor is more significant.
- **Pct |t|>2**: As we have multiple t-statistics across the whole studied periods, we also report the percent of days when the corresponding risk factor is statistical significant ($|t| > 2$).
- **VIF**: This is the variance inflation factor (VIF) defined in Equation 14, which shows whether the current factor is collinear with the remaining factors. A lower VIF is always preferred.
- **Auto Corr.**: Auto correlation measures the stability of the risk factor. We will report the auto correlation of each factor with 1 day's lagging. A higher correlation is preferred.

5.3 Main Results

5.3.1 Risk Forecasting Performance. The risk forecasting metrics for the compared methods are summarized in Table 1. From the table, we have the following observations and conclusions:

- **DRM** can improve R^2 by 0.6% when generating the same number of risk factors compared to **FRM**, which shows the risk factors designed human experts still have room for improvement. Besides, with **DRM**'s ability to learn more risk factors, we can further improve R^2 by 1.9%. We also present R^2 over the entire testing period in Figure 2, from which we can see **DRM** can consistently outperforms the competitors.
- **DRM** can also reduce the volatility risk of both the **GMV** portfolio and the **GMV+** portfolio, by 1.0% and 0.6% respectively. As **GMV** portfolio is exactly determined by the covariance matrix, lower out-of-sample volatility means the estimated covariance is more accurate. This further demonstrated the learned risk factors can consistently improve covariance estimation.
- The other data-driven approach for risk factor mining, **SRM**, can also improve R^2 but still underperforms our method. Besides, the **GMV** portfolio solved with **SRM** often performs badly. Therefore, the proposed solution is superior to **SRM** for mining latent risk factors.

5.3.2 Factor Characteristics. We further present the statistical metrics of the learned risk factors in Table 2. We can see that the average absolute t-statistics for all factors are higher than 2 and more than

Table 1: The performance of the compared risk models measured by explained variance of stock returns (R^2) and portfolio risk forecasting accuracy (GMV and GMV+). \uparrow means the higher the better while \downarrow means the opposite.

Risk Model	R^2 (\uparrow)	GMV (\downarrow)	GMV+ (\downarrow)
Market	-	18.4%	18.4%
FRM	29.8%	12.6%	12.9%
SRM	30.1%	18.6%	13.5%
SRM (x2)	30.7%	16.5%	16.2%
DRM	30.4%	12.3%	12.9%
DRM (x2)	31.7%	11.6%	12.3%

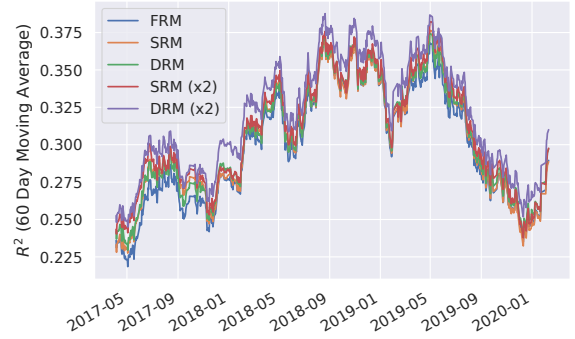


Figure 2: R^2 of the compared methods over the entire testing period. DRM (x2) consistently outperforms other methods.

50% of the time more factors have t-statistics higher than 2, which shows the learned risk factors are indeed significant. Further, all our risk factors have VIF scores around 1.0, which shows there is little multicollinearity in our risk factors. Last, the auto correlation of the learned risk factors is high, which shows our method can also ensure the stability of the learned risk factors.

Table 2: A summary statistics of the learned risk factors in terms of significance (t-statistics), orthogonality (VIF) and stability (auto correlation).

Factor ID	Mean t	Pct t >2	VIF	Auto Corr.
0	4.878	72.9%	0.984	0.995
1	3.559	65.9%	1.301	0.995
2	2.748	55.0%	1.180	0.997
3	2.173	45.9%	1.277	0.998
4	3.244	59.0%	1.309	0.997
5	2.831	56.0%	1.163	0.966
6	3.461	64.7%	1.007	0.994
7	2.697	56.8%	1.211	0.966
8	3.247	60.3%	1.293	0.990
9	2.699	55.6%	1.104	0.985

Figure 3 shows the cumulative factor returns, i.e., regression coefficients in Equation 1. We can see the regression coefficients

may change from positive to negative at a specific time (e.g., 3), which indicates the investment philosophy of the entire market may have changed. This indicates the risk factors from our model do capture certain trading patterns. However, unlike fundamental risk factors, the learned risk factors still lack interpretability. We will leave the explanation of the learned risk factor as future work.

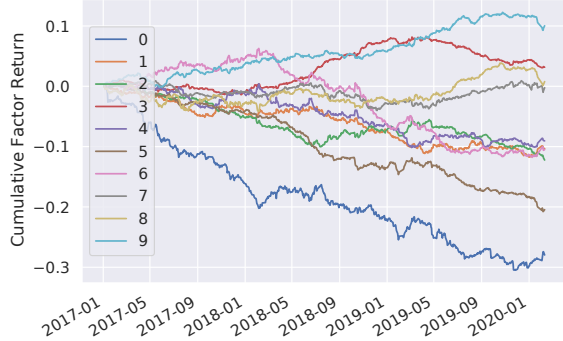


Figure 3: Cumulative factor returns (i.e., regression coefficients) of the learned risk factors.

5.4 Incremental Analysis

In this section, we want to answer the following research questions through incremental experiments:

- **RQ1:** Is GAT indeed a necessary component in the proposed neural network architecture?
- **RQ2:** What is the effect of the multi-collinearity regularization parameter λ in Equation 18?
- **RQ3:** Does the multi-task learning objective help stabilize the learned risk factors?

5.4.1 RQ1. We train the proposed network with and without the GAT module in parallel. To give a more convincing conclusion, we compare these two models with different number of risk factors as learning targets. The final R^2 of these models are summarized in Table 3. We can see that, introducing the GAT module can obtain higher R^2 in most cases. Besides, considering GAT is designed as a new transformation of the input data, we believe it will be necessary for mining more high quality risk factors if more information is introduced for risk factor mining.

Table 3: R^2 of DRM with or without GAT.

# Factors	10	12	14	16	18	20
w/ GAT	30.4%	30.6%	30.8%	31.1%	31.4%	31.7%
w/o GAT	30.3%	30.5%	30.9%	31.1%	31.3%	31.5%

5.4.2 RQ2. We further compare training the network with different regularization parameter λ . We report the VIF of the learned risk factors in Figure 4. It can be observed that as the regularization strength increases, the VIF gradually goes down. When setting $\lambda = 0.001$, the VIF can be higher than 5, which implies the existence of high collinearity among the learned risk factors.

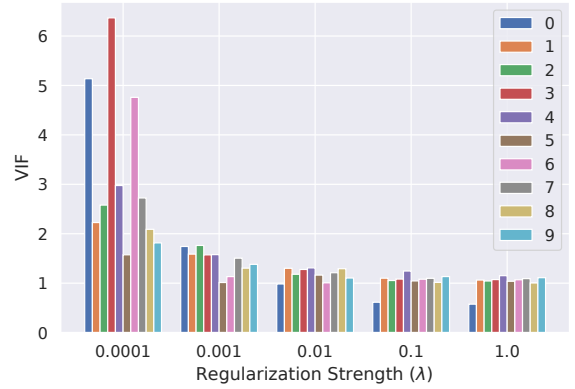


Figure 4: Variance inflation factor (VIF) of the learned risk factors with different regularization strength (λ) in Equation 18.

5.4.3 RQ3. Last, we verify the necessity of the multi-task objective in Equation 17 by comparing with the single-task objective (set $H = 1$ in Equation 17). Figure 5 shows the average auto correlation of all learned risk factors with either single-task objective or multi-task objective. We can see that without using the multi-task objective, the risk factor becomes unstable. Therefore, we should use the multi-task objective to optimize the model.

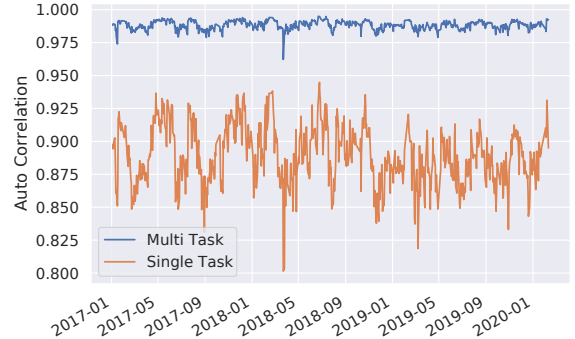


Figure 5: Auto correlation of learned risk factors with single task ($H = 1$) or multi task ($H = 20$) in Equation 18.

6 CONCLUSION

In order to improve the estimation of the covariance matrix of stock returns, we proposed a deep learning solution to effectively design risk factors. Fundamental risk model requires prodigious human professionals' effort, meanwhile statistical risk model lack non-linear capacity and is deficient in out-of-sample performance. Our *Deep Risk Model* (DRM) demonstrates impressive performance in forecasting risk with a more effective and adaptive estimate of covariance matrix. Our method also exhibits better factor characteristics in terms of high stability and low multi-collinearity. Ablation studies exemplifies the necessity and superiority of our design in the architecture and learning objective.

REFERENCES

- [1] Carol Alexander. 2001. Market models. *A Guide to Financial Data Analysis* 1 (2001).
- [2] Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. 2020. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations (ICLR)*.
- [3] Marco Avellaneda and Jeong-Hyun Lee. 2010. Statistical arbitrage in the US equities market. *Quantitative Finance* 10, 7 (2010), 761–782.
- [4] Suresh Balakrishnama and Aravind Ganapathiraju. 1998. Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing* 18, 1998 (1998), 1–8.
- [5] David A Belsley, Edwin Kuh, and Roy E Welsch. 2005. *Regression diagnostics: Identifying influential data and sources of collinearity*. Vol. 571. John Wiley & Sons.
- [6] Peter J Bickel and Elizaveta Levina. 2008. Covariance regularization by thresholding. *The Annals of Statistics* 36, 6 (2008), 2577–2604.
- [7] David Blitz, Joop Huij, and Martin Martens. 2011. Residual momentum. *Journal of Empirical Finance* 18, 3 (2011), 506–521.
- [8] Mark M Carhart. 1997. On persistence in mutual fund performance. *The Journal of finance* 52, 1 (1997), 57–82.
- [9] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 132–149.
- [10] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs.CL]*
- [12] Steven Diamond and Stephen Boyd. 2016. CVXPY: A Python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research* 17, 1 (2016), 2909–2913.
- [13] Fama Eugene and Kenneth French. 1992. The cross-section of expected stock returns. *Journal of Finance* 47, 2 (1992), 427–465.
- [14] Jianqing Fan, Yuan Liao, and Han Liu. 2016. An overview of the estimation of large covariance and precision matrices. *The Econometrics Journal* 19, 1 (2016), C1–C32.
- [15] Jianqing Fan, Yuan Liao, and Martina Mincheva. 2011. High dimensional covariance matrix estimation in approximate factor models. *Annals of statistics* 39, 6 (2011), 3320.
- [16] Jianqing Fan, Han Liu, and Weichen Wang. 2018. Large covariance estimation through elliptical factor models. *Annals of statistics* 46, 4 (2018), 1383.
- [17] Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua. 2019. Enhancing Stock Movement Prediction with Adversarial Training. *IJCAI* (2019).
- [18] Guanhao Feng, Stefano Giglio, and Dacheng Xiu. 2020. Taming the factor zoo: A test of new factors. *The Journal of Finance* 75, 3 (2020), 1327–1370.
- [19] Harry H Harman. 1976. *Modern factor analysis*. University of Chicago press.
- [20] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [21] Clifford Lam and Jianqing Fan. 2009. Sparsistency and rates of convergence in large covariance matrix estimation. *Annals of statistics* 37, 6B (2009), 4254.
- [22] Olivier Ledoit and Michael Wolf. 2004. Honey, I shrunk the sample covariance matrix. *The Journal of Portfolio Management* 30, 4 (2004), 110–119.
- [23] Olivier Ledoit and Michael Wolf. 2012. Nonlinear shrinkage estimation of large-dimensional covariance matrices. *The Annals of Statistics* 40, 2 (2012), 1024–1060.
- [24] Hengxu Lin, Dong Zhou, Weiqing Liu, and Jiang Bian. 2021. Learning Multiple Stock Trading Patterns with Temporal Routing Adaptor and Optimal Transport. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21)*. ACM.
- [25] Harry Markowitz. 1952. Portfolio selection. *The journal of finance* 7, 1 (1952), 77–91.
- [26] J. Menchero, D. J. Orr, and J. Wang. 2011. The Barra US Equity Model (USE 4) Methodology Notes.
- [27] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
- [28] DJ Orr and Igor Mashtaler. 2012. Supplementary Notes on the Barra China Equity Model (CNE5). (2012).
- [29] Philippe Rigollet and Alexandre Tsybakov. 2012. Estimation of covariance matrices under sparsity constraints. *arXiv preprint arXiv:1205.1210* (2012).
- [30] William F Sharpe. 1964. Capital asset prices: A theory of market equilibrium under conditions of risk. *The journal of finance* 19, 3 (1964), 425–442.
- [31] Aamir Sheikh. 1996. BARRA's risk models. *Barra Research Insights* (1996), 1–24.
- [32] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.
- [33] Lloyd N Trefethen and David Bau III. 1997. *Numerical linear algebra*. Vol. 50. Siam.
- [34] Jeffrey Uhlmann. 2018. A generalized matrix inverse that is consistent with respect to diagonal transformations. *SIAM J. Matrix Anal. Appl.* 39, 2 (2018), 781–800.
- [35] Charles F Van Loan. 1976. Generalizing the singular value decomposition. *SIAM Journal on numerical Analysis* 13, 1 (1976), 76–83.
- [36] M Alex O Vasilescu and Demetri Terzopoulos. 2005. Multilinear independent components analysis. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. IEEE, 547–553.
- [37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- [38] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2, 1-3 (1987), 37–52.