Conditional Latent Factor Models Via Econometrics-Based Neural Networks

Job Market Paper

Hao Ma [*]

First Draft: October 20, 2021
This Version: January 15, 2022

Abstract

I develop a hybrid methodology that incorporates an econometric identification strategy into artificial neural networks when studying conditional latent factor models. The time-varying betas are assumed to be unknown functions of numerous firm characteristics, and the statistical factors are population cross-sectional OLS estimators for given beta values. Hence, identifying betas and factors boils down to identifying only the function of betas, which is equivalent to solving a constrained optimization problem. For estimation, I construct neural networks customized to solve the constrained optimization problem, which gives a feasible non-parametric estimator for the function of betas. Empirically, I conduct my analysis on a large unbalanced panel of monthly data on US individual stocks with around $30,000$ firms, 516 months, and 94 characteristics. I find that 1) the hybrid method outperforms the benchmark econometric method and the neural networks method in terms of explaining out-of-sample return variation, 2) betas are highly non-linear in firm characteristics, 3) *two* conditional factors explain over 95% variation of the factor space, and 4) hybrid methods with literature-based characteristics (e.g., book-to-market ratio) outperform ones with COMPUSTAT raw features (e.g., book value and market value), emphasizing the value of academic knowledge from an angle of Man vs. Machine.

Keywords: Large Unbalanced Panel, Unobservable Factors, Time-Varying Betas, High-dimensionality, Identification Strategy, OLS, Artificial Neural Networks.

[*] Swiss Finance Institute (SFI) & Università della Svizzera italiana (USI, Lugano), Email: hao.ma@usi.ch. I am grateful to my advisor, Patrick Gagliardini, for his invaluable support and patient guidance. I thank Federico Carlini, Pierre Collin-Dufresne, Eric Ghysels, Lorenz Kueng, Hanno Lustig, Loriano Mancini, Raphaël Parchet, Markus Pelger, Giovanni Pica, Mirco Rubin, Olivier Scaillet, Paul Schneider, Martin Weidner, and participants at EWMES 2021, CFE-CMStatistics 2021, SFI Research Days 2021, the SoFiE seminar series, and seminars at the Shanghai University of Finance and Economics, Shandong University, Swiss Finance Institute, Università della Svizzera italiana for their valuable feedbacks and discussions.

1 Introduction

In the most general sense, the paper tries to answer the following two questions. 1) Can researchers develop hybrid methods that combine advantages from traditional econometric theories (e.g. interpretability) and artificial neural networks estimation methods (e.g. high-dimensional and non-parametric) for asset pricing models? 2) If yes, are the hybrid methods better than pure econometric or pure neural networks methods in terms of out-of-sample performance?

Unlike most papers that adopt machine learning techniques in the existing financial or econometric structures, this paper builds both asset pricing models and econometric conditions into a broad framework of artificial neural networks. With such a different approach, this paper gives affirmative answers to both questions when studying a specific class of asset pricing models. Under the setup of this paper, a hybrid methodology equipped with an identification strategy from econometric theories and neural networks estimation framework from deep learning literature displays good interpretability, flexible assumptions for functions, strong dimension-reduction capability, lower computational cost, and better out-of-sample performance.

[Insert Figure 1 about here]

The asset pricing models this paper focuses on are conditional latent factor models, where betas are time-varying and factors are unobservable. As comparison, observable factor models with constant betas, such as CAPM or Fama-French factor models, suffer from misspecification issues as hundreds of empirical factors leave a wide latitude for researchers to choose from. A solution to the misspecification issue is, therefore, to use statistical (latent) factors, for which the Principal Component Analysis (PCA) is often adopted. On the other hand, the risk exposures (betas) of a firm are also likely to vary over time because of the changes in its fundamentals. For example, Amazon started its business as an online shopping website but now has also become the biggest cloud computing provider in the world. Its stocks therefore are likely to be exposed to different types of common risks at present compared with when the company was listed. Therefore, the risk exposures should be time-varying and possibly driven by numerous firm characteristics. As a result, the misspecification concern in selecting factors and the time-variation feature in betas bring researchers' attention to conditional latent factor models.

There have been many studies on the conditional latent factor models, most of which follow a pure econometric approach while some most recent ones use machine learning techniques. All papers studying such models have three major difficulties to overcome. One is how to disentangle the time-varying betas from the statistical factors as both are not observable. The second is how to give a feasible nonparametric estimator for the time-varying betas, if they are assumed to be a general function of firm characteristics. The third one is how to ensure the statistical factors to be constructible by portfolios so that investors can trade on them. However, few papers so far have successfully accommodated all of them.

The line of literature that adopts traditional econometric methods include work such as Connor, Linton, and Hagmann (2012), Kelly, Pruitt, and Su (2017, 2019), Pelger and Xiong (2018), Zaffaroni (2019) and so on. In order to give feasible estimators for betas and factors, most methods assume an a-priori specification for the betas. For instance, Connor, Linton, and Hagmann (2012) treat the betas as a weighted additive sum of univariate characteristic-based functions. Kelly, Pruitt, and Su (2017)'s IPCA method (Instrumented Principal Component Analysis) assume the time-varying betas to be linear in firm characteristics. Although assuming the function of betas to be known provides a solution to the identification problem, econometricians often encounter with two consequent problems. First, there are no good reasons to believe that the betas follow some specific dynamics such as a linear structure in characteristics. So what functional form should econometricians use? Secondly, the literature has documented hundreds of characteristics that seem to correlate with common risks. How should econometricians select the useful ones and reduce the dimensionality?

Such issues thus motivate researchers to assume betas as unknown functions of a large number of asset-level characteristics. In the recent literature adopting machine learning technologies in asset pricing, Gu, Kelly, and Xiu (2021) develops the Autoencoder Asset Pricing models that non-parametrically estimates both time-varying betas and statistical factors with conditional autoencoder neural networks. Nonetheless, the paper does not discuss the identification strategy, and the statistical factors in this method are not tradeable as they are assumed to be non-linear functions of a cross-section of returns. Gagliardini and Ma (2020) also use asset characteristics as conditioning information in describing betas in a model-free way. Their method comes up with identification and statistical inference on the factor space with machine learning techniques.

However, their method only gives consistent estimators for latent factors but not for time-varying betas.

This paper tackles the above issues by incorporating an identification strategy into a broad framework of artificial neural networks (ANN). Specifically, I assume the time-varying betas to be a general function of firm-level characteristics. As a consequence of the linear factor models, the statistical factors would be the coefficients of the time-varying betas, assuming betas were already given. Therefore, the true functional form of the statistical factors are the population OLS estimator by regressing returns on betas. As a result, statistical factors can be regarded as beta-based portfolios and therefore tradeable. Since the factors are now deterministic functions of returns and beta, the identification of both betas and factors boils down to identifying only the betas. Under a few very mild econometric constraints, I then prove that the true functions of betas is uniquely identified by solving a constrained optimization problem.

For estimation, econometricians find it hard to give a feasible non-parametric estimator to the high-dimensional function for betas. This is because traditional non-parametric econometric methods, such as kernel regression or polynomial regression, can only deal with up to 3 to 4 independent variables at the same time. Therefore, it is necessary to resort to the machine learning toolbox. More importantly, the machine learning tool should also be able to solve for the constrained optimization problem proposed in the identification process.

There are many machine learning tools that can give non-parametric estimators for a high-dimensional function, such as random forest and support vector machines. However, artificial neural networks turns out to be one of the few tools that not only serves as a universal approximator for any continuous functions, but also gives numerical solutions to generic optimization problems. Consequently, this paper designs artificial neural networks with structures that have one-to-one mapping to the ones in the identification problem. Specifically, I use standard feedforward neural networks to approximate time-varying betas and an OLS estimator for the statistical factors in the framework of artificial neural networks. With initialized parameters for the function of betas, the algorithm will solve for the optimal solution to the built-in optimization problem with all econometric conditions imposed. And the feasible estimator of the function of time-varying betas will be recovered from the trained feedforward neural networks. In the end, the hybrid methodology solves simultaneously the misspecification issue in time-varying betas and untradeability issue in statistical factors, obtaining flexibility in function forms and interpretability in the neural network structures.

Empirically, I conduct the analysis on a large unbalanced panel of monthly data of the US individual stock. I have around $30,000$ firms and 516 months. I use 94 firm characteristics for modeling the time-varying betas. To evaluate the performance of the model, I report an out-of-sample total R^2 as a measure of explained variation of returns. I then compare it with the benchmark econometric methods, the IPCA method developed in Kelly, Pruitt, and Su (KPS, 2019), and the benchmark artificial neural networks methods, the conditional autoencoder networks method by Gu, Kelly, and Xiu (GKX, 2021). My results show that the hybrid model outperforms both benchmark methods in terms of explaining variation in out-of-sample stock returns.

One additional advantage that the hybrid method has is that the number of parameters to be estimated in neural networks is considerably reduced by 99.7% compared with the GKX method. In their settings, the authors use ANN estimators for betas and factors, which result in around $3,000$ and $960,000$ parameters[1], separately. On the contrary, the number of parameters in the econometrics-based ANN will be merely $3,000$ because the statistical factors are estimated by OLS, in which no parameter actually incurs.

To further test how much nonlinearities account for in beta's specification, I resort to the Post-Lasso method. Specifically, I use Lasso to run a panel regression of beta estimates on the firm characteristics for dimension reduction. After dropping negligible firm characteristics in the first step, I run a standard OLS regression and report the adjusted R^2. My result shows that most characteristics have a significant impact on the betas. However, a linear specification on betas only delivers an R^2 of less than 45% for the statistical factors. The result strongly suggests that betas are highly non-linear in the pre-defined characteristics from the literature.

In determining the number of conditional factors, I propose the cumulative explanatory power ratio (CEP) that measures how much variation different number of factors account for in the conditional factor space. My result shows that 80% of variation in factor space can be captured by only 1 factor, and over 95% of variation can be captured by 2 factors. As comparison, researchers usually take at least 3 factors and often 5 to 6 in the unconditional factor models. These results show that

[1] Let us assume we both have $30,000$ firms and 94 characteristics in the dataset and always use 1 hidden layer with 32 neurons in the neural networks.

the dimension of the conditional space is smaller than that of the unconditional space, indicating that some variation in unconditional factor models may result from variation in time-varying betas.

Regarding what characteristics drive the dynamics of different betas, this paper ranks the contribution of characteristics to the variation of different betas. The analysis shows that the beta associated with the first factor, which explains over 80% of the variation in factor space, is mostly driven by unconditional betas and turnover. This finding makes the first factor look like some "Market+Turnover" mixed factor. The second beta, the loading for the second factor accounting for 15% additional variation in the factor space, is mostly explained by volatility and momentum.

Furthermore, I carry out analysis to understand if the conditional latent factor space captures business cycles. I compute the conditional canonical correlation between the factor estimates and 13 contemporaneous financial indicators, including 8 variables from Goyal and Welch (2008). The result shows that the latent factor estimates on average can be well spanned by the financial indicators, suggesting that literature does good job in extracting common risk sources. Meanwhile, the correlation is much lower during financial crisis. This result indicates rare systematic risks specific to the financial crisis are not well captured by factor models where only firm-level characteristic are assumed to drive the time-varying betas.

Following a recent topic of Man vs. Machine, I also propose a comparison analysis of hybrid models with different versions of firm characteristics. Version 1 is to use the literature-based characteristics that are proposed by researchers (e.g., book-to-market ratio). Version 2 is to use the raw financial features directly extracted from COMPUSTAT dataset (e.g., book value and market value). It turns out that hybrid models with literature-based characteristics perform better than COMPUSTAT raw features in terms of out-of-sample total R^2. Such conclusions emphasize the value of human knowledge in academic research in the era of artificial intelligence.

This paper contributes to the existing literature both methodologically and empirically. Methodologically, it is the first paper to develop a hybrid methodology (Econometrics + Neural Networks) for conditional latent factor models (Asset Pricing). Put more vividly, this paper shows the possibility of teaming up an econometrician and a deep learner for a project started by an asset pricer. This new way of working together will bring advantages that no one of them alone is able to deliver. Specifically, the asset pricer is interested in conditional latent factor models and wants the function of betas to be model-free and the statistical factors to be tradeable. The econometrician gives econometric constraints based on the asset pricer's request and converts the issue into solving for a constrained optimization problem. Later, the deep learner builds up the econometrics-based neural networks with a one-to-one mapping between the network structures and econometric constraints. Finally, three of them together solve the problem by developing a hybrid methodology, which is equipped with interpretable structures, model-free assumptions, high-dimensionality, lower computational cost, and better out-of-sample performance. Empirically, the paper shows evidence that time-varying betas are highly-nonlinear in firm characteristics. In the topic of Man vs. Machine, the paper provides innovative evidence of the value of human research by showing models with literature-based characteristics outperform ones with COMPUSTAT raw features in terms of explaining individual returns variations out-of-sample.

The rest of the paper is organized as follows. Section 2 introduces the conditional factor model and explains the identification strategy of extracting time-varying betas and latent factors. Section 3 elaborates in detail how to construct neural networks with identification conditions. Section 4 carries out empirical analysis on the US stock market. Section 5 concludes the paper.

2 Identification Strategy

Consider the following asset pricing model for asset i at time t:

$$y_{i,t} = b'_{i,t-1} f_t + u_{i,t}, \tag{1}$$

where $y_{i,t}$ is the individual stock excess return on asset i in period t, for $i = 1, \cdots, n$ and $t = 1, \cdots, T$. Here, $b_{i,t-1}$ is a $k \times 1$ time-varying vector of factor loadings for asset i at period t, and is often modeled as a function of firm characteristics. The latent vector f_t represents the systematic risk factors, which is unobservable to the econometrician. The idiosyncratic error terms are denoted by $u_{i,t}$.

As can be seen from equation (1), it is challenging to disentangle $b_{i,t-1}$ from f_t because both are time-varying and unobservable, unless assumptions are made on the specification of $b_{i,t-1}$. One treatment is to assume betas have some form of linear relationship with firm characteristics like KPS, which, however, may further cause misspecification issue. Alternatively, one may assume time-varing betas and statistical factors are both unknown functions of firm characteristics and a cross-section of individual returns, respectively (GKX). However, the statistical factors untradeable as they are nonlinear in cross-sectional returns. Even if the functions turn out to be linear and tradeable, the portfolio weights for constructing factors will be constant over time. This means the statistical factors are not conditional but unconditional.

In order to accommodate model-free time-varying betas and tradeable conditional factors, I give the following identification conditions by generalizing Assumption 2 in Gagliardini and Ma (2020). For simplicity, I hereafter assume the number of true factors to be a known constant k and impose the no-arbitrage restriction by setting abnormal returns to zero.

$$\begin{aligned}
&(i) \quad b_{i,t} = b^0(w_{i,t}),\\
&(ii) \quad \mathbb{E}^c\left[b^0(w_{i,t})u_{i,t+1}\right] = 0,\\
&(iii) \quad \mathbb{E}^c\left[b^0(w_{i,t})b^0(w_{i,t})'\right] \text{ is full-rank,}
\end{aligned}$$

where $w_{i,t-1}$ is a $K \times 1$ vector of observed instrumental variables, and the unknown function $b^0(\cdot)$ is assumed non-constant, bounded and continuous and is independent of asset i at time t. Here, $\mathbb{E}^c[\cdot]$ is defined as the large n limit of cross-sectional averages:

$$\mathbb{E}^c[x_{i,t}] = \operatorname*{plim}_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} x_{i,t}$$

Condition (i) states that $b_{i,t}$ is an unknown function of firm characteristics $w_{i,t-1}$ and there exists a true function $b^0(\cdot)$, which is independent of asset i and time t [2]. Conditions (ii) and (iii) require the betas to be well-behaved in the cross-section. Specifically, condition (ii) says instrument variables $w_{i,t-1}$ and error terms $u_{i,t}$ are cross-sectionally independent to some extent [3], which will eliminate the interference from error terms in the cross-sectional aggregation. The full-rank property in condition (iii) implies that the number of factors is k at each period. Otherwise, a reduced-rank second moment indicates that some betas are collinear and therefore redundant.

If $b_{i,t-1}$ were given and the latent factors are defined by the population cross-sectional OLS estimator:

$$f_t = \mathbb{E}^c\left[b_{i,t-1}b_{i,t-1}'\right]^{-1}\mathbb{E}^c\left[b_{i,t-1}y_{i,t}\right] \qquad (2)$$

[2] This is also ensured in the estimation process.

[3] The cross-sectional independence between $x_{i,t}$ and $y_{i,t}$ is defined by: $Cov^c[g(x_{i,t}), h(y_{i,t})] = 0$ for $\forall g(\cdot), h(\cdot)$, where Cov^c denotes the cross-sectional covariance

However, the constraints above are not sufficient to uniquely identify the latent factors. To show this, consider $b_{i,t-1} = Ab^0(w_{i,t-1})$ that also satisfies above conditions, where matrix A is an orthogonal matrix so that $AA' = I_k$:

$$b'_{i,t-1}f_t = b^0(w_{i,t-1})'A'Af_t^0 = b^0(w_{i,t-1})f_t^0,$$

where f_t^0 represents the factors associated to the true betas $b^0(\cdot)$ and $f_t = Af_t^0$.

This is to say, both betas and factors are identifiable up to an unknown rotation A [4]. Consequently, all linear transformations of $b^0(\cdot)$ are equivalent solutions to minimizing pricing errors. Moreover, it also means in practice the solution can vary across different samples in the estimation process due to marginal changes in the data.

GKX's method is also subject to such an identification issue. In their assumptions, both time-varying betas and latent factors are unknown functions in forms of $b_{i,t} = b(w_{i,t})$ and $f_t = f(y_t)$, where y_t stacks the cross-section of individual returns at time t. Consider $b(w_{i,t}) = Db^0(w_{i,t})$ and $f(y_{i,t}) = (D')^{-1}f^0(y_t)$ where D is non-singular, solutions up to linear transformations will be equivalent to each other: $b(w_{i,t})'f(y_t) = b(w_{i,t})'D'(D')^{-1}f(y_t)$. Because the transformation is unknown and sample-specific as shown above, estimates from different samples will not be comparable.

To uniquely identify function $b^0(\cdot)$, I give the following Identification Assumption and Normalisation Restriction.

Identification Assumption. *Time-varying betas and error terms are assumed to behave well such that:*

$$\begin{aligned} &(i) \quad b_{i,t} = b^0(w_{i,t}), \\ &(ii) \quad \mathbb{E}^c\left[b(w_{i,t})u_{i,t+1}\right] = 0, \quad \forall b(\cdot), \end{aligned}$$

where the unknown function $b(\cdot)$ includes the true function $b^0(\cdot)$ and is assumed non-constant, bounded and continuous and is independent of asset i at time t.

[4] Here, one may argue the rotation should be time-varying.However, the rotation must be static in this case because $b(\cdot)$ is assumed to be independent of i and t. A time-varying rotation A_t will only appear when $b(\cdot)$ becomes dependent on t (e.g. by including state variables Z_t in its argument).

Normalisation Restriction. *Without loss of generality, the following normalization restrictions hold:*

$$
\begin{aligned}
&(i) \quad \mathbb{E}\left[b^0(w_{i,t-1})b^0(w_{i,t-1})'\right] = I_k, \\
&(ii) \quad \mathbb{E}\left[f_t^0 f_t^{0\prime}\right] \textit{ is diagonal, with distinct diagonal entries ranked in decreasing order} \\
&(iii) \quad \mathbb{E}\left[f_t^0\right] \geq 0
\end{aligned}
$$

With Normalisation Restriction, matrix A is identified by an identity matrix. As a result, both time-varying betas and latent factors are now uniquely identifiable. I now conclude the above argument by giving the following proposition. For its proof, please refer to the appendix.

Proposition. *Under Identification Assumption and Normalisation Restriction, function $b^0(\cdot)$ is uniquely identified by solving the following constrained least square criterion:*

$$
b^0(\cdot) = \underset{b(\cdot)}{\arg\min}\ \mathbb{E}\left[(y_{i,t} - b(w_{i,t-1})' f_t)^2\right]
$$

$$
s.t. \quad \begin{cases}
(i) \quad f_t = \mathbb{E}^c\left[b(w_{i,t-1})b(w_{i,t-1})'\right]^{-1} \mathbb{E}^c\left[b(w_{i,t-1})y_{i,t}\right], \textit{ for any } t = 1, ..., T, \\
(ii) \quad \mathbb{E}\left[b(w_{i,t-1})b(w_{i,t-1})'\right] = I_k \\
(iii) \quad \mathbb{E}\left[f_t\ f_t'\right] \textit{ is diagonal, with distinct entries in descending order} \\
(iv) \quad \mathbb{E}\left[f_t\right] \geq 0
\end{cases} \tag{3}
$$

Notably, the constrained minimization problem does not have a closed-form solution because the first-order conditions are not linear for function $b(\cdot)$. Furthermore, the firm characteristics are high-dimensional. Because traditional non-parametric econometric approaches, such as kernel regressions, are not good at dealing with high-dimensional problems, it is necessary to resort to the machine learning toolbox in the estimation process.

3 Estimation Via Artificial Neural Networks

According to Proposition, the betas and the factors will be solutions of the constrained optimization problem (3). Let us define $n \times K$ matrix W_t and $n \times 1$ vector y_t as the collection of the cross-section of $w_{i,t}$ and $y_{i,t}$ at time t. Because the data pair (W_{t-1}, y_t) will be treated as one observation in the estimation process, I re-express the problem in the matrix form:

$$\min_{b(\cdot)} \frac{1}{nT}\sum_{t=1}^{T}(y_t - B_{t-1}\hat{f}_t)'(y_t - B_{t-1}\hat{f}_t)$$
$$\text{s.t.} \begin{cases} \text{(i)} & B_{t-1} = b_{\odot}(W_{t-1}) \\ \text{(ii)} & \hat{f}_t = (B'_{t-1}B_{t-1})^{-1}B'_{t-1}y_t, \\ \text{(iii)} & \frac{1}{nT}\sum_{t=1}^{T} B'_{t-1}B_{t-1} = I_k, \\ \text{(iv)} & \frac{1}{T}\sum_{t=1}^{T}\hat{f}_t\hat{f}'_t \text{ is diagonal, with distinct} \\ & \text{entries ranked in descending order,} \\ \text{(v)} & \frac{1}{T}\sum_{t=1}^{T}\hat{f}_t \geq 0 \end{cases} \tag{4}$$

where $b_{\odot}(\cdot)$ is the asset-wise operation of W_{t-1} with function $b(\cdot)$. The argument of the minimization problem (4) is function $b(\cdot)$ where $b : \mathbb{R}^K \to \mathbb{R}^k$.

3.1 The feasible tool – artificial neural networks

Traditional non-parametric econometric methods are not good at handling high-dimensional data. For instance, kernel regression estimators will have low convergence rates when the sparsity of data points increases with the dimension of independent variables (curse of dimensionality). As a result, papers like Connor, Hagmamn and Linton (2012) and Fan, Liao and Wang (2015) both assume an additive form in the betas in order to non-parametrically estimate functions of interest and avoid the curse of dimensionality. In this paper, condition (i) in problem (4) assumes betas to be an unknown function of a large-dimensional firm characteristic vector.

There are numerous machine learning techniques for non-parametric estimation with high-dimensional data. For example, popular ones include k-nearest neighbors, decision trees, SVM (support vector machines) etc. Even Lasso (least absolute shrinkage and selection operator) as a linear approach can be used for approximating nonlinear functions by using as regressors basis functions of independent variables (e.g. Belloni and Chernozhukov (2013)). If the task is to non-parametrically estimate the conditional expectations of a variable, any machine learning methods mentioned above will potentially work well. For example, suppose we can observe the true values of betas, then the unknown function $b(\cdot)$ can be estimated by simply regressing these true betas on firm characteristics with some suitable machine learning approach. Nonetheless, these true values of time-varying betas are not observable, and even noisy estimates of them are hard to obtain. More importantly, Proposition states that betas and factors are identifiable only if all of the identification conditions hold at the same time. Hence, all the conditions should be imposed simultaneously in the estimation process.

To summarize, a big number of firm characteristics introduces large-dimensionality issue. Assuming the functional form of betas to be unknown asks for a non-parametric estimation approach. And the need for solving for both betas and factors simultaneously requests all constraints to be deployed during the estimation process. As traditional econometric methods cannot deliver non-parametric estimates with high-dimensional input variables, we are essentially looking for certain machine learning tools that can well preserve the structure of the optimization problem (4). So, does there exist one in the machine learning toolbox?

The answer is yes. It turns out that neural networks with structured layers can handle all three issues at once. Firstly, the Universal Approximation Theorem states that a standard single-layer feedforward neural network [5] can approximate any continuous function (Hornik, Stinchcombe, and White 1989; Cybenko 1989). The method has been widely applied in many high-dimensional settings like image recognition and has showed great power in solving complex problems. This is to say, the unknown function of betas can be non-parametrically estimated by a standard single-layer structure, which will be used to deploy condition (i). Secondly, neural networks are extremely flexible as mathematical operations are allowed to be carried on different components (usually called tensors[6]). Notice that the objective function and the conditions (ii) to (iv) in the optimization problem (4) involves common operations, it is therefore feasible to construct neural networks equipped with the objective function and the constraints from the identification problem (4).

[5] A single-layer neural network represents the simplest form of neural network, in which there is only one hidden layer.

[6] A tensor is a data structure of any dimensions, which can be a scaler, a vector, a matrix or an array of higher dimensions.

Lastly, one has good reasons to be concerned about the non-convexity[7] of problem (4) because such problems normally have numerous local minima. Alghouth Proposition proves that there exists only one global optimum to the problem in population, the possibility of being trapped in the local optima in sample makes it very challenging for any methods to find the global one. Fortunately, neural networks are able to provide solutions to non-convex problems as long as they adopt some variant of stochastic gradient descent (SGD). Such optimizer introduces noise by considering fewer points so the algorithm can jump out of local minima and reach the global one.

3.2 Build the econometrics-based neural networks

In general, artificial neural networks equipped with corresponding objective functions and constraints can always provide solutions to the constrained optimization problem. More importantly, the method gives non-parametric estimation of unknown functions rather than numerical values. Therefore, the estimation process boils down to building the constrained optimization problem (4) from the identification process into artificial neural networks, for which I call them "econometrics-based neural networks".

In this section, I will explain how to build problem (4) into neural networks step by step by reviewing what has been done in the related literature in machine learning in asset pricing. Meanwhile, I will also compare the computational cost of each model measured by the number of parameters that need to be estimated via neural networks.

Let us first consider the neural networks used in Gu, Kelly and Xiu (2020). Following this paper's notation, their model is expressed as:

$$y_t = h_{\odot}(W_{t-1}) + u_t, \tag{5}$$

where $h_{\odot}(\cdot)$ is the asset-wise operation of W_{t-1} with unknown function $h(\cdot)$ and depends neither on i nor t. This is a direct deployment of standard feedforward neural networks with no asset pricing structures imposed. It is a convenient strategy in their case because one of their main purposes is to achieve good predictability. Although they show better out-of-sample performance than

[7] The optimization problem is not convex because the criterion is not quadratic in betas and the functional form in the equality constraint (i) is unknown and not affine. For a problem to be convex it must satisfy three requirements: the objective function must be convex, the inequality constraint functions must be convex, and the equality constraint functions must be affine (Boyd and Vandenberghe (2004))

previous asset pricing models, their model loses interpretability because no economic structures are imposed. Figure 2a shows the standard single-layer feedforward neural networks that correspond to the estimation of model (5).

[Insert Figure 2 about here]

As mentioned above, neural networks are highly parameterized so the computational cost may soar when the model is too complex. To measure the computational cost, I use the number of parameters to be estimated as the proxy. Figure 3a visualizes the parameters in the neural networks in model (5). Each arrow is associated with a weight parameter, and each neuron in the hidden and output layers is associated with an intercept parameter. The number of parameters in model (5) is equal to $(K+1) \times M + M + 1$, where K is the dimension of the predictors (number of firm characteristics) and M is the number of neurons in the hidden layer.

[Insert Figure 3 about here]

If the true betas were observable, or some estimates were available, time-varying betas can be approximated with a single-layer neural network with firm characteristics as input variables $B_t = b_{\odot}(W_t) + u_t$. Although such a regression is infeasible as the true betas B_t are not observable, it is feasible to deploy constraint (i) as an intermediate step in a broader neural networks structure:

$$B_t = b_{\odot}(W_t), \tag{6}$$

Figure 2b shows the corresponding neural network. Similarly, the number of parameters is equal to $(K+1) \times M + (M+1) \times k$ as shown in Figure 3b, where k is the number of factors which is assumed known and constant.

Now, let us come to the latent factor models with time-varying betas. For estimating both betas and factors, GKX proposes the following model:

$$\begin{aligned} y_t &= B_{t-1} f_t + u_t, \\ \text{s.t.} &\begin{cases} B_{t-1} = b_{\odot}(W_{t-1}) \\ f_t = f(y_t) \end{cases} \end{aligned} \tag{7}$$

where $f(\cdot)$ is an unkown function for risk factors with the cross-section of returns as its argument. Figure 4 shows the architecture of the conditional autoencoder model. As factors are assumed to be unknown functions of returns in model (7), an additional neural network is thus added to estimate $f(y_t)$ in the first step.

[Insert Figure 4 about here]

First, the algorithm assigns arbitrary parameters to estimate function $b(\cdot)$ and $f(\cdot)$. Next, fitted values of returns are then produced by $\hat{y}_t = \hat{b}_{\odot}(W_{t-1})\hat{f}(y_t)$. And similarly, parameters in neural networks for approximating $b(\cdot)$ and $f(\cdot)$ will be updated with some gradient descent algorithm.

Notably, the first step involves two separate neural networks for estimating two different functions. Step 1 (Figure 4) gives the same $(K+1) \times M + (M+1) \times k$ parameters for estimating betas if we keep the same structure as in model (6). Step 2 further introduces $(n+1) \times M + (M+1) \times k$ for estimating the factors if we assume the same number of neurons in the hidden layer[8], where n is the number of assets.

In model (4), factors are products of betas and returns and therefore do not need to be estimated. More visually, the neural network for estimating $f(y_t)$ in Figure 4 is simply replaced by matrix products $(B'_{t-1}B_{t-1})^{-1}B'_{t-1}y_t$ in Figure 5. This is a very important property as it removes the parameters to be estimated for $f(\cdot)$, which immediately implies dramatic decrease in the computational cost.

[Insert Figure 5 about here]

Specifically, the number of parameters in the hybrid model is the same $(K+1) \times M + (M+1) \times k$ as in model (6), which is much lower than $(n+1) \times M + (M+1) \times k$ in model (7) by Gu, Kelly and Xiu (2021). In reality, $n \geq 30,000$ and $K \leq 100$. And if we assume the number of factors $k = 3$, then the inclusion of the extra neural network increases the number of parameters by a multiplier of at least $\frac{n+k+1}{K+k+1} \approx \frac{n}{K} \geq 300$.

[8] In general, the number of neurons in the hidden layers should be much bigger than M.

3.3 Impose the normalisation restrictions

There are two strategies for imposing the normalisation restrictions in (4), which are visualized in Figure 6. The dotted vertical line represents the normalisation restriction. One strategy is to normalise only after the algorithm stops and report the final estimates. The other is to normalise at each iteration when estimates are updated. As indicated in the figures, the two algorithms in theory are equivalent and should give the same estimates. In my application, I adopt the first algorithm because it has less computational cost.

4 Empirical Analysis

4.1 Data Description

I carry out the analysis of the US stock market. First, I extract the stock returns from the CRSP dataset and subtract them by the risk-free rates from French's library to get the excess returns. I use the characteristics in Gu, Kelly and Xiu (2020) obtained from Dacheng Xiu's personal website. The raw sample period is from July 1971 to December 2016.

Since the distributions of some characteristics can be highly skewed and leptokurtic, I remap the characteristics into the interval $(0, 1)$ for each period t. To ensure an accurate estimation of the cross-sectional expectation in the identification assumptions, a large cross-sectional sample size n is required. As a result, I have an unbalanced panel of $n = 29,778$ stocks and $T = 516$ months from January 1974 to December 2016.

4.2 Out-of-sample performance

I assess the out-of-sample model performance of the hybrid model using the total R^2 proposed in Kelly, Pruitt and Su (2019). The total R^2 evaluates how well the model explains variation in returns with contemporaneous factor realizations.

$$R^2_{total} = 1 - \frac{\sum_{(i,t)\in OOS}(y_{i,t} - \hat{b}'_{i-1,t}\hat{f}_t)^2}{\sum_{(i,t)\in OOS} y^2_{i,t}}$$

Table 1 reports the out-of-sample total R^2 for individual stocks. The number of factors varies from 1 to 6 and the number of hidden layers varies from 0 to 3. Setting the model with 0 hidden layer means the input variables will not receive non-linear transformations as the non-linear activation functions are removed from the model. Therefore, neural networks with no hidden layers is equivalent to assuming betas to be a linear specification of firm characteristics (see Figure 7).

[Insert Table 1 about here]

Table 1 shows that the out-of-sample total R^2 in general increases with both the number of factors and the number of layers. The fact that models with hidden layers outperforms the one without strongly suggest that betas are not linear in characteristics. Moreover, the hybrid model outperforms both the conditional autoencoder model by Gu, Kelly, and Xiu (2021) as well as the IPCA model by Kelly, Pruitt, and Su (2019). This result gives strong support to the value of combining econometrics and deep learning methods.

4.3 The number of conditional factors

Determining the number of factors has been an important topic in the study of factor models. To understand the dimension of the conditional latent factor space, I use cumulative explanatory power (CEP) ratios[9] as a proxy to measure how many factors are needed to explain a certain level of total variation in the whole factor space. The CEP ratio for the jth factor is defined as:

$$\rho_{j,t} = \sum_{i=1}^{j} \frac{V[f_{j,t}]}{\sum_{i=1}^{j} V[f_{j,t}]}, \quad \text{for } j = 1, 2, ..., k$$

As condition (ii) in Normalisation Restriction requires that the second moment of factors is a diagonal matrix with distinct entries in descending order, it ensures that the variation generated by factors decreases with the factor index. Figure 8 displays the time series of accumulative explanatory power ratios for different number of factors, using variance values with a 12-month rolling window. Grey vertical bars represent economic crises as from NBER and financial crises as in Zaffaroni (2019). It is easy to see that the first factor explains most of the factor space – on average over 80% of total variation. The second factor explains around 15%, showing that a

[9] The measure is inspired by the accumulative explanatory power ratio in Gagliardini and Ma (2019).

2-factor model can explain 95% of the factor space. In other words, there is one conditional factor if we aim at an AEP level of 80%, two factors at AEP level of 95%, and three factors for 99%.

[Insert Figure 8 about here]

4.4 Characteristics that matter

I also investigate into the importance of characteristics for the beta networks in Figure 10. Specifically, I set all values of a given characteristic in the beta network to zero while keeping other characteristics unchanged, and then measure the reduction in the total variation in the beta values.

[Insert Figure 10 about here]

Figure 10a reports the top 10 most important characteristics in driving betas associated with the first factor. Not surprisingly, the unconditional beta values is the most important driving force. Then the turnover and the standard deviation of turnover comes the second and third, with 1-month momentum and 1-period lags of market value follow next. Overall, the characteristics in the first beta makes the first factor look like some "Market+Turnover" factor. Figure 10b reports the characteristics ranking for the second beta, which is shown to behave like a "Volatility+Momentum" factor. Apart from the similar characteristics from the first two betas, the third beta loads additionally on sales, dividend yield, and cash to debt ratio as shown in Figure 10c.

4.5 Linearity of betas

To further test whether betas take a linear specification, I resort to Post-Lasso estimation method considering the high-dimensionality issue here. The method consists of two steps. First, I use a standard Lasso with l_1 penalty to run a panel regression of beta estimates on the firm characteristics for dimension reduction. As a dimension reduction tool, Lasso will drop variables that have negligible explanatory power. I take the beta estimates from the model with 6 factors and 1 hidden layer to run the following Lasso regression:

$$\hat{b}^j_{i,t} = \alpha^j + w'_{i,t}\gamma^j + \varepsilon_{i,t}, \qquad \text{for } j = 1, ..., k, \quad i = 1, ..., n, \quad t = 1, ...T,$$

and its objective function:

$$\min_{\alpha^j,\gamma^j} \sum_{t=1}^{T} \|(\hat{b}^j_{i,t} - \alpha^j - w'_{i,t}\gamma^j)\|^2 \text{ subject to } \sum_{j=1}^{k} |\gamma^j| \leq c,$$

where α, γ are constant and coefficient vector, and c is a hyper-parameter that determines the degree of regularization.

With the non-negligible variables selected by Lasso, I then run an OLS regression:

$$\hat{b}^j_{i,t} = \alpha^j + \tilde{w}^{l'}_{i,t}\gamma^j + \varepsilon_{i,t}, \qquad \text{for } j = 1, ..., k, \quad i = 1, ..., n, \quad t = 1, ...T,$$

where $\tilde{w}$ represents the selected firm characteristics.

As shown above, Post-Lasso estimator is good at both dimension reduction and statistical inference. The characteristics are normalised so that all have the same distribution. The absolute values of the coefficients are therefore equivalent to their explanatory power in the regression. In Figure 9, I report the explanatory power ranking of characteristics as well as the adjusted R^2 when $k = 1$.

[Insert Figure 9 about here]

As the first factor explains over 80% variation in the factor space, I run the regression when $k = 1$. Based on the Post-Lasso regression results, 79 out of 94 firm characteristics have significant coefficients but a linear specification only explains an R^2 of 44.96% for the first conditional factor. The result suggests that the first beta is highly non-linear in most pre-defined characteristics from the literature.

4.6 Understanding the factors

As stated in section 2, condition (ii) in Normalisation Restriction is equivalent to defining the factors as conditional principal components. Therefore, the factor values do not admit a direct economic interpretation and must be analyzed in a conditional setting. To understand what financial indicators span the conditional factor space, I adopt the conditional canonical correlation method with machine learning techniques proposed in Gagliardini and Ma (2020).

Let us consider a vector Z_t of K^O financial indicators. The conditional canonical correlations between these vectors are denoted by $\rho_{r,t}$, $r = 1, 2, \cdots, \underline{K}$, where $\underline{K} = \min\{K^O, k\}$. Specifically, the first conditional canonical correlation is defined by:

$$\begin{aligned} \rho_{1,t} &= \max_{c_1,\ d_1} Cov(c_1' Z_t, d_1' f_t | Z_{t-1}) \\ &\quad \text{s.t. } V(c_1' Z_t | Z_{t-1}) = 1, \quad V(d_1' f_t | Z_{t-1}) = 1. \end{aligned}$$

The second, third, etc. conditional canonical correlations are defined recursively (see appendix).

Figure 11 reports the conditional canonical correlation between factor estimate vector $\hat{f}_t$ with dimension $k = 6$, and financial indicator vector Z_t with dimension $K^O = 13$. In this case, I use 8 indicators from Goyal and Welch (2008) [10]. Each curve in the plot represents one time series of conditional correlation. Grey vertical bars represent economic crises as from NBER and financial crises as in Zaffaroni (2019).

From the figure, I find that the financial indicators span well the first and second factors, suggesting that economists have done a good job in extracting common risk sources in normal times. Nonetheless, the estimated common risks seem not to correlate well with the financial indicators during financial crisis, indicating that state variables should be included to drive the dynamics of time-varying betas.

[Insert Figure 11 about here]

4.7 Literature-based characteristics vs. COMPUSTAT raw features

In this section, I look into the Man vs. Machine topic from the researcher's angle. Specifically, I want to understand whether human research can bring advantage even if neural networks in theory can approximate any functions given the unprocessed characteristics. I compare the out-of-sample performance of the hybrid models with different versions of firm characteristics: 1) literature-based characteristics (e.g., book-to-market ratio) and 2) COMPUSTAT raw features (e.g., book value and market value). Table 2 shows the results.

[Insert Table 2 about here]

[10] The 8 indicators from Goyal and Welch (2008) include Dividend-Price ratio, Earnings-Price ratio, Book-to-Market ratio, Net Equity Expansion, Treasury Bills, Term Spread, Default Yield Spread and Stock Variance.

The results show that models with literature-based characteristics generally outperforms the one with COMPUSTAT raw features, indicating the advantage that human research brings in selecting better specifications for betas.

5 Conclusion

I develop a hybrid methodology that incorporates an econometric identification strategy into neural networks when studying conditional latent factor models. The time-varying betas are high-dimensional unknown functions of firm characteristics. The statistical factors are population cross-sectional OLS estimators for given beta values. Under very general econometric assumptions, I prove that identifying betas and factors boils down to identifying only the function of betas and can be written as a constrained optimization problem. I then construct neural networks equipped with the objective function and identification constraints from the identification process. As a result, such econometrics-based neural networks will give non-parametric estimates of functions of betas and ensure tradeable factors constructible by portfolio returns. Compared with benchmark econometric models, the hybrid model has multiple advantages such as model-free assumptions for functions and capability of dealing with high-dimensional datasets. Compared with benchmark neural networks methods, the econometrics-based neural networks benefit from econometric foundation in terms of tradeability of statistical factors and improvement in computational efficiency.

Empirically, I conduct the analysis on an unbalanced panel of monthly data of the US individual stocks with $30,000$ firms, 516 months and 94 characteristics. First, the hybrid method outperforms benchmark econometric methods, the IPCA method by Kelly, Pruitt, and Su (2019), and neural networks methods, the conditional autoencoder neural networks methods in Gu, Kelly, and Xiu (2021) when explaining out-of-sample individual returns. Second, I find that betas are highly non-linear in firm characteristics as linear regressions of betas on characteristics can only yield an adjusted-R^2 less than 45%. Third, I determine the number of conditional factors based on how much variation they account for with the cumulative explanatory power ratio. I show that 1 factor is able to explain over 80% variation in the factor space, and 2 factors will increase the percentage to 95%. These results indicates that the dimension of the conditional space is smaller than that of the unconditional space, and that some variation in unconditional factor models may result from variation in time-varying betas. Fourth, the beta associated with the first factor, which explains over 80% of the variation in factor space, is mostly driven by unconditional betas and turnover. The second beta is mostly explained by volatility and momentum. Furthermore, I use conditional canonical correlation to measure how well financial market indicators can span the conditional latent factors. Results show that they coincide well when the market is in good condition, but comparably worse during financial distresses. This suggests that macro variables should be included as driving forces for time-varying betas. Lastly, I compare the out-of-sample performance of the hybrid models across different firm characteristics, which are literature-based characteristics (e.g., book-to-market ratio) and COMPUSTAT raw features (e.g., book value and market value). The results show that models with literature-based characteristics generally outperforms the one with COMPUSTAT raw features, indicating the advantage that human research brings in selecting the better specification.

References

Ait-Sahalia, Y., and Xiu, D. (2017). Using Principal Component Analysis to Estimate a High Dimensional Factor Model with High-Frequency Data. *Journal of Econometrics*, 201, 384-399.

Andrews, D. W. (2005). Cross-section Regression with Common Shocks. *Econometrica*, 73(5), 1551-1585.

Belloni, A., Chen, D., Chernozhukov, V., and Hansen, C. (2012). Sparse Models and Methods for Optimal Instruments with an Application to Eminent Domain. *Econometrica*, 80(6), 2369-2429.

Belloni, A., & Chernozhukov, V. (2013). Least squares after model selection in high-dimensional sparse models. *Bernoulli*, 19(2), 521-547.

Chamberlain, G., and Rothschild, M. (1983). Arbitrage, Factor Structure, and Mean-Variance Analysis on Large Asset Markets. *Econometrica*, 51 (5), 1281-1304.

Chen, L., Pelger, M., & Zhu, J. (2019). Deep learning in asset pricing. *Working Paper*

Cochrane, J. H. (2011). Presidential Address: Discount Rates. *Journal of Finance*, 66(4), 1047-1108.

Connor, G., Hagmann, M. and Linton, O. (2012): Efficient Semiparametric Estimation of the Fama-French Model and Extensions, *Econometrica*, 80, 713-754.

Fama, E. F., and French, K. R. (1993). Common Risk Factors in the Returns on Stocks and Bonds. *Journal of Financial Economics*, 33(1), 3-56.

Fan, J., Liao, Y., and Wang, W. (2016): Projected Principal Component Analysis in Factor Models, *Annals of Statistics*, 44, 219-254.

Feng, G., Giglio, S., and Xiu, D. (2017). Taming the Factor Zoo: A Test of New Factors. *Journal of Finance.*

Ferson, W. E., and Harvey, C. R. (1991). The Variation of Economic Risk Premiums. *Journal of Political Economy*, 99(2), 385-415.

Ferson, W. E., and Harvey, C. R. (1999). Conditioning Variables and the Cross Section of Stock Returns. *The Journal of Finance*, 54(4), 1325-1360.

Freyberger, J., Neuhierl, A., and Weber, M. (2017). Dissecting Characteristics Nonparametrically, Working Paper. *National Bureau of Economic Research.*

Gagliardini, P., and Gourieroux, C. (2017). Double Instrumental Variable Estimation of Interaction Models with Big Data. *Journal of Econometrics*, 201(2), 176-197.

Gagliardini, P., and Ma, H. (2020). Extracting Statistical Factors When Betas Are Time-Varying. *Working Paper*

Gagliardini, P., Ossola, E., and Scaillet, O. (2016). Time-Varying Risk Premium in Large Cross-Sectional Equity Data Sets. *Econometrica*, 84(3), 985-1046.

Gagliardini, P., Ossola, E., and Scaillet, O. (2019). A Diagnostic Criterion for Approximate Factor Structure. *Journal of Econometrics.*

Gallant, A. R., and White, H. (1988). There Exists a Neural Network that Does not Make Avoidable Mistakes. In *IEEE Second International Conference on Neural Networks*, I, 657-664.

Giglio, S., and Xiu, D. (2017). Asset Pricing with Omitted Factors. *Journal of Political Economy.*

Goyal, A. and Welch, I. (2007). A Comprehensive Look at the Empirical Performance of Equity Premium Prediction. *The Review of Financial Studies*, 21(4), 1455-1508.

Gu, S., Kelly, B. T., and Xiu, D. (2020). Empirical Asset Pricing via Machine Learning. *Review of Financial Studies.*

Gu, S., Kelly, B. T., and Xiu, D. (2021). Autoencoder Asset Pricing Models. *Journal of Econometrics.*

Ghysels, E. (1998). On stable factor structures in the pricing of risk: do time-varying betas help or hurt?. *The Journal of Finance*, 53(2), 549-573.

Haerdle, W., and Linton, O. (1994). Applied Nonparametric Methods. In *Handbook of Econometrics*, R. Engle and D. McFadden eds., Volume 4, Chapter 38, 2295-2339.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, 2, 359-366.

Hornik, K. (1991). Approximation Capabilities of Multilayer Feed-Forward Networks. *Neural Networks*, 4(2), 251-257.

Kelly, B., Pruitt, S., and Su, Y. (2017): Instrumented Principal Component Analysis. Working Paper.

Kelly, B., Pruitt, S., and Su, Y. (2019): Characteristics are Covariances: A Unified Model of Risk and Return, *Journal of Financial Economics.*

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Pelger, M. (2019). Large-Dimensional Factor Modeling Based on High-Frequency Observations. *Journal of Econometrics*, 4, 23-42.

Pelger, M. (2019). Understanding Systematic Risk: A High-Frequency Approach. *Journal of Finance*

Pelger, M., and Xiong, R. (2018): State-Varying Factor Models of Large Dimensions, Working Paper.

Singleton, K. (2006). Empirical Dynamic Asset Pricing. Model Specification and Econometric Assessment. Princeton University Press.

Zaffaroni, P. (2019) Factor Models for Asset Pricing. Working Paper.

Appendix 1: Proof of Proposition

Let us start from the minimization problem

$$\min_{b(\cdot),\{f_t\}} \mathbb{E}\left[(y_{i,t} - b(w_{i,t-1})'f_t)^2\right] \tag{8}$$

where minimization is w.r.t. function $b(\cdot)$ and process $\{f_t\}$ that is measurable with respect to the sigma-field of limit cross-sectional averages (namely $\mathcal{F}_t$ in Gagliardini and Ma (2020)). I use that $\mathbb{E}\left[(y_{i,t} - b(w_{i,t-1})'f_t)^2\right] = \mathbb{E}\left[\mathbb{E}^c[(y_{i,t} - b(w_{i,t-1})'f_t)^2]\right]$. Thus, the minimization with respect to process f_t yields the solution

$$f_t = \mathbb{E}^c[b(w_{i,t-1})b(w_{i,t-1})']^{-1}\mathbb{E}^c[b(w_{i,t})y_{i,t}] \tag{9}$$

For this solution,

$$\mathbb{E}^c[(y_{i,t} - b(w_{i,t-1})'f_t)^2] = \mathbb{E}^c[y_{i,t}\{y_{i,t} - b(w_{i,t-1})'\mathbb{E}^c[b(w_{i,t-1})b(w_{i,t-1})']^{-1}\mathbb{E}^c[b(w_{i,t-1})y_{i,t}]\}]$$

for the properties of projection residuals. Now, I plug in the DGP $y_{i,t} = b^0(w_{i,t-1})'f_t^0 + u_{i,t}$ and use the property of the instruments: $\mathbb{E}^c[b(w_{i,t-1})u_{i,t}] = 0$ for any function $b(\cdot)$. I get:

$$\begin{aligned}
&\mathbb{E}^c[(y_{i,t} - b(w_{i,t-1})'f_t)^2] \\
&= \mathbb{E}^c[u_{i,t}^2] + f_t^{0\prime}\left(\mathbb{E}^c[b^0(w_{i,t-1})b^0(w_{i,t-1})']\right. \\
&\quad \left. -\mathbb{E}^c[b^0(w_{i,t-1})b(w_{i,t-1})']\mathbb{E}^c[b(w_{i,t-1})b(w_{i,t-1})']^{-1}\mathbb{E}^c[b(w_{i,t-1})b^0(w_{i,t-1})']\right) f_t^0.
\end{aligned}$$

Then, the minimization problem (8) becomes

$$\min_{b(\cdot)} \sigma^2 + \mathbb{E}\left[f_t^{0\prime}\left(\mathbb{E}^c[b^0b^{0\prime}] - \mathbb{E}^c[b^0b']\mathbb{E}^c[bb']^{-1}\mathbb{E}^c[bb^{0\prime}]\right)f_t^0\right] \tag{10}$$

where σ^2 is a constant independent of $b(\cdot)$, and I use $\mathbb{E}^c[bb'] \equiv \mathbb{E}^c[b(w_{i,t-1})b(w_{i,t-1})']$ to ease notation. Now, $\mathbb{E}^c[b^0b^{0\prime}] - \mathbb{E}^c[b^0b']\mathbb{E}^c[bb']^{-1}\mathbb{E}^c[bb^{0\prime}]$ is the cross-sectional second moment of the cross-sectional regression of $b^0(w_{i,t})$ onto $b(w_{i,t-1})$. Thus, the minimum in (10) is achieved uniquely when

$$b(\cdot) = Ab^0(\cdot), \tag{11}$$

where A is a non-singular matrix, that is individual and time independent. By plugging into (9) and using again $y_{i,t} = b^0(w_{i,t-1})' f_t^0 + u_{i,t}$, the corresponding solution for the factor is

$$f_t = \mathbb{E}^c[bb']^{-1}\mathbb{E}^c[b(b^0)']f_t^0 = (A[b^0(b^0)']A')^{-1}A\mathbb{E}^c[b^0(b^0)']f_t^0 = (A')^{-1}f_t^0.$$

To identify matrix A, I use the normalization restrictions. From $E[bb'] = I_k$ I get $I_k = AE[b^0(b^0)']A' = AA'$, i.e. matrix A is orthogonal. Further, from the condition that the unconditional variance-covariance matrix of the factors is diagonal with decreasing elements, I get that both

$$(A')^{-1}E[f_t^0(f_t^0)']A^{-1}$$

and

$$E[f_t^0(f_t^0)']$$

are diagonal, with ranked diagonal elements. If the eigenvalues of $E[f_t^0(f_t^0)']$ are distinct, it follows that $A = I_k$.

Appendix 2: Keras functional API

Keras is an open-source software library written in Python for artificial neural networks, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation so that researchers can implement their ideas with very low learning cost.

Keras offers two types of models to build a deep learning architecture. The simpler type is called the Sequential model, which is a linear stack of layers. The other type is called the Keras functional API, which allows to build arbitrary graphs of layers with non-linear topology, shared layers, and even multiple inputs or outputs. I use the Keras functional API because my model is too complex to be implemented with the Sequential model.

With the functional API, I am able to carry out math operations on the tensors by calling different classes of layers. A tensor is generalization of matrices of any dimensions, which can be a scaler, a vector, a matrix or an array of higher dimensions. For example, I can apply matrix addition/subtraction by calling *add/subtraction layers*, matrix multiplication by calling *dot layers*, and even a custom function using *lambda layers*. This makes it possible for us to find a one-to-one mapping from the econometric conditions to the deep learning modules.

Lastly, the optimization algorithm can always find us a solution for a deep learning architecture equipped with a valid loss function as long as all modules are well connected, no matter how complicated the model is.

Appendix 3: Conditional Canonical Correlation

The second, third, etc. conditional canonical correlations are defined recursively by:

$$\begin{aligned}
\rho_{r,t} \quad &= \quad \max_{c_r,\ d_r} \ Cov(c_r' Z_t, d_r' f_t | Z_{t-1}) \\
&\quad \text{s.t.} \ \ V(c_r' Z_t | Z_{t-1}) = 1, \quad V(d_r' f_t | Z_{t-1}) = 1, \\
&\qquad\quad Cov(c_r' Z_t, c_j' Z_t | Z_{t-1}) = 0, \quad Cov(d_r' f_t, d_j' f_t | Z_{t-1}) = 0, \quad j = 1, \cdots, r-1,
\end{aligned}$$

for $r = 2, ..., \underline{K}$.

By analogy with the unconditional setting (see e.g. Anderson (2003)), the squared conditional canonical correlations are the $\underline{K}$ largest eigenvalues of matrix

$$\mathcal{R}_{t-1} = V(Z_t|Z_{t-1})^{-1} Cov(Z_t, f_t|Z_{t-1}) V(f_t|Z_{t-1})^{-1} Cov(f_t, Z_t|Z_{t-1}),$$

where conditional variance and covariance $V(\cdot|Z_{t-1})$ and $Cov(\cdot,\cdot|Z_{t-1})$ are estimated based on machine learning methods developed in Gagliardini and Ma (2020).

Appendix 4: Figures and Tables

Figure 1. The colliding trends of econometrics and deep learning

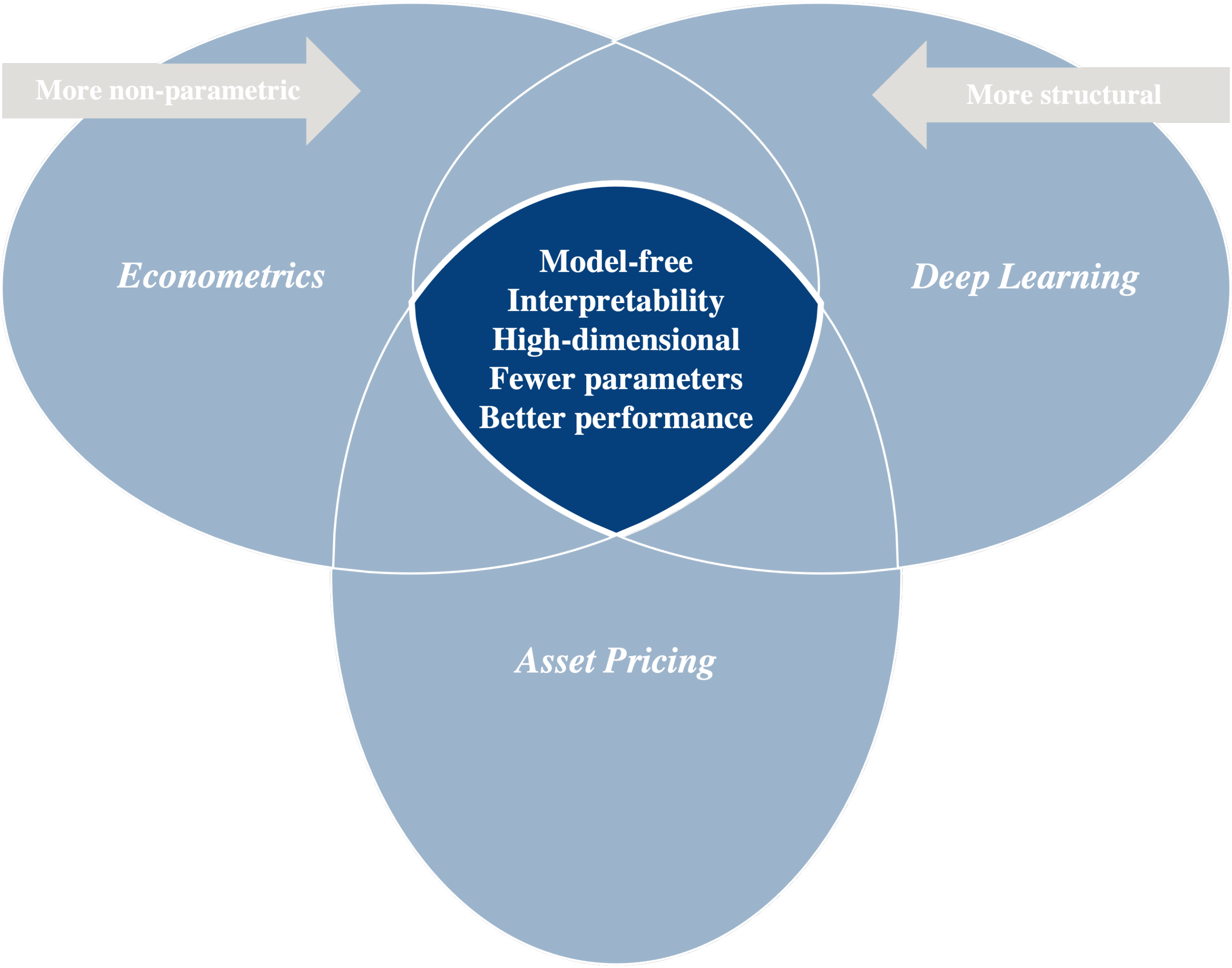
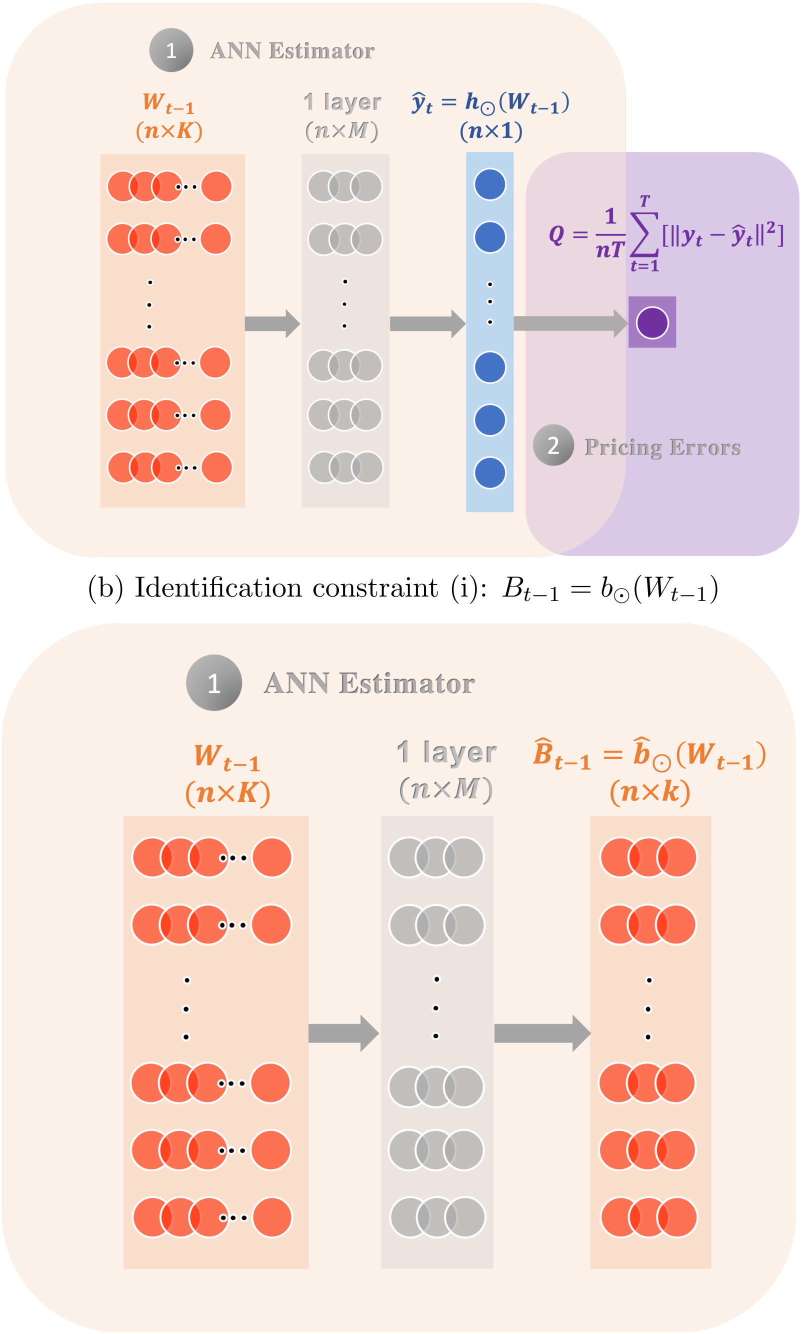


Figure 2. Diagram of standard neural networks


1 ANN Estimator
W_{t-1} $(n\times K)$
1 layer $(n\times M)$
$\hat{y}_t = h_{\odot}(W_{t-1})$ $(n\times 1)$
$Q = \frac{1}{nT}\sum_{t=1}^{T}[\|y_t - \hat{y}_t\|^2]$
2 Pricing Errors
1 ANN Estimator
W_{t-1} $(n\times K)$
1 layer $(n\times M)$
$\hat{B}_{t-1} = \hat{b}_{\odot}(W_{t-1})$ $(n\times k)$

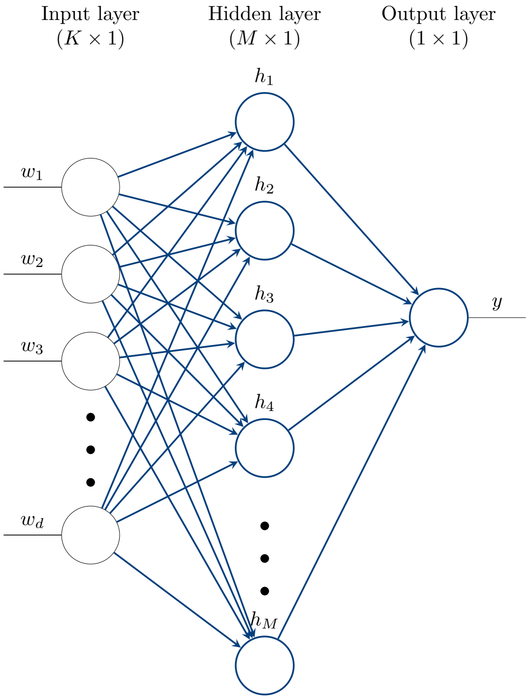
(a) Model (5): $y_t = h_{\odot}(W_{t-1}) + \varepsilon_t$

(b) Identification constraint (i): $B_{t-1} = b_{\odot}(W_{t-1})$

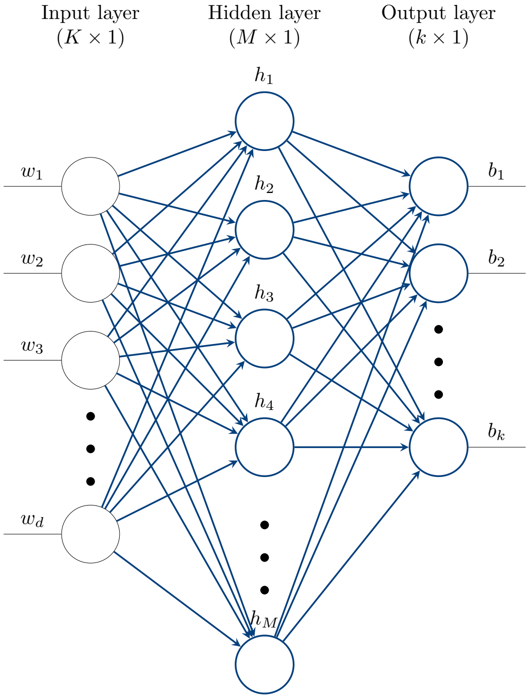
W_{t-1} is the $n \times K$ matrix of characteristics, y_t is the $n \times 1$ vector of individual stock excess returns, and $b_{\odot}(\cdot)$ is the asset-wise operation of W_{t-1} with unknown function $b(\cdot)$.

Figure 3. Number of parameters for the neural networks displayed in Figure 2

(a) Model (5): $(K+1) \times M + M + 1$

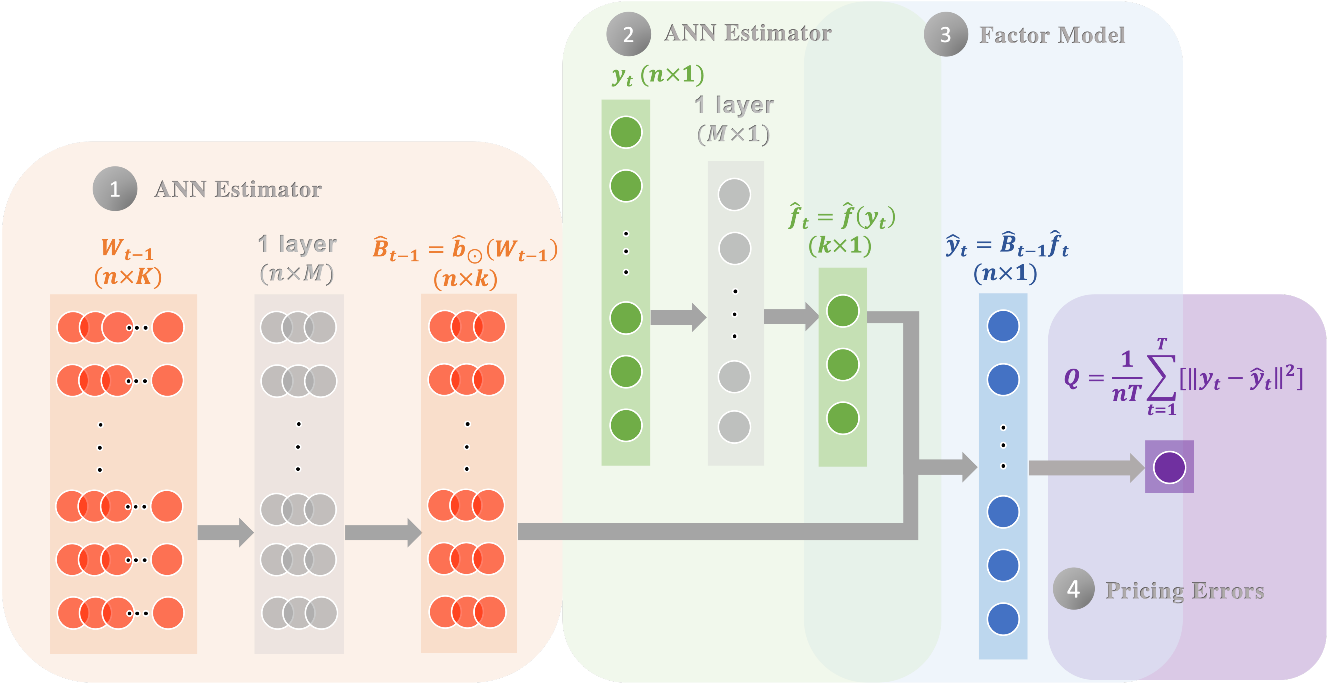


(b) Model (6): $(K+1) \times M + (M+1) \times k$



Operations involving parameters are painted in dark blue. The number of parameters are obtained by counting the arrows and circles in dark blue.

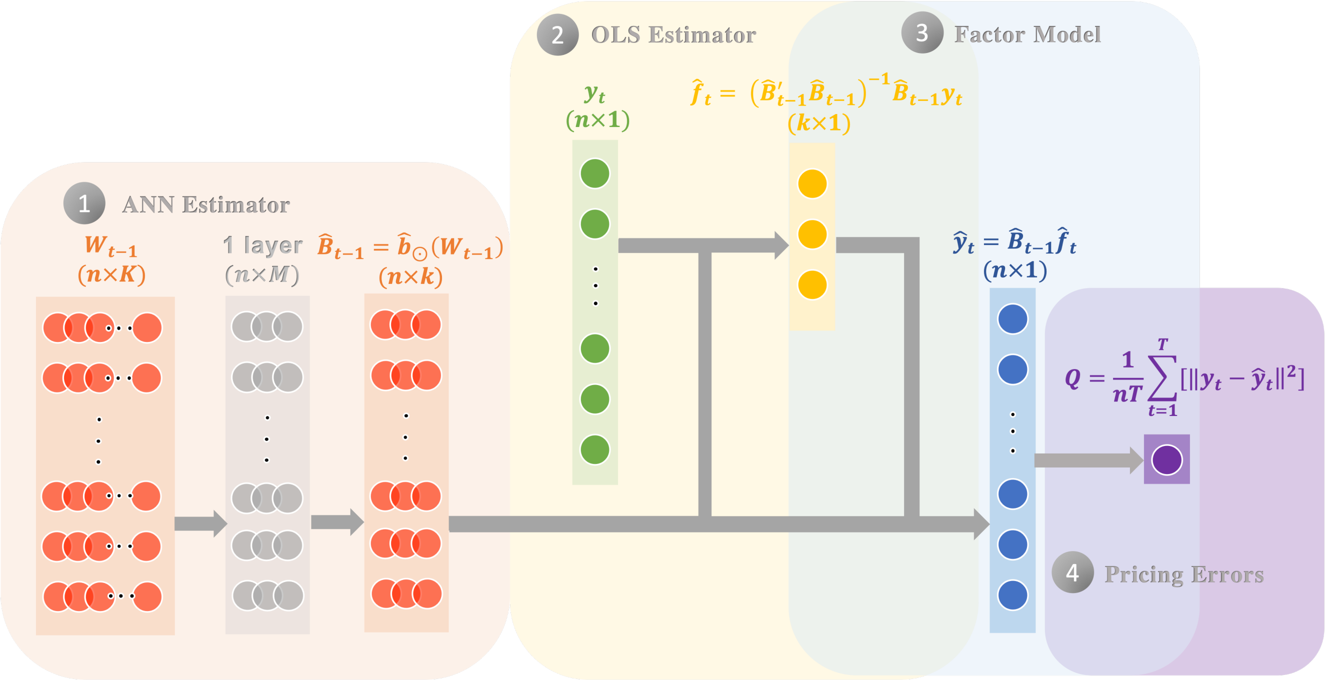
Figure 4. Diagram of autoencoder asset pricing model (7)


1 ANN Estimator
W_{t-1} $(n\times K)$
1 layer $(n\times M)$
$\hat{B}_{t-1} = \hat{b}_{\odot}(W_{t-1})$ $(n\times k)$
2 ANN Estimator
y_t $(n\times 1)$
1 layer $(M\times 1)$
$\hat{f}_t = \hat{f}(y_t)$ $(k\times 1)$
3 Factor Model
$\hat{y}_t = \hat{B}_{t-1}\hat{f}_t$ $(n\times 1)$
$Q = \frac{1}{nT}\sum_{t=1}^{T}[\|y_t - \hat{y}_t\|^2]$
4 Pricing Errors

$$
\begin{aligned}
&y_t = B_{t-1} f_t + u_t, \\
&s.t. \begin{cases} B_{t-1} = b_{\odot}(W_{t-1}) \\ f_t = f(y_t) \end{cases}
\end{aligned}
$$

W_{t-1} is the $n \times K$ matrix of characteristics, y_t is the $n \times 1$ vector of individual stock excess returns, $b_{\odot}(\cdot)$ is the asset-wise operation of W_{t-1} with unknown function $b(\cdot)$ for modeling betas, and $f(\cdot)$ is the unknown functional form of latent factors.

Figure 5. Diagram of my econometric-based neural networks (4)



1 ANN Estimator
W_{t-1}
$(n\times K)$
1 layer
$(n\times M)$
$\hat{B}_{t-1} = \hat{b}_{\odot}(W_{t-1})$
$(n\times k)$
2 OLS Estimator
y_t
$(n\times 1)$
$\hat{f}_t = (\hat{B}'_{t-1}\hat{B}_{t-1})^{-1}\hat{B}_{t-1}y_t$
$(k\times 1)$
3 Factor Model
$\hat{y}_t = \hat{B}_{t-1}\hat{f}_t$
$(n\times 1)$
$Q = \frac{1}{nT}\sum_{t=1}^{T}[\|y_t - \hat{y}_t\|^2]$
4 Pricing Errors

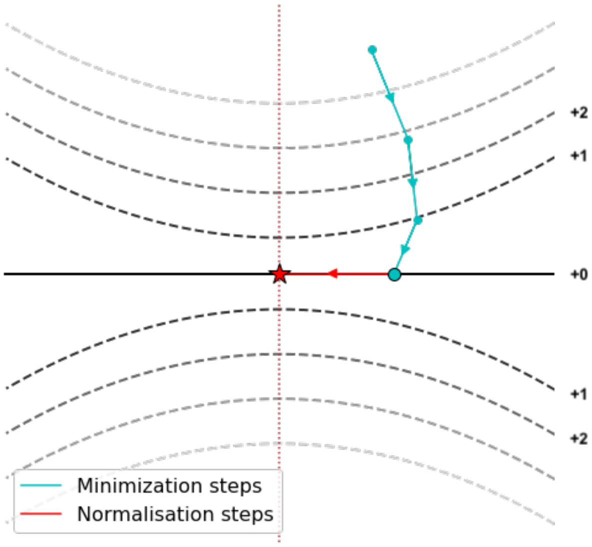
$$\min_{b(\cdot)} \frac{1}{nT}\sum_{t=1}^{T}(y_t - B_{t-1}\hat{f}_t)'(y_t - B_{t-1}\hat{f}_t)$$

$$\text{s.t.} \begin{cases} B_{t-1} = b_{\odot}(W_{t-1}) \\ \hat{f}_t = (B'_{t-1}B_{t-1})^{-1}B'_{t-1}y_t, \\ \dfrac{1}{nT}\displaystyle\sum_{t=1}^{T} B'_{t-1}B_{t-1} = I_k, \\ \dfrac{1}{T}\displaystyle\sum_{t=1}^{T} \hat{f}_t\hat{f}'_t \text{ is diagonal, with distinct} \\ \text{entries ranked in descending order,} \\ \dfrac{1}{T}\displaystyle\sum_{t=1}^{T} \hat{f}_t \geq 0 \end{cases}$$

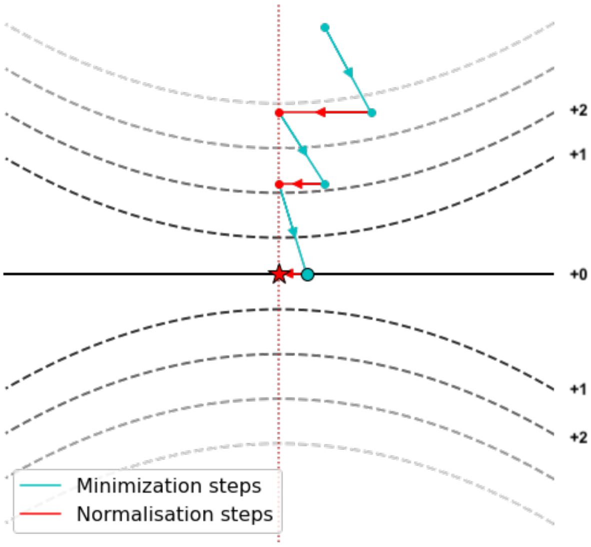
W_{t-1} is the $n \times K$ matrix of characteristics, y_t is the $n \times 1$ vector of individual stock excess returns, and $b_{\odot}(\cdot)$ is the asset-wise operation of W_{t-1} with unknown function $b(\cdot)$ for modeling betas.

Figure 6. Two normalisation strategies

(a) Normalisation after minimization



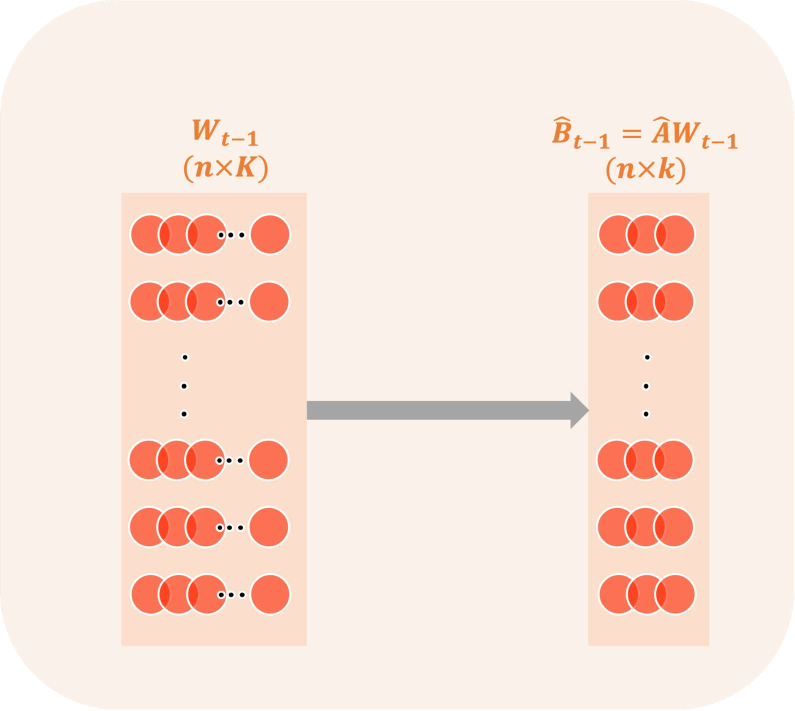
(b) Normalisation during minimization



Plots above show two different normalisation strategies. One is to normalise only after the algorithm stops. The other is to normalise at each iteration when estimates are updated. In theory, the two strategies are equivalent and should give the same estimates.

The curves and line in black represent the contours of loss function values, where the horizontal line represents the minimum loss and also indicates an infinite number of equivalent solutions. The dotted vertical line represents the normalisation restriction. The star indicates the unique solution under normalisation restruction, and each circle indicates an updated estimate.

Figure 7. Diagram of neural networks with 0 hidden layers



The neural networks with 0 hidden layers is equivalent to assuming the beta vector B_{t-1} is a linear transformation of firm characteristics matrix W_{t-1}: $B_{t-1} = AW_{t-1}$.

Figure 8. Cumulative Explanatory Power Ratio (CEP)

This figure displays the time series of cumulative explanatory power (CEP) ratios for different number of factors, using variance values with a 12-month rolling window, namely $\rho_{j,t} = \sum_{i=1}^{j} \frac{V[f_{j,t}]}{\sum_{i=1}^{k} V[f_{j,t}]}$ with $j = 1, ..., k$ and $k = 6$. The ratio is equal to 1 when $j = k$ as it measures the total variation of all factors. Grey vertical bars represent economic crises as from NBER and financial crises as in Zaffaroni (2019).

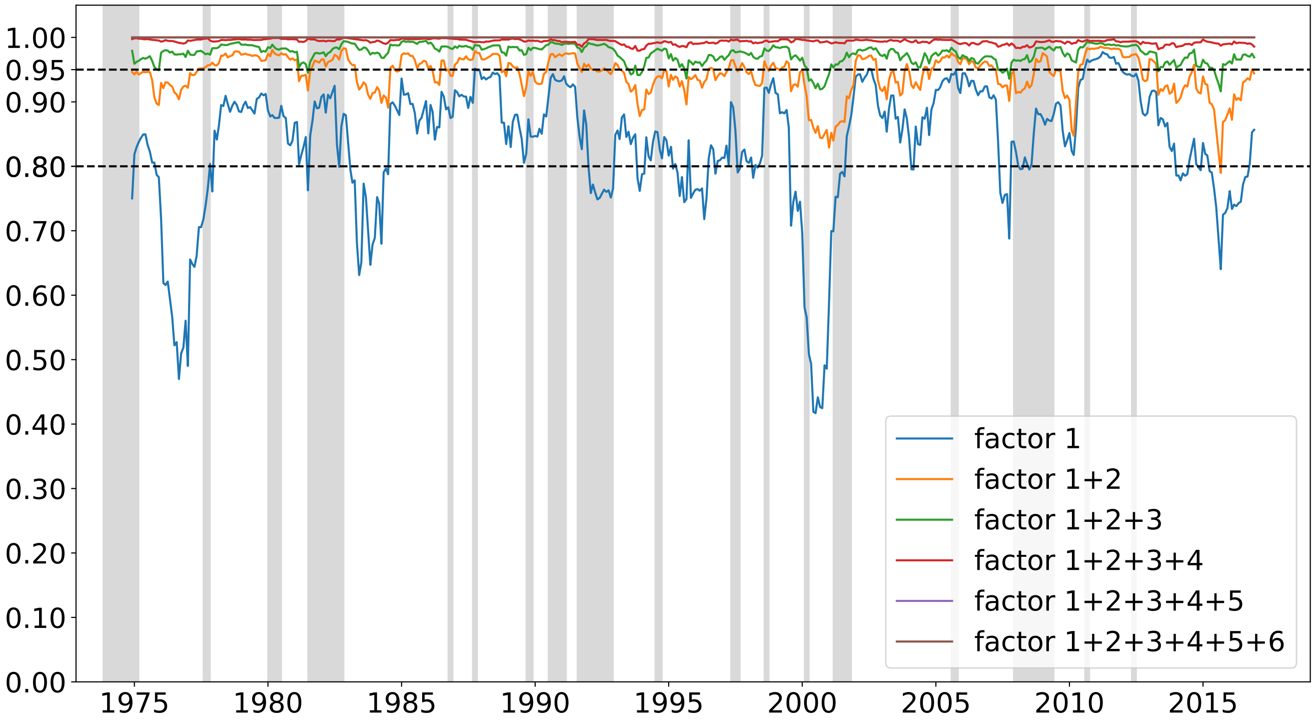


Figure 9. Linearity of $\hat{b}^1_{i,t}$

$$\hat{b}^1_{i,t} = \alpha^1 + w'_{i,t}\gamma^1 + \varepsilon_t$$

As firm characteristics $w_{i,t}$ are all normalised with same distribution, their coefficient ranking is equivalent to their explanatory power ranking for the beta estimates $\hat{b}_{i,t}$. Coefficients equal to zero are found not significant at 1% level. The adjusted R^2 of above model is 44.96%

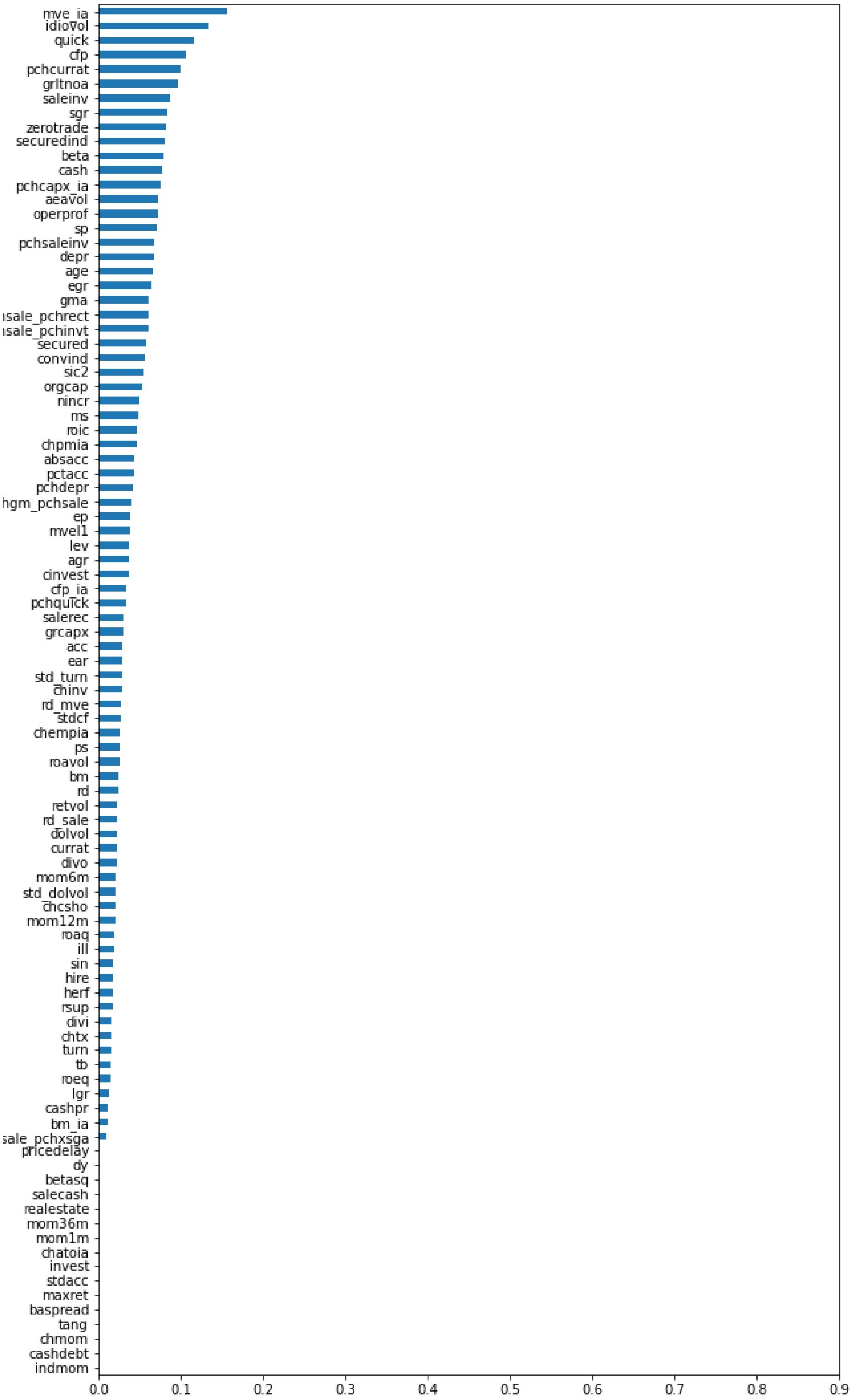
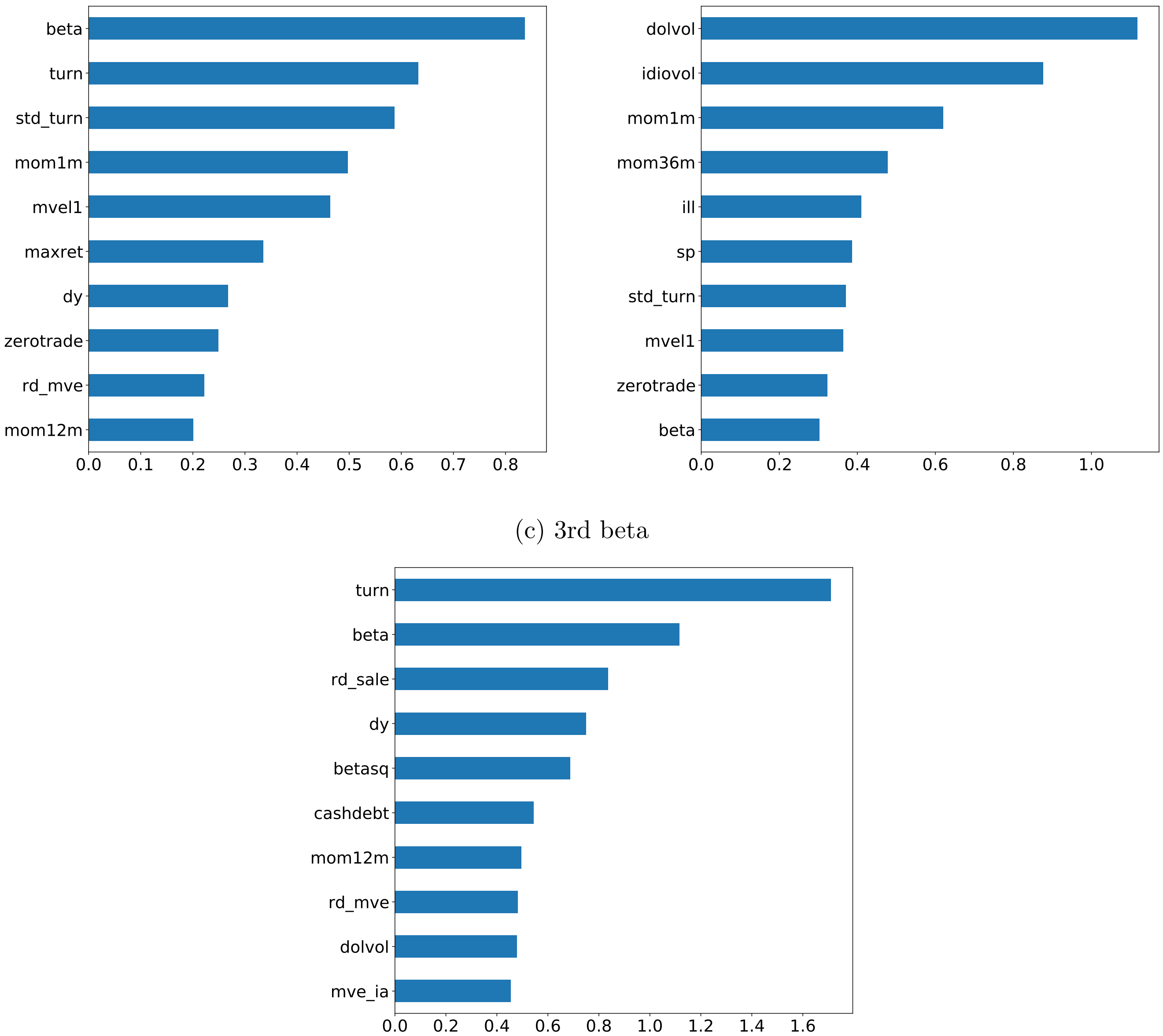


Figure 10. Importance ranking of characteristics

This figure reports the top 10 stock-level characteristics in terms of overall model contribution for the three betas associated with the first three factors.



(a) 1st beta

(b) 2nd beta

(c) 3rd beta

Figure 11. CCC between conditional latent factor estimates and financial indicators

Figure below reports the conditional canonical correlation (CCC) between the conditional latent factor estimate vector $\hat{f}_t$ with dimension $k = 6$, and the financial indicator vector Z_t with dimension $K = 13$, including 8 Goyal and Welch indicators and Fama French 5 factors. Plot shows that (1) these financial indicators span well the first factor but (2) does not depict the common risks during financial distresses. Grey vertical bars represent economic crises as from NBER and financial crises as in Zaffaroni (2019).

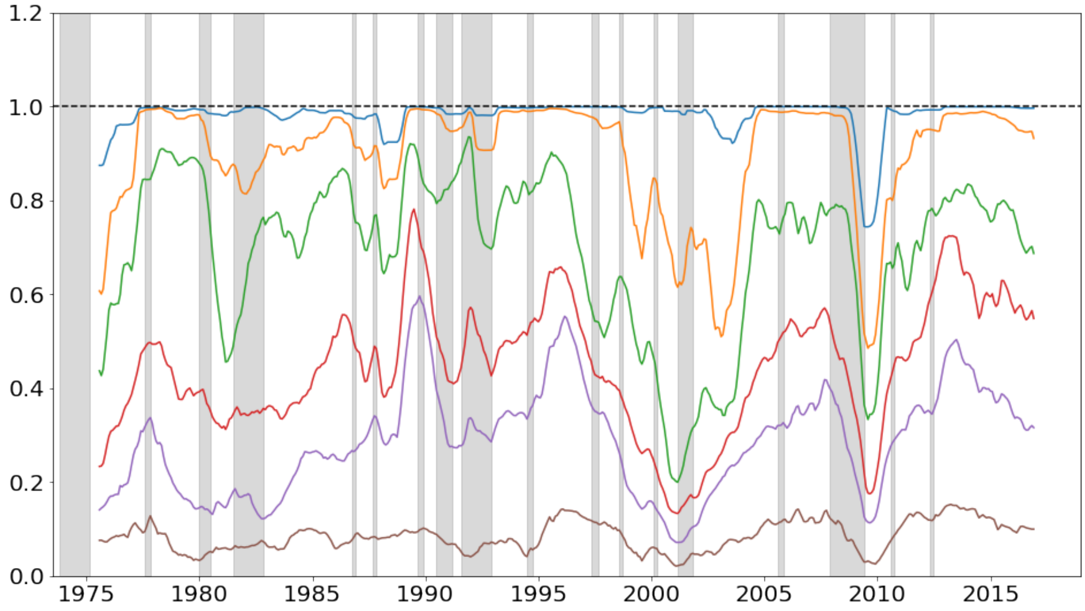


Table 1
Out-of-sample R^2_{total}: model comparison

This table reports the out-of-sample total R^2 for individual stocks across different models. The number of factors varies from 1 to 6. The number of hidden layers varies from 0 to 3. A model with no hidden layer is equivalent to assuming a linear specification for betas.
1) **AP+ECMX+ANN** represents my hybrid model. The number of stocks used is around $30,000$.
2) **AP+ANN** represents the conditional autoencoder estimation method in Gu, Kelly, and Xiu (2021). The number of stocks used is around $30,000$.
3) **AP+ECMX** represents the IPCA method by Kelly, Pruitt, and Su (2019). The number of stocks used is around $12,000$.

		Number of factors					
Model	**Number of layers**	**1**	**2**	**3**	**4**	**5**	**6**
AP+ECMX+ANN	**0**	9.62%	11.36%	12.01%	12.92%	13.77%	14.08%
	1	10.85%	13.03%	14.62%	14.89%	15.90%	15.71%
	2	**11.71%**	**13.89%**	**14.69%**	**15.65%**	**16.37%**	**16.85%**
	3	10.12%	13.76%	14.68%	15.65%	16.19%	16.50%
AP+ANN	**0**	**10.90%**	**11.80%**	12.30%	12.20%	12.50%	12.40%
	1	10.40%	11.50%	12.20%	12.90%	13.40%	**14.30%**
	2	10.70%	11.80%	**12.60%**	13.20%	13.60%	13.80%
	3	10.70%	11.80%	12.50%	**13.30%**	**13.70%**	13.80%
AP+ECMX		**11.20%**	**12.40%**	**13.30%**	**13.70%**	**14.30%**	**14.50%**

Table 2
Out-of-sample R^2_{total}: Literature-Based Features vs. COMPUSTAT Raw Features

This table reports the out-of-sample total R^2 for individual stocks using my model with different firm characteristics $w_{i,t}$. The number of factors varies from 1 to 6 and the number of hidden layers varies from 0 to 3. A model with no hidden layer is equivalent to assuming a linear specification for betas.
The literature-based characteristics are from Gu, Kelly, and Xiu (2020) (e.g., book-to-market ratio). COMPUSTAT raw features are directly extracted from COMPUSTAT (e.g., book value and market value). The number of stocks is around $1,100$ for computational efficiency.

		Number of factors					
Model	**Number of layers**	**1**	**2**	**3**	**4**	**5**	**6**
Literature-based Characteristics	**0**	19.01%	20.94%	22.34%	**23.25%**	23.96%	24.49%
	1	20.09%	21.32%	22.38%	23.09%	23.74%	24.40%
	2	**20.13%**	**21.43%**	**22.84%**	23.22%	**24.29%**	**24.67%**
	3	15.83%	21.00%	21.44%	22.21%	22.42%	23.53%
COMPUSTAT Raw Features	**0**	18.99%	20.33%	**21.26%**	**22.32%**	**23.02%**	**23.62%**
	1	**19.23%**	**20.50%**	20.43%	21.98%	22.59%	22.19%
	2	19.12%	19.80%	20.64%	21.55%	22.33%	22.29%
	3	19.09%	19.67%	20.80%	21.14%	21.77%	21.87%