Deep Learning with Non-Linear Factor Models: Adaptability and Avoidance of Curse of Dimensionality

Mehmet Caner[*] Maurizio Daniele[†]

September 12, 2022

Abstract

In this paper, we connect the deep learning literature with non-linear factor models and show that deep learning estimation leads to a substantial improvement in the non-linear factor model literature. We provide bounds on the expected risk and prove that these upper bounds are uniform over a set of multiple response variables. We extend our results to an additive model setting and show its connection to non-linear factor models for financial applications. Compared to traditional factor models which assume rigid linear relations between the factors and the underlying observed variables, our deep neural network factor model (DNN-FM) offers major improvements in modeling flexibility. Moreover, we provide a uniform sample prediction error bound for the unknown functions of factors, and the upper bound does not depend on the number of factors.

We develop a novel data-dependent estimator of the error covariance matrix in deep neural networks. The estimator refers to a flexible adaptive thresholding technique which is robust to outliers in the innovations. We prove that the estimator is consistent in spectral norm. Using this result, we show the consistency and provide the rates of convergence of the covariance matrix and precision matrix estimators for asset returns. The rates of convergence of both estimators do not depend on the number of factors. Hence, our theory leads to new results in the factor model literature due to the fact that an increasing number of factors in traditional factor models are an impediment to better estimation and prediction. Except for the precision matrix estimator, all our results are obtained even with a number of assets that is larger than the time span, where both quantities are allowed to grow.

Various Monte Carlo simulations confirm our large sample findings and reveal superior accuracies of the DNN-FM in estimating the true underlying functional form which connects the factors and observable variables, as well as the covariance and precision matrix compared to competing approaches. Moreover, in an out-of-sample portfolio forecasting application it outperforms in most of the cases alternative portfolio strategies in terms of out-of-sample portfolio standard deviation and Sharpe ratio.

Keywords: Deep neural networks, feedforward multilayer neural network, sparsity, nonparametric regression, covariance matrix estimation, factor models

1 Introduction

The Great Financial Crisis 2008/09 and the COVID-19 Crisis 2020/21 revealed major problems and disadvantages of existing models for forecasting and policy analysis. Especially during periods with high uncertainties

[*] North Carolina State University, Nelson Hall, Department of Economics, NC 27695. Email: mcaner@ncsu.edu.

[†] ETH Zürich, KOF Swiss Economic Institute, 8092 Zurich, Switzerland. Email: daniele@kof.ethz.ch

these models commonly cause large prediction errors and lead to misleading policy recommendations. Recent research developments indicate that machine learning methods suit better in these circumstances: they can measure non-linear structures and sudden changes in the relations between economic variables. Deep neural networks, i.e., artificial neural networks with many hidden layers, as part of the most important machine learning techniques nowadays, received increasing attention in statistics and economics over the recent years. Their usefulness has been shown on several complex machine learning problems that concern e.g., natural language processing and image recognition. A detailed overview of applications of deep neural networks on observed data can be found e.g., in Schmidhuber (2015) and the literature cited therein. More recently, Gu et al. (2021) develop a non-linear conditional asset pricing model based on an extended autoencoder incorporating latent factors that depend on asset characteristics and illustrate its superiority in terms of lowest pricing errors compared to standard methods commonly used in the finance literature.

In light of the advantages of deep neural networks in terms of prediction accuracy compared to traditional econometric models testified in various empirical studies and research fields, there is an increasing interest in analyzing their large sample properties. Important theoretical contributions that deal with the approximation properties of deep neural networks in the nonparametric regression framework are provided e.g., by Schmidt-Hieber (2020), Farrell et al. (2021) and Kohler and Langer (2021). For a rich class of composition based functions which include (generalized) additive models, Schmidt-Hieber (2020) shows that sparse deep neural networks do not suffer from the curse of dimensionality in contrast to traditional nonparametric estimation methods and achieve minimax rate of convergence in expected risk under the squared loss. While Schmidt-Hieber (2020) and the corresponding working paper version Schmidt-Hieber (2017) rely on the rectifier linear unit (ReLU) activation function in the deep neural network to derive the large sample properties, Bauer and Kohler (2019) also illustrate that deep neural networks with a smooth activation function, i.e., the sigmoidal activation function, avoid the curse of dimensionality in the nonparametric regression framework.

In this paper, we investigate the theoretical properties of feedforward multilayer neural networks (multilayer perceptron) with ReLU activation function and sparsely connected units in multivariate nonparametric regression models. Our modeling framework builds upon the research contributions of Schmidt-Hieber (2020) and offers convenient theoretical generalizations and extensions. Specifically, we relax the distributional assumption on the innovations in Schmidt-Hieber (2020) from standard normally distributed errors to subgaussian noise which enhances the theoretical scope of the deep neural network to a broader range of data generating processes.

Moreover, we contribute to the deep learning literature with uniform results on the expected estimation risk. More precisely, we analyze the properties of the deep neural network estimator in case of incorporating a potential set of J response variables compared to the current deep learning research that refers to univariate responses. Our results provide bounds on the expected risk and show that these upper bounds are uniform over the J variables. Hence, our model framework offers considerable extensions in terms of functional composition of the true underlying model and estimation. In fact, we allow for different unknown functional forms in the nonparametric regression model across each variable. The advantage of concentrating on the theoretical framework of Schmidt-Hieber (2020) lies in its applicability to factor models. We extend our deep learning results to an additive model setting and show its connection to factor models in finance. Compared to traditional static factor models, our deep learning factor model framework enhances the model flexibility, allows for measuring potential non-linear patterns in the data and avoids the curse of dimensionality when the number of factors increases with the number of variables in the system.

Linear factor models are helpful in understanding the behavior of asset returns. Fan et al. (2011) and Fan et al. (2013) use observed and unobserved linear factors in a static factor model framework to measure large asset portfolios. They show that linear factor models combined with a sparse error covariance matrix can be used to consistently estimate the precision matrix of asset returns even in high dimensions. These two papers are benchmarks in the literature and provide an important contribution in merging the factor models literature with the high-dimensional statistics literature. Fan et al. (2016) show that in case of an "almost" block diagonal covariance matrix of the errors, a sparse precision matrix of the innovations can exist, and can be used in linear factor models. Fan et al. (2021) propose factor models with a sparse regression structure. They build a test for analyzing the entries in the covariance matrix of the residuals estimated based on principal component regression. A key paper in the factor model literature is provided by Gagliardini et al. (2016). They analyze risk premia in large portfolios by introducing a structural model that can be tied to factor models. Gagliardini et al. (2019) propose a test for omitted factors in factor models by assessing the corresponding residuals which make the technical analysis challenging. Gagliardini et al. (2020) provide conditional factor models and analyze the risk premia when the number of assets in the portfolio is dominating its time span. Shrinkage methods that are tied to factor models have also been proposed in the literature. Linear shrinkage models are introduced in Ledoit and Wolf (2003) and Ledoit and Wolf (2004) for estimating the covariance matrix of asset returns and are applied to portfolio optimization. Recently, Ledoit and Wolf (2017) provide a non-linear shrinkage estimator and show its superiority in out-of-sample portfolio performance compared to the linear shrinkage based estimation. This is an important finding testifying that the flexibility of non-linear models can substantially improve the forecast performance. The non-linear method involves increasing the small eigenvalues, decreasing the large eigenvalues of a covariance matrix of asset returns, and optimizing a loss function by selecting a non-linear shrinkage function. Ao et al. (2019) show that choosing the weights of a portfolio with the lasso approach, and applying the method for estimating the mean return and variance of the portfolio leads to a consistent estimation of these two quantities. Ao et al. (2019) illustrate that they can convert a difficult constrained optimization problem into a feasible unconstrained portfolio optimization where the lasso estimation is crucial.

Recently, there is an increasing attention on factor models with deep learning methods. A major study is conducted by Fan et al. (2022) and tries to understand asset returns by conditional time varying factor models. The factor loadings are represented by nonparametric functions, and the authors use a structural model to decompose the predictors as risk-related and mispricing-related. They propose a three step method using deep learning to estimate the nonparametric spot asset returns, apply local averages to estimate the long term asset returns and rely on local principal components to estimate the factor loadings. They also develop the asymptotic theory to obtain the out-of-sample prediction of the risk. Chen et al. (2021) concentrate on a structural model and analyze asset prices with deep learning. They provide an empirical study showing that with a no-arbitrage condition as criterion, their method outperforms benchmark models.

Moreover, machine learning methods are also used to understand the cross-section of asset returns. Freyberger et al. (2020) fit an additive non-linear factor model to explain expected asset returns. The non-linear factor model is estimated by an adaptive group lasso, and the authors show that linear factor models overfit the data. In an empirical study based on data of the US stock market, the authors illustrate that the return to risk ratio of a portfolio constructed by a non-linear factor model can be considerably higher compared to the one resulting from a linear factor model. This is an important finding highlighting the key role of measuring potential non-linearities in the data. Giglio and Xiu (2021) analyze a linear factor model with omitted factors. They develop a three step method to estimate risk premia by principal component analysis with two-pass regressions. Bianchi et al. (2021) analyze the bond market with machine learning techniques. Callot et al. (2021) show that risks of large portfolios can be estimated consistently based on nodewise regression. Moreover, the authors illustrate that the approach performs very well in terms of risk and return prediction in practice.

Traditional factor models, as in Fama and French (1993) or Fan et al. (2011) assume rigid linear relations between the factors and the underlying observed variables. Especially during crisis periods, however, economic time series follow non-linear relations and are subject to sudden changes. Hence, the linearity assumption of standard factor models would be inappropriate to measure time series with these patterns. The structure of our deep neural network factor model (DNN-FM) mitigates this limitation and allows for measuring non-linear and complex dynamic relations between the factors and economic variables. Our theoretical elaboration provides convergence results for the covariance and precision matrix estimators based on the sparse deep neural network. Moreover, we show that the convergence rate is not affected by the number of included factors which potentially increases with the number of variables in the asset space. Hence, the DNN-FM estimator avoids the curse of dimensionality and provides major improvements in modeling flexibility compared to traditional factor models. In fact, the convergence rate of the corresponding covariance matrix estimator based on traditional factor models is affected by the number of factors and hence, may severely deteriorate in high-dimensional asset spaces with an increasing number of factors.

In order to obtain an estimate of the data covariance matrix based on the DNN-FM, we require a consistent estimate for the residual covariance matrix. For this purpose, we develop a novel data-dependent covariance matrix estimator of the innovations in deep neural networks. The estimator refers to a flexible adaptive thresholding technique which is robust to outliers in the innovations. We elaborate on the consistency of the estimator in l_2-norm under rather mild assumptions.

The favorable large sample properties of the DNN-FM are confirmed by our Monte Carlo study based on various simulation designs. Specifically, the DNN-FM consistently determines the true underlying function connecting the factors and observable variables, as the number of periods increases. Further, its estimators for the covariance and precision matrix of the returns are as well consistent. Compared to competing approaches, as e.g., the traditional static factor model which are sensitive to an increase of the number of factors d (i.e., their error rate deteriorate as d increases), the convergence rates of the DNN-FM estimators are stable against an increasing number of factors. Hence, the DNN-FM avoids the curse of dimensionality compared to standard factor models. Moreover, in most cases, the simulation results demonstrate the superior accuracy in estimating the unknown functional form linking the factors and the observed time series, as well as the corresponding covariance and precision matrix.

In an out-of-sample portfolio forecasting application based on assets constituents of the S&P 500 stock index and concentrating on a global minimum variance portfolio setting, we show that the DNN-FM is superior in most cases to compared to competing portfolio estimators that are commonly used in the literature. More precisely, it often leads to the lowest out-of-sample portfolio standard deviation and avoids large changes in the portfolio constellation. Consequently, it provides low portfolio turnover rates and prevents high transaction costs. This generally leads to the highest out-of-sample Sharpe rations across different asset spaces compared to the competing methods when transaction costs are taken into account. The superiority in forecasting precision is particularly pronounced during turbulent times, such as during the Great Financial Crisis of 2008/09 and the COVID-19 Crisis. Hence, the flexibility of the deep learning estimator thus allows for capturing strong changes and high uncertainties in financial time series during volatile periods.

The remainder of the paper is organized as follows. In Section 2, we introduce the sparse deep neural network framework used for estimating nonparametric regressions and provide our main theoretical findings. Section 3 extends the results to an additive model setting applied to factor models in financial applications and elaborates on the convergence results. In Section 4, we introduce a novel estimator for the covariance and precision matrix for the idiosyncratic errors of the deep neural network factor model by means of a robust adaptive threshold estimator and prove its consistency. Moreover, we elaborate on the consistency of the corresponding covariance matrix estimator of the returns in Section 5, while Section 6 introduces the precision matrix estimator of the returns based on the DNN-FM and shows the consistency of the estimator.

Details on the implementation are discussed in Section 7. In Section 8, we present Monte-Carlo evidence on the finite sample properties of our new sparse deep neural network factor model in estimating the unknown function which connects the factors with the observable variables, as well as the corresponding covariance and precision matrix. In Section 9 we analyze the empirical performance of our approach in an out-of-sample portfolio forecasting exercise. Section 10 summarizes the main findings. The proofs are provided in the Appendix A. For a generic $n \times 1$ vector, v, $\|v\|_n := \frac{1}{n}\sum_{i=1}^{n} v_i^2$. For a generic matrix A, $\|A\|_{l_2}, \|A\|_{l_\infty}$ represent the spectral norm, and maximum row sum norm, respectively, as defined on pp. 345-346 in Horn and Johnson (2013). Furthermore $\|A\|_\infty$ is the sup norm which is the maximum absolute value element of the matrix A.

2 Deep Learning

In this section, we extend the deep learning results of Schmidt-Hieber (2020) from Gaussian noise to subgaussian noise and analyze the properties of deep neural networks (i.e., neural networks with many hidden layers) incorporating a large set of response variables. Moreover, in the next sections we use our framework in an additive model setting and apply it to factor models in finance.

Assume the following nonparametric regression model for $j = 1, \cdots, J$

$$Y_{j,i} = f_{0,j}(X_i) + u_{j,i}, \tag{1}$$

where $Y_{j,i}$ denotes the i-th observation of the j-th response variable, the regressors X_i are iid across $i = 1, \cdots, n$, including observed variables of dimension $d \times 1$, and each $X_i \in [0,1]^d$. J and n correspond to the number of response variables and the number of observations, respectively. Let $u_{j,i}$ be subgaussian noise, and iid across $i = 1, \cdots, n$, with zero mean, and variance as one,[1] for all $j = 1, \cdots, J$. Define the $J \times J$ covariance matrix of the errors as $\Sigma_u := E u_i u_i'$, where $u_i := (u_{1,i}, \cdots, u_{j,i}, \cdots, u_{J,i})'$ is a $J \times 1$-dimensional vector. $f_{0,j}(.)$ is an unknown function linking the regressors to the response variables. For each $j = 1, \cdots, J$, it may be of a different functional form, however $f_{0,j}(.) : [0,1]^d \to R$. Moreover, we assume that $u_{j,i}$ is independent of X_i for a given j. In order to estimate $f_{0,j}(.)$ with deep learning, we need to specify a structural form on $f_{0,j}(.)$. More precisely, we impose $f_{0,j}(.)$ to be a composition of several functions with lower dimensions which exhibit Hölder smoothness. To break the curse of dimensionality that may arise from a large number of regressors, d, we require a similar model structure as in the theoretical framework of Schmidt-Hieber (2020). The results of Schmidt-Hieber (2020) are path breaking in nature due to overcoming the curse of dimensionality. Specifically, he shows that compared to traditional nonparametric estimators, the convergence of deep learning methods in expected risk is unaffected by d. In addition to the composite structure for $f_{0,j}(.)$, we use deep learning as in Schmidt-Hieber (2020) to estimate $f_{0,j}(.)$. We benefit from newly developed s-sparse deep neural networks. In the following, we define the deep learning architecture.

2.1 Multilayer Neural Networks

We rely on multilayer (deep) feedforward neural networks to estimate the unknown function $f_{0,j}(.)$. The "Deepness" in the neural network arises from using multiple hidden layers to estimate $f_{0,j}(.)$, for $j = 1, \cdots, J$. "Learning" comes from estimating and correcting the errors in the network parameters via an algorithm (i.e., stochastic gradient descent) which will be described in more detail in Section 7 dealing with the implementation issues. Further information can be also found e.g., in Goodfellow et al. (2016). We

[1] A variance of one is chosen to simplify the proofs. However, it can be a positive constant as well.

are concerned about the statistical properties of deep learning, specifically multilayer feedforward neural networks. In the following, we briefly describe their structural architecture. As usual in nonparametric problems, we start with the regressors, X_i and need to determine the response $Y_{j,i}$. In order to achieve that we have to estimate the unknown function $f_{0,j}(.)$ in (1) connecting X_i with $Y_{j,i}$. The deep learning approach puts forward L hidden layers between the input layer containing the regressors and the output layer incorporating the response. Hence, $f_{0,j}(.)$ will be estimated in $L \geq 1$ hidden layers. A given hidden layer is composed of p_l hidden units, with $l = 1, \cdots, L$. These units are transformed by an activation function: $\sigma : R \to R$ at each layer. We impose a rectifier linear unit (ReLU) activation function for σ

$$\sigma(x) = \max(x, 0),$$

which censors all negative real numbers to zero, and keeps all positive. The choice of this activation function is motivated by several factors. First, the partial derivative of the ReLU function is either zero or one, which facilitates computations. Moreover, the ReLU activation can pass a signal without any change through all layers, i.e., it has the projection property. For each $j = 1, \cdots, J$, we assume the same number of layers L. Moreover, we assume the same number of units at each layer for $j = 1, \cdots, J$. However, for each j, the sparsity structure of the network can be different. In other words, even though the number of layers and the number of units that are fitted are the same, the number of nonzero parameters (network weights) in each layer can be different, for each $j = 1, \cdots, J$.

We define the network architecture by the parameters (L, p), where the number of hidden layers L represents the depth of the neural network, and p constitutes the width of the network. Specifically, $p = (p_0, p_1, \cdots, p_L, p_{L+1})'$ which is a $L + 2$ vector, where p_0 is the number of input variables and p_{L+1} defines the number of units in the output layer. The remaining quantities $p_1, \cdots, p_L$, represent the number of units in the layers $1, \cdots, L$, respectively. In this paper, we impose one output variable, hence $p_{L+1} = 1$. We define the shifted activation function as the $p_l \times 1$ vector

$$\sigma_{v_l} := [\sigma(z_1 - v_1), \cdots, \sigma(z_{p_l} - v_{p_l})]',$$

where z_l are the input variables of layer l. In order to describe the network, $f : R^{p_0} \to R$, $x \to f(x)$, we specify the input layer, hidden layers and output relation using the following simplistic structure

$$f(x) = W_L \sigma_{v_L} W_{L-1} \sigma_{v_{L-1}} \cdots W_1 \sigma_{v_1} W_0 x,$$

where W_l denotes the $p_{l+1} \times p_l$-dimensional weight matrix at the l-th layer, for $l = 0, 1, \cdots, L$, and $v_l \in R^{p_l}$ is the l-th shift vector, for $l = 1, \cdots, L$. The network functions are built by matrix-vector multiplication in a sequential manner and are transformed based on the non-linear activation function $\sigma(.)$. The columns of the weight matrix W_l denote the number of weights corresponding to the input dimension of layer l and the rows represent the number of weights in the following layer $l + 1$.

Layer $l = 0$ represents the input layer which incorporates the observed regressors X_i. The layers $l = 1, \cdots, L$ are hidden layers, whereas $l = L + 1$ represents the output layer. The inputs/regressors are of dimension $p_0 = d$, and the output layer is uni-dimensional. Let W represent the total number of parameters that are optimized in the neural network. Then

$$W = \sum_{l=0}^{L} (p_l + 1) p_{l+1} - p_{L+1},$$

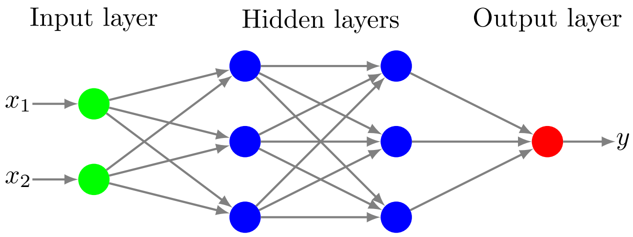


Figure 1: Graphical representation of a multilayer feedforward neural network with two hidden layers, two regressors, one response and three units at each hidden layer.

where we set $p_{L+1} = 1$. Our paper allows for $W > n$, hence, the number of weights in the neural network can exceed the number of observations of the underlying variables.

In the following, we provide a simple example of a multilayer feedforward neural network illustrated in the directed acyclic graph in Figure 1. The neural network incorporates two regressors $d = 2$ and one response. Moreover, it contains two hidden layers ($L = 2$) with three units at each hidden layer, i.e., $p_1 = p_2 = 3$.

The first connection arises from the regressors which are collected in the input layer to the first hidden layer. Each regressor is forwardly connected with each unit in the first layer. Moreover, at each hidden unit the regressors are multiplied with the corresponding weights and an intercept is added. Hence, for the first unit in the first hidden layer three parameters need to be estimated. The remaining units follow an identical structure. In addition, we apply the ReLU activation function to each unit. In our example, nine parameters have to estimated in total, and three ReLU units are applied in the first hidden layer. In a second step, each unit in the first hidden layer is connected with each unit in the second hidden layer. More specifically, the outputs from the first hidden layer are used as inputs to each unit at the second hidden layer. Each of those inputs is multiplied with the corresponding weights, an intercept is added and the the ReLU activation function is applied. For our example, we need to estimate 12 parameters, formed by three weights and an intercept at each unit in second hidden layer. Finally, in the output layer, we use the least squares loss and regress Y_i on all three responses from the second hidden layer. This adds three additional parameters that have to be estimated. Hence, in total we need to estimate 24 parameters, if the regressor vector X_i is 2×1-dimensional and if we incorporate two hidden layers with three units each.

2.2 S-sparse Deep Neural Networks

In this part, we introduce s-sparse deep neural networks which are a subset of multilayer feedforward neural networks. Schmidt-Hieber (2020) introduces them in equation (4) of his paper and requires two important restrictions. First, the parameters will be bounded by one, and second the network will be sparse. The first restriction can be formally depicted as follows: For each $j = 1, \cdots, J$

$$\max_{0 \leq l \leq L} [\|W_l\|_\infty \vee \|v_l\|_\infty] \leq 1. \tag{2}$$

This restriction is necessary to ensure the applicability of the neural networks in practice. Statistical results using large parameters/weights are usually not observed, see Goodfellow et al. (2016). To be specific, Schmidt-Hieber (2020) and Zhong et al. (2022) argue that computational algorithms use random, nearly orthogonal matrices as initial weights. Hence, the initial weights are bounded by one, and the model training leads to trained weights which are close to the initial values. Therefore, it seems reasonable to set a bound of one for the weights.

In the next step, we introduce our sparsity assumption which is crucial to prevent any overfitting caused by a probable large number of parameters to be estimated in each layer, as discussed in p. 1352 of Zhong et al. (2022). Let $\|W_l\|_0$ and $\|v_l\|_0$ denote the number of nonzero entries of W_l and v_l, respectively. The s-sparse deep neural networks are given by the following equation, subject to (2)

$$\mathcal{F}_j(L, p, s_j) := \{f_j(\cdot) : \sum_{l=0}^{L} \|W_l\|_0 + \sum_{l=1}^{L} \|v_l\|_0 \leq s_j, \quad \max_{1\leq j\leq J} |f_j(\cdot)| \leq F\}, \tag{3}$$

where the output dimension is one and the functions are uniformly bounded by the positive constant F. Clearly, the sparsity is different for each $\mathcal{F}_j(L, p, s_j)$. In order to simplify the notation, we represent $\mathcal{F}_j(L, p, s_j)$ as $\mathcal{F}_j$ in what follows. The sparsity structure imposed on the neural network weights are additionally valuable for the consistency of the of deep learning estimators compared to model specifications that only rely on the implicit regularization introduced by the stochastic gradient decent algorithm during the computational stage. This point is conjectured explicitly, and an example is provided on p. 1916 of Schmidt-Hieber (2020). To obtain a sparse structure, pruning the weights is another solution as illustrated by Zhong et al. (2022). Note that Kohler and Langer (2021) show that deep learning error bounds can be obtained without sparsity and bounded weights but with a truncated least squares function.

2.3 Composite True Function

In the following, we incorporate smoothness restrictions and a composite form to estimate the true function $f_{0,j}(.)$ by s-sparse deep neural networks. For each $j = 1, \cdots, J$

$$f_{0,j}(.) = g_{q,j}(.) \circ g_{q-1,j}(.) \circ \cdots \circ g_{1,j}(.) \circ g_{0,j}(.), \tag{4}$$

where $f_{0,j}(.)$ is a composition of $q+1$ functions. Let $h = 0, \cdots, q$, $g_{h,j} : [a_h, b_h]^{d_h} \to [a_{h+1}, b_{h+1}]^{d_{h+1}}$. Hence, each $g_{h,j}$ has $m = 1, \cdots, d_h + 1$ components:

$$g_{h,j} = (g_{h1,j}, \cdots, g_{hm,j}, \cdots, g_{hd_{h+1},j})'.$$

At each $g_{hm,j}$ we assume that it depends on maximal t_h number of variables. In other words, each $g_{hm,j}$ is a t_h variate function itself. To give an example, for any j (suppressing j only here for clarity), let $f_0(x_1, x_2, x_3) = g_1(.) \circ g_0(.) = g_{11}(g_{01}(x_1, x_3), g_{02}(x_2, x_3))$, with $g_1 = g_{11}, d_2 = 1, g_0 = (g_{01}(x_1, x_3), g_{02}(x_2, x_3))$, such that $d_0 = 3, t_0 = 2$ and $d_1 = t_1 = 2$. Note that d_0 shows the dimension of the input variables. However, both g_{01}, g_{02} depend on only 2 variables, hence $t_0 = 2$. We always demand that $t_h \leq d_h$. This restriction is crucial, and is available in additive models which we thoroughly analyze in the subsequent sections. The previous example is taken from p. 1880 of Schmidt-Hieber (2020).

We impose the following smoothness restrictions on $g_{hm,j}$. Let β_f denote the largest integer strictly smaller than β. A function has Hölder smoothness index β if all partial derivatives up to order β_f exist and are bounded, and the partial derivatives of order β_f are $\beta - \beta_f$ Hölder. We impose that each $g_{hm,j}$ has Hölder smoothness β_h. Also remember that $g_{hm,j}$ has t_h variables, so each $g_{hm,j} \in \mathcal{C}^{\beta_h}_{t_h}([a_h, b_h]^{t_h}, K_h)$ and $\mathcal{C}^{\beta_h}_{t_h}([a_h, b_h]^{t_h}, K_h)$ is the ball of β_h Hölder functions with radius K_h defined as

$$\begin{aligned} \mathcal{C}^{\beta_h}_{t_h} \quad (\quad & [a_h, b_h]^{t_h}, K_h) := [g_{hm,j} : [a_h, b_h]^{t_h} \to R : \\ & \sum_{\alpha:\|\alpha\|_1<\beta_h} \|\partial^\alpha g_{hm,j}\|_\infty + \sum_{\alpha:\|\alpha\|_1=\beta_f} \sup_{x,y\in[a_h,b_h]^{t_h}, x\neq y} \frac{\|\partial^\alpha g_{hm,j}(x) - \partial^\alpha g_{hm,j}(y)\|_1}{\|x-y\|_\infty^{\beta-\beta_f}} \leq K_h.] \end{aligned}$$

So we can have different flexible composite functions, where the smoothness will be the same for each $m = 1, \cdots, d_{h+1}, j = 1, \cdots, J$, but differ across h. We impose the following for the true function, $f_{0,j} \in \mathcal{G}(q, d, t, \beta, K)$ with

$$\begin{aligned} \mathcal{G}(q, d, t, \beta, K) \quad := \quad & \{f_{0,j} = g_{q,j} \circ g_{h,j} \circ \cdots \circ g_{0,j} : g_{h,j} = (g_{hm,j})_{m=1}^{d_h+1} : [a_h, b_h]^{d_h} \to [a_{h+1}, b_{h+1}]^{d_{h+1}} \\ & g_{hm,j} \in \mathcal{C}_{t_h}^{\beta_h}([a_h, b_h]^{t_h}, K), \text{ for some } \quad |a_h|, |b_h| \leq K\}. \end{aligned} \tag{5}$$

We define $d := (d_0, \cdots, d_h, \cdots, d_{q+1})', t := (t_0, \cdots, t_h, \cdots, t_q),\ \beta := (\beta_0, \cdots, \beta_h, \cdots, \beta_q)'$. There are two key terms to be defined: the effective smoothness index

$$\beta_h^* := \beta_h \, \Pi_{l=h+1}^{q} (\beta_l \wedge 1),$$

and the rate which will be used in prediction error estimation

$$\phi_n := \max_{h=0,\cdots,q} n^{\frac{-2\beta_h^*}{2\beta_h^* + t_h}}.$$

2.4 Estimator

We define the estimator as the empirical minimizer, with $\mathcal{F}_j$ representing the s-sparse deep neural network in (2) and (3) as follows

$$\hat{f}_j(X_i) \in \operatorname*{argmin}_{f_j(X_i) \in \mathcal{F}_j} \sum_{i=1}^{n} (Y_{j,i} - f_j(X_i))^2.$$

Let $E_{f_{0,j}}$ represent the expectation with respect to a sample generated by $f_{0,j}(.)$. We evaluate the out-of-sample performance of s-sparse deep neural networks. We assume that X has the same distribution as X_i, but is independent from the sample $(X_i, Y_{j,i})$ across i and given j. In the following, we want to evaluate the prediction error

$$R(\hat{f}_j(X), f_{0,j}(X)) := E_{f_{0,j}}[(\hat{f}_j(X) - f_{0,j}(X))^2].$$

For this, we formalize an assumption about the data.

Assumption 1. *Assume that $u_{j,i}$ are iid zero mean with unit variance across $i = 1, \cdots, n$ with subgaussian distribution and Orlicz norm, $\max_{1 \leq j \leq J} \|u_{j,i}\|_{\psi_2} := C_\psi$ which is a positive constant. Let the minimum eigenvalue of the covariance matrix of the errors be: $Eigmin(\Sigma_u) \geq c > 0$, where c is a positive constant. X_i are iid across $i = 1, \cdots, n$, independent from $u_{j,i}$, for each j, and the model is defined by (1).*

Assumption 2. *Let $\hat{f}_j(.) \in \mathcal{F}_j$ satisfy (2) and (3).*

Assumptions 1 and 2 are used to derive an oracle inequality which is provided by Lemma A.3. Assumption 1 is a subgaussian noise extension of the gaussian noise assumption imposed in Schmidt-Hieber (2020). Assumption 2 is similarly used as in Schmidt-Hieber (2020) to describe the sparse neural network architecture.

Assumption 3. *Assume $f_{0,j}(.)$ is a composite function generated by equation (4), and $f_{0,j}(.) \in \mathcal{G}(q, d, t, \beta, K)$, for each $j = 1, \cdots, J$.*

Define $\bar{s} := \max_{1 \leq j \leq J} s_j$, and $\tilde{s} := \min_{1 \leq j \leq J} s_j$.

Assumption 4. *Let $\hat{f}_j(.) \in \mathcal{F}_j$ satisfy*

(i) $F \geq \max(K, 1)$,

(ii)

$$\sum_{h=0}^{q} log_2(4t_h \cup 4\beta_h)\, log_2 n \leq L \leq Cn\phi_n,$$

(iii) $n\phi_n \leq C \min_{1\leq l \leq L} p_l$,

(iv) $C_0 n\phi_n \log n \leq \tilde{s} \leq \bar{s} \leq Cn\phi_n \log n$*, with* $C_0 < C$*, where both are positive constants.*

Assumptions 3 and 4 are used to approximate the true function, $f_{0,j}(.)$ by the sparse deep learning estimator $\hat{f}_j$. Assumption 3 specifies the true composite function. Assumptions 4(i)-(ii) put an upper bound on the functions, and specify the number of layers L. The number of layers is an increasing function of n, and the optimal number of layers is shown in Remark 1 on the Remarks for Theorem 1, depicted after Theorem 1. The number of units at each layer is specified in Assumption 4(iii) which provides a lower bound of this quantity. Hence, the number of units increases with n, and form a wide layer. The sparsity restriction is specified in Assumption 4(iv). Given the specification of the rate ϕ_n, the sparsity assumption shows that $\bar{s}/n \to 0$. Assumptions 3 and 4 are directly taken from Schmidt-Hieber (2020). In the following, we provide the upper bound on the risk for our functions that are estimated by deep learning.

Theorem 1. *Under Assumptions 1-4, for a large positive constant* $C > 0$

(i)

$$\max_{1\leq j\leq J} R(\hat{f}_j(X), f_{0,j}(X)) \leq C\phi_n L \log^2 n.$$

(ii)

$$\max_{1\leq j\leq J} E\left[\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i) - f_j(X_i))^2\right] \leq C\phi_n L \log^2 n.$$

Remarks.

1. To minimize the right side in Theorem 1 set $L = O(log_2 n)$, in which case the right side is

$$\max_{1\leq j\leq J} R(\hat{f}_j(X), f_{0,j}(X)) \leq C\phi_n \log^3 n.$$

2. In order to obtain a good function approximation, we need a large number of layers, and the number of units at each hidden layer cannot be small. These restrictions are specified in Assumption 4. However, for a better prediction we require fewer layers as observed by our result in Theorem 1. Hence, there is a tradeoff in the selection of the number of layers. The deepness of the neural network is needed but the issue becomes how deep? Theorem 5 in Schmidt-Hieber (2020) makes the point for deep layers to obtain a better function approximation. At each hidden layer we can have at least $n\phi_n$ units, hence there is a lower bound to obtain a finer function approximation by deep neural networks.

3. Sparsity can be achieved by active regularization at each layer through elastic net penalization, or an iterative pruning approach as suggested by p. 1882, and p. 1917 of Schmidt-Hieber (2020), respectively. Lemma 5 of Schmidt-Hieber (2020) combined with our Lemma A.3 shows that the sparsity in the network parameters at each layer is essential for a better prediction. Implicit regularization in stochastic gradient descent algorithm will not be sufficient, and consistency may be not possible as shown in pp. 1916-1917 of Schmidt-Hieber (2020).

4. Our result further shows that the upper bound is uniform over $j = 1, \cdots, J$. This is due to Assumption 4(iv). Moreover, we also allow that the true functional forms are different. However, they have the same Hölder smoothness. In case of deep learning estimators, we have the same number of layers and units for each estimate but their sparsity pattern can vary.

3 Additive Models and Factor Models

In the finance literature, asset returns, $Y_{j,i}$, for asset j, and time period i, are governed by common factors. The following model is used in Fan et al. (2011)

$$Y_{j,i} = \sum_{m=1}^{d} b_{j,m} X_{m,i} + u_{j,i}, \tag{6}$$

for all $j = 1, \cdots, J$, where J represents the number of assets in the portfolio, and $i = 1, \cdots, n$ denotes the time span of the portfolio. The number of factors is d and the factors are observed. $X_{m,i}$ represents the m-th factor at time period i, and $b_{j,m}$ depicts the factor loading corresponding to the j-th asset and m-th factor. The model described above defines a linear relation between the factors and returns through the factor loadings. However, a more flexible relationship between the asset returns and factors can be put forward through the additive model described in Section 2.3. Instead of the linear model specification in (6), we assume that the returns evolve through the following model

$$Y_{j,i} = f_{0,j}(X_i) + u_{i,j}, \tag{7}$$

with

$$f_{0,j}(X_i) := \sum_{m=1}^{d} f_{j,m}(X_{m,i}),$$

where $f_{0,j}(.)$ represents the true but unknown function relating the factors $X_i := (X_{1i}, \cdots, X_{m,i}, \cdots, X_{d,i})' : d \times 1$ to the returns for each asset. $f_{j,m}(.)$ corresponds to the unknown function of factor $X_{m,i}$ for the m-th factor at time period i. Hence, even though the underlying factor is the same, the functional forms are different across the assets. The model specification (7) enhances the flexibility in the relationship between the factors and assets to a large extend, compared to the rigid all linear additive formulae (6).

Note that Freyberger et al. (2020) also use a non-linear model with firm characteristics to explain asset returns. The authors rely on the adaptive group lasso approach and benefit from the sparsity structure in the model. We want to estimate $f_{0,j}(.)$ with sparse deep learning, for each $j = 1, \cdots, J$. The additive-flexible model (7) is related to Section 4 and equation (12) of Schmidt-Hieber (2020). We can specify the true function $f_{0,j}(.)$ as the composite of two functions as follows

$$f_{0,j}(.) = g_{1,j}(.) \circ g_{0,j}(.), \tag{8}$$

where

$$g_{0,j}(.) := [g_{0,j,1}(.), \cdots, g_{0,j,m}(.), \cdots, g_{0,j,d}(.)]' := [f_{j,1}(.), \cdots, f_{j,m}(.), \cdots, f_{j,d}(.)]',$$

which is a $d \times 1$-dimensional vector. Then

$$g_{1,j}(X_i) := \sum_{m=1}^{d} g_{0,j,m}(X_{m,i}) := \sum_{m=1}^{d} f_{j,m}(X_{m,i}).$$

The following notation is also used on pp. 1884-1885 of Schmidt-Hieber (2020) and we adopt it to our specification. The sum above is valid for $X : d \times 1$ as well. These last definitions show that, $d_0 = d, t_0 = 1, d_1 = t_1 = d$, $d_2 = 1$ in (4). The dimension of $g_{0,j}(.)$ is d, however each scalar function is used in a sum of d terms. See that $g_{0,j} : [0,1]^d \to R^d$, and $g_{1,j}(.) : R^d \to R$. Suppose that $f_{j,m} \in \mathcal{C}_1^\beta([0,1], K)$ for $m = 1, \cdots, d$. It follows that

$$f_{0,j} : [0,1]^d \overset{g_{0,j}}{\to} [-K, K]^d \overset{g_{1,j}}{\to} [-Kd, Kd].$$

For any $\gamma > 1$, $g_{1,j}(X_i) \in \mathcal{C}_d^\gamma([-K,K]^d, (K+1)d)$, and set the dimension of inputs d, d, in each composite function $g_{0,j}, g_{1,j}$ and output dimension 1, as $\tilde{d} := (d, d, 1)$ with $d_0 = d, d_1 = d, d_2 = 1$, and $\tilde{t} = (1, d)$. Last, set the smoothness indicators for each composite as $\tilde{\beta} := (\beta, (\beta \vee 2)d)$ for g_0, g_1, respectively. Also define a positive constant $c_1 > 0$.

In the following, we introduce modified versions of Assumptions 3 and 4 for additive factor models. We use equations (5) and (8) for the notation below.

Assumption 5.

$$f_{0,j} \in \mathcal{G}(1, \tilde{d}, \tilde{t}, \tilde{\beta}, (K+1)d),$$

and

Assumption 6.

$$F \geq (K+1)d,$$

$$L = O(log_2 n),$$

$$n^{1/(2\beta+1)} \leq C \min_{1 \leq l \leq L} p_l,$$

$$C_0 n^{1/(2\beta+1)} \log n \leq \tilde{s} \leq \bar{s} \leq C n^{1/(2\beta+1)} \log n,$$

In the following, we provide a maximal inequality based on Theorem 1.

Theorem 2. *Under Assumptions 1-2, 5-6*

$$P\left[\max_{1 \leq j \leq J} \frac{1}{n} \sum_{i=1}^{n} [\hat{f}_j(X_i) - f_{0,j}(X_i)]^2 \geq r_{n1} + r_{n2}\right] \leq \frac{1}{J^{c_1^2}},$$

with rates $r_{n1} = O(n^{\frac{-2\beta}{2\beta+1}} \log^3 n)$ *and* $r_{n2} = O(\sqrt{\log J/n})$.

Remarks.

1. An important aspect concerns the consistency of the deep neural network estimator which is not affected by number of factors, d. This is due to additive structure of the model, the novel proofs in Schmidt-Hieber (2020) and our proof for the subgaussian noise in Theorem 1. The rates r_{n1} and r_{n2} correspond to the rate of convergence of the deep learning risk and the rate following because of considering J assets in a portfolio, respectively.

2. Theorem 2 is a new result and shows how the size of the portfolio and the number of factors affect the deep learning estimate of the true underlying function that relates the factors to the returns. Our result extends Schmidt-Hieber (2020) to the estimation of multiple functions, and analyzes the sample prediction error rather than its expected value. Hence, we obtain the additional rate $r_{n,2}$ due to estimation of J functions.

3. We can specify which rate may be the slowest among $r_{n,1}$ and $r_{n,2}$. Suppose that $J = \exp(n^{a_1})$ and $0 < a_1 < 1$. We obtain $r_{n,2}$ as the slowest one as long as $(a_1 - 1)/2 > -2\beta/(2\beta + 1)$ which is true with $\beta \geq 1/2$. Hence, regardless of $0 < a_1 < 1$, when $\beta \geq 1/2$, we have the rate $r_{n,2}$ with $J = \exp(n^{a_1})$.[2] If we consider $J = a_2 n$ and $0 < a_2 \leq C < \infty$, then with $\beta > 1/2$, we obtain $a_n^2 = O(r_{n,2})$. A similar logic as in case of $J = a_2 n$ applies also to the case of $J = n^{a_3}$, with $a_3 > 0$. Note that if $\beta < 1/2$, it is not clear which rate will be slowest. The rate depends on the tradeoff between the smoothness coefficient β, and the number of assets J.

4 Covariance and Precision Matrix Estimate for Errors

In this section, we analyze the large sample properties of the error covariance and precision matrix estimates based on the sparse deep neural network. We start with denoting $\sigma_{j,k} := (\Sigma_u)_{j,k}$ as the (j,k)-th element of the Σ_u matrix, with $j = 1, \cdots, p, k = 1, \cdots, p$. The following assumption sets a restriction on J. Moreover, it constrains the minimal absolute covariance of the errors to be positive, and bounded away from zero uniformly in n.

Assumption 7.

(i)

$$\sqrt{\log J/n} \to 0.$$

(ii)

$$\min_{1 \leq j \leq J, 1 \leq k \leq j} E|u_{j,i} u_{k,i} - \sigma_{j,k}| \geq c > 0.$$

In Lemma 1 we provide two maximal inequalities: the first one for estimation of covariance matrix for errors with infeasible sample average via Bernstein's inequality, and the second one provides a centered absolute version of the first maximal inequality.

Lemma 1. *Under Assumption 1, with positive constants $C, c_2 > 0$, and with sufficiently large n*[3]

(i)

$$P\left[\max_{1 \leq j \leq J} \max_{1 \leq k \leq J} |\frac{1}{n} \sum_{i=1}^{n} u_{j,i} u_{k,i} - E u_{j,i} u_{k,i}| \geq C \frac{\sqrt{\log J}}{\sqrt{n}}\right] \leq \frac{2}{J^{c_2}}.$$

(ii)

$$P\left[\max_{1 \leq j \leq J} \max_{1 \leq k \leq J} \left| \frac{1}{n} \sum_{i=1}^{n} |u_{j,i} u_{k,i} - \sigma_{j,k}| - E|u_{j,i} u_{k,i} - \sigma_{j,k}| \right| > C \frac{\sqrt{\log J}}{\sqrt{n}}\right] \leq \frac{2}{J^{c_2}}$$

In what follows, we establish novel asymptotic results on deep learning residuals. As far as we know, these will be the first in the literature. Let $M > 0$ be a large positive constant. Moreover, define the expression

$$a_n^2 := r_{n1} + r_{n2} = O(\max(r_{n1}, r_{n2})). \tag{9}$$

[2] We can have a more precise result with rate r_{n2} as the rate of convergence if we know a_1, and if $\beta > \frac{1-a_1}{2(1+a_1)}$

[3] To be specific, a sufficiently large n means that $n \geq \frac{C^2}{C_\psi^4} \log J$ for Lemma 1.

The results provided in Lemma 2 are new for residuals based on deep learning estimators in factor models. Define $C_* \geq (2C+1) > 0, C_{**} \geq (3C+1) > 0$ to be positive constants.

Lemma 2. *Under Assumptions 1,2,5-7(i)*

(i)

$$P\left[\max_{1\leq j\leq J} \frac{1}{n}\sum_{i=1}^{n}(u_{j,i}-\hat{u}_{j,i})^2 > a_n^2\right] \leq \frac{1}{J^{c_1}}.$$

(ii)

$$P\left[\max_{1\leq j\leq J}\max_{1\leq k\leq J} |\frac{1}{n}\sum_{i=1}^{n}(\hat{u}_{j,i}\hat{u}_{k,i}-u_{j,i}u_{k,i})| > C_* a_n\right] \leq \frac{2}{J^{c_2}}.$$

(iii)

$$P\left[\max_{1\leq j\leq J}\max_{1\leq k\leq J} |\frac{1}{n}\sum_{i=1}^{n}\hat{u}_{j,i}\hat{u}_{k,i}-Eu_{j,i}u_{k,i}| > C_{**}(\sqrt{\frac{\log J}{n}}+a_n)\right] \leq \frac{2}{J^{c_1}}+\frac{2}{J^{c_2}}.$$

Note that the results in Lemma 2 are used to obtain the consistency of the covariance and precision matrix estimators of the returns based on the sparse deep neural network which is illustrated in the upcoming sections.

In the following, we specify the covariance matrix of the errors and its thresholding counterpart. Define the covariance matrix of the errors as $\Sigma_u := Eu_iu_i'$, with $u_i := (u_{1,i},\cdots,u_{j,i},\cdots,u_{J,i})' : J\times 1$. Set $Y_i := (Y_{1,i},\cdots,Y_{j,i},\cdots,Y_{J,i})' : J\times 1$. The sample estimator is given by

$$\hat{\Sigma}_u := \frac{1}{n}\sum_{i=1}^{n}\hat{u}_i\hat{u}_i',$$

where $\hat{u}_i := Y_i - \hat{f}(X_i)$ is the $J\times 1$ vector of residuals based on the deep learning estimator. Also define the j,k-th element of $\hat{\Sigma}_u$ as $\hat{\sigma}_{j,k}$, $j=1,\cdots,J, k=1,\cdots,J$. We provide a new robust-adaptive threshold estimator for the error covariance matrix estimation. This is robust to outliers in the data, i.e., it is robust to large residuals. To that end, define, for $j=1,\cdots,J, k=1,\cdots J$

$$\hat{\theta}_{j,k} := \frac{1}{n}\sum_{i=1}^{n}|\hat{u}_{j,i}\hat{u}_{k,i}-\hat{\sigma}_{j,k}|. \tag{10}$$

Robust-adaptive thresholding estimator for covariance matrix of errors is $\hat{\Sigma}_u^{Th}$, which is a $J\times J$ matrix, and the (j,k)-th element in that matrix is

$$\hat{\sigma}_{j,k}^{Th} = \hat{\sigma}_{j,k}\mathbb{1}_{\{|\hat{\sigma}_{j,k}|\geq\hat{\theta}_{j,k}\omega_n\}},$$

with rate

$$\omega_n := C\left[\sqrt{\frac{\log J}{n}}+a_n\right]. \tag{11}$$

Note that definition (10) and our robust-adaptive thresholding estimator are different from the error covariance estimator of Fan et al. (2011) given in equation (2.5). Specifically, they use a squared term instead of the absolute value in the definition of $\hat{\theta}_{j,k}$ and their indicator function involves the square-root of $\hat{\theta}_{j,k}$. The main reason lies in the fact that the adaptive thresholding estimator in Fan et al. (2011) is not appropriate for the deep learning structure used in this paper. By relying on the absolute value function in (10), we develop an estimator which is suitable for deep neural networks. At the same time, we can efficiently handle outliers in the innovations.

Define the sparsity pattern in the covariance matrix of the errors as

$$s_n := \max_{1 \leq j \leq J} \sum_{k=1}^{J} \mathbb{1}_{\{\sigma_{j,k} \neq 0\}},$$

where s_n represents the maximum number of nonzero elements across the rows of the error covariance matrix.

We provide our theorem about consistency of the adaptive thresholding estimator for the error covariance matrix, by using the deep learning residuals in factor models.

Theorem 3. *Under Assumptions 1,2,5-7 then with $C_2 > 0, c_1 > 0, c_2 > 0$ both positive constants*

(i)

$$P\left(\|\hat{\Sigma}_u^{Th} - \Sigma_u\|_{l_2} \leq C_2 \omega_n s_n \right) \geq 1 - O(\frac{1}{J^{\min(c_1,c_2)}}).$$

(ii) Furthermore, if $\omega_n s_n = o(1)$ as an additional assumption then with probability at least $1 - O(\frac{1}{J^2})$

$$Eigmin(\hat{\Sigma}_u^{Th}) \geq Eigmin(\Sigma_u)/2,$$

and

$$\|[\hat{\Sigma}_u^{Th}]^{-1} - \Sigma_u^{-1}\|_{l_2} \leq C_2 \omega_n s_n.$$

Remarks.

1. Note that if $\beta > 1/2$, we have $\omega_n = O(\sqrt{r_{n,2}}) = O\left(\left(\frac{\log J}{n}\right)^{1/4}\right)$. Hence, the smoothness coefficient plays a crucial role. The rate may change with $\beta < 1/2$, see Remark 3 in the Remarks on Theorem 2.
2. Theorem 3.1 of Fan et al. (2011) provides a rate of $s_n d \sqrt{\log J/n}$ for their estimator of the error covariance matrix in linear factor models. Hence, with a larger number of factors d, $\beta > 1/2$ and if $d(\log J/n)^{1/4} \to \infty$, our deep learning estimator yields a faster rate.

5 Covariance Matrix Estimator for the Returns

In this section, we show that the covariance matrix of the returns can be estimated consistently based on the sparse deep neural network estimator. We start with specifying the covariance matrix of the returns and the corresponding estimator. In that respect, for each $i = 1, \cdots, n$

$$Y_i = f_0(X_i) + u_i,$$

with $Y_i := (Y_{1,i}, \cdots, Y_{j,i}, \cdots, Y_{J,i})' : J \times 1$,

$$f_0(X_i) := (f_{0,1}(X_i), \cdots, f_{0,j}(X_i), \cdots, f_{0,J}(X_i))' : \quad J \times 1.$$

and $u_i := (u_{1,i}, \cdots, u_{j,i}, \cdots, u_{J,i})' : \quad J \times 1$. We also know the structure

$$f_{0,j}(X_i) = \sum_{m=1}^{d} f_{j,m}(X_{m,i}).$$

The covariance matrix of the returns is given by

$$\Sigma_y := \Sigma^f + \Sigma_u.$$

We define the covariance matrix for the functions of the factors first.

$$\Sigma^f := E[(f_0(X_i) - Ef_0(X_i))(f_0(X_i) - Ef_0(X_i))'],$$

which is a $p \times p$ matrix. The (j,k)-th element of Σ^f is

$$\Sigma^f_{j,k} := E[(f_{0,j}(X_i) - Ef_{0,j}(X_i))(f_{0,k}(X_i) - Ef_{0,k}(X_i))] = E[f_{0,j}(X_i)f_{0,k}(X_i)] - E[f_{0,j}(X_i)]E[f_{0,k}(X_i)].$$

The estimator for the covariance matrix of the function of factors is defined as follows

$$\hat{\Sigma}^f := \frac{1}{n}\sum_{i=1}^{n}(\hat{f}(X_i) - \bar{f}(X_i))(\hat{f}(X_i) - \bar{f}(X_i))',$$

with $\bar{f}(X_i) := \frac{1}{n}\sum_{i=1}^{n}\hat{f}(X_i)$ which is $J \times 1$ vector, and

$$\hat{f}(X_i) := (\hat{f}_1(X_i), \cdots, \hat{f}_j(X_i), \cdots, \hat{f}_J(X_i))' : \quad J \times 1.$$

We obtain the following Lemma that proofs the consistency of the estimate for the covariance matrix of the functions of factors in a non-linear model. As far as we know, this is a new result in the deep learning literature. Moreover, we allow for $J > n$.

Lemma 3. *Under Assumptions 1,2,5-7*

$$\|\hat{\Sigma}^f - \Sigma^f\|_\infty = O_p(a_n).$$

Note that the estimator for the covariance matrix of returns is given by

$$\hat{\Sigma}_y := \hat{\Sigma}^f + \hat{\Sigma}^{Th}_u.$$

We establish the following theorem for the consistency of the covariance matrix of returns based on the sparse deep neural network. To the best of our knowledge this is a novel result in the deep learning literature.

Theorem 4. *Under Assumptions 1,2,5-7*

$$\|\hat{\Sigma}_y - \Sigma_y\|_\infty = O_p(\omega_n) = O_p(a_n).$$

Remarks.

1. Theorem 4 allows for $J > n$, and shows that the rate of convergence is not affected by number of factors d. Hence, it avoids the curse of dimensionality.

2. The sparsity of the error covariance matrix does not play any role in the estimation error, i.e., the estimation error is not affected by s_n.

3. Remark 3 of the Remarks on Theorem 2 applies here as well. With $\beta > 1/2$, and $J = \exp(n^{a_1}), J = a_2 n, J = n^{a_3}, 0 < a_1 < 1, 0 < a_2 \leq C < \infty, a_3 > 0$, we have the rate, with $\omega_n = O(a_n)$

$$\omega_n = O(\sqrt{r_{n_2}}) = O\left(\left(\frac{\log J}{n}\right)^{1/4}\right).$$

4. Theorem 3.2(ii) of Fan et al. (2011) provides the error rates for the covariance matrix of returns, in the linear factor model with an OLS based estimation of the factors, as $\max(\frac{d(\log J)^{1/2}}{n^{1/2}}, \frac{d^2(\log n)^{1/2}}{n^{1/2}})$. Hence, compared to the rate of our deep learning estimator in Remark 3, as long as $d(\log J/n)^{1/4} \to \infty$, $\frac{d^2}{n^{1/4}} \frac{\log n^{1/2}}{\log J^{1/4}} \to \infty$ we achieve a faster rate due to the avoidance of the number of factors in the convergence rate of the estimator of our error covariance matrix.

6 Precision Matrix Estimator for the Returns

In this section, we provide an explicit formula for the precision matrix for the returns. We start with providing three assumptions that are essential for the theoretical elaboration.

Assumption 8.

(i)

$$Eigmax(\Sigma_u) \leq C\delta_n,$$

with $C > 0$ is a positive constant, and $\delta_n \to \infty$ with $n \to \infty$ and $\delta_n/J \to 0$ with $n \to \infty, J \to \infty$.

(ii)

$$0 \leq r_n \leq Eigmin(\Sigma^f) \leq Eigmax(\Sigma^f) \leq c_3 J,$$

with $c_3 > 0$ a positive constant, and $r_n \to 0$, with $n \to \infty$, or $r_n = 0$.

(iii) $\Sigma_u^{-1}\Sigma^f$ is symmetric.

Assumption 9. $J^2 \omega_n s_n \to 0$.

Assumption 8(i) is standard in large $J > n$ asymptotics, and used in the literature, for the matrices $\Sigma_u : J \times J$, and $\Sigma^f : J \times J$. Hence, the maximum eigenvalue of the covariance matrix of errors is growing with J at the rate δ_n but slower than J itself. In that sense we control the rate of the noise. Moreover, the maximum eigenvalue of the covariance matrix of the factors is upper bounded by a multiple of J. This is also an extension of the linear-additive factor models in Fan et al. (2011) to flexible non-linear-additive factor models considered here. The minimum eigenvalue of the covariance matrix of the factors is either zero, or local to zero, reflecting the large dimensional problems resulting in singularity-near-singularity. [4] Assumption 8(iii) is technical in nature and helps us to derive a lower bound on minimum eigenvalue of a specific matrix in Lemma A.6(i). As an example, if Σ_u^{-1} and Σ^f are commuting this assumption is satisfied, see p. 604, Fact 7.6.16 of Bernstein (2018). Also a common commuting type of matrices are block diagonal matrices with conformable blocks, as illustrated in section 0.7.7 of Horn and Johnson (2013). A good example provides a large $J \times J$ block diagonal matrix, where the zero blocks will be of a small dimension compared to the entire matrix, and the zero blocks in each matrix will be at the same position.

[4] We thank Yuan Liao for pointing us the possible nonsingular nature of non-linear factor models in large dimensions.

Assumption 9 restricts $J << n$, unlike in all the results of the previous Theorems in this paper. The main reason is the lower bound on the minimum eigenvalue of covariance matrix of the functions of factors.

Now we provide a formula for the inverse of the sum of two $J \times J$-dimensional square matrices, where $J < n$, A is a nonsingular matrix, and B can be any square matrix

$$(A+B)^{-1} = A^{-1} - A^{-1}B(I + A^{-1}B)^{-1}A^{-1}, \tag{12}$$

which is from p. 349, Fact 3.20.8 of Bernstein (2018). Set $A = \Sigma_u$ and $B = \Sigma^f$. Since $\Sigma_y = \Sigma^f + \Sigma_u$, where Σ_u is a nonsingular matrix, we obtain the following expression for the precision matrix of the returns

$$\Sigma_y^{-1} = \Sigma_u^{-1} - \Sigma_u^{-1}\Sigma^f[I_J + \Sigma_u^{-1}\Sigma^f]^{-1}\Sigma_u^{-1}. \tag{13}$$

It is important to note that even with Σ^f being singular, the inverse, Σ_y^{-1} exists. In the same way, since $\hat{\Sigma}_u^{Th}$ is nonsingular, with probability approaching one as illustrated in Lemma A.7(ii), the estimate for the precision matrix of the returns based on the sparse deep neural network is

$$\hat{\Sigma}_y^{-1} = (\hat{\Sigma}_u^{Th})^{-1} - (\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f[I_J + (\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f]^{-1}(\hat{\Sigma}_u^{Th})^{-1}. \tag{14}$$

In linear factor models as in Fan et al. (2011) or Caner et al. (2022), it is possible to apply the Sherman-Morrison-Woodbury formula which involves inverting a $d \times d$ matrix, where d is the number of factors which is constant. However, in our formula we have the inversion of a $J \times J$ matrix in (14), and $J < n$, $J/n \to 0$ when $n \to \infty$ in Section 6. By using the symmetry of a product of a certain matrix in Assumption 8(iii), and the sparsity Assumption 9 we can invert this $J \times J$ matrix. The details are provided in Lemma A.6. Hence, this proof provides novel results. In the following, we provide our theorem on the consistency of the precision matrix of the returns.

Theorem 5. *Under Assumptions 1,2,5-9*

$$\|\hat{\Sigma}_y^{-1} - \Sigma_y^{-1}\|_{l_2} = O_p(J^2\omega_n s_n) = o_p(1).$$

Remarks.

1. This is a new result that provides a consistent deep learning based non-linear factor model estimate of the precision matrix of the returns. It is important to note that this estimation error does not depend on the number of factors. However, it is affected by the sparsity (s_n), the number of assets J, and the estimation error rate ω_n which is used for the estimator of the error covariance matrix. Hence, we can only allow for $J << n$, but still $J \to \infty$ when $n \to \infty$.

2. We can compare this rate with Theorem 3.2 of Fan et al. (2011). Their rate is $ds_n\sqrt{\log J/n}$, with d denoting the number of factors, which are growing. Since they assume that the maximum eigenvalue of the errors is finite, in their case the quantity $\delta_n = O(1)$, and not diverging. With a diverging δ_n, Lemma B.4(ii) of Fan et al. (2011) changes to the rate $O(\delta_n/J)$ from $O(1/J)$. This results in a precision matrix of returns estimation error rate of $d\delta_n^2 s_n\sqrt{\log J/n}$ in Theorem 3.2(ii) of Fan et al. (2011).

   If $\beta > 1/2$, the rate of our deep learning estimator is $J^2\omega_n s_n = O(J^2 s_n[\frac{\log J}{n}]^{1/4})$. As long as $d\frac{\delta_n^2}{J^2}[\frac{\log J}{n}]^{1/4} \to \infty$, we achieve a better rate since $\frac{d\delta_n^2 s_n\sqrt{\log J/n}}{J^2 s_n[\frac{\log J}{n}]^{1/4}} \to \infty$.

3. Another paper of interest is Caner et al. (2022). They analyze Sharpe-Ratios for large dimensional portfolios via a feasible weighted nodewise regression. They allow for increasing number of factors,

d, but assume sparsity of the precision matrix of errors. In the case of block-diagonal matrices, the sparsity of the covariance matrix of the errors is the same as in the precision matrix of errors. Since their paper also allows for an increasing maximum eigenvalue $\delta_n \to \infty$, their consistency and the rate of the precision matrix of returns are comparable in case of a block-diagonal covariance matrix of errors. Theorem 2 of Caner et al. (2022) has the estimation error rate for the precision matrix for returns as

$$s_n \delta_n^2 d^{5/2} \max(s_n \lambda_n, s_n^{1/2} d^{1/2} \sqrt{\log J/n}),$$

with $\lambda_n = O\left(\max\left[d^2 \bar{s}_n^{1/2} \frac{\log J}{n}, \sqrt{\frac{\log J}{n}}\right]\right)$ as the tuning parameter for lasso in the nodewise regression. In the case of a large number of factors $\lambda_n = O(d^2 \bar{s}_n^{1/2} \frac{\log J}{n})$.

In case of $\beta > 1/2$, our rate is

$$s_n J^2 (\log J/n)^{1/4}.$$

If $d^{9/2} s_n^{3/2} (\log J/n)^{3/4} \frac{\delta_n^2}{J^2} \geq 1$, our deep learning estimator achieves a better rate. This result is plausible with large number of factors, or with a less parsimonious structure in the covariance matrix of the errors.

7 Implementation

In the following, we discuss the implementation details of our sparse deep neural network factor model (DNN-FM) which are crucial for the model performance. In order to optimally exploit the advantages in enhanced model flexibility of the DNN-FM compared to traditional factor models, it is necessary to avoid the overfitting of the DNN-FM which is associated with the rich model parametrization. Following Schmidt-Hieber (2020) and Gu et al. (2021), we adopt different modeling strategies which are commonly used in the literature to counteract any overfitting.

In order to train and validate our model on different sets of data, we split our data into two distinct subsets which correspond to the training and validation dataset, respectively. This data division is essential to reduce the overfitting. More precisely, only the data in the training set is used to estimate the DNN-FM for a specific set of hyperparameters. The validation set serves as proxy for an out-of-sample test of the model and is used to optimize the hyperparameters. In our implementation, we use the first 80% of data for training the model, whereas the remaining 20% are used for the model validation.

Moreover, we make use of regularization techniques to obtain more parsimonious models by reducing the effective number of parameters to be estimated in the neural network. Specifically, we augment the objective function by a l_1-norm penalty on the weights of the neural network which sets elements of the weight matrices to zero. Hence, the l_1-norm penalty induces sparsity in the parameters of the neural network and allows for disregarding uninformative weights. The strength of the penalty is controlled by a tuning parameter, which we select based on the validation set. Furthermore, we use Dropout, introduced by Srivastava et al. (2014) as a second technique to reduce overfitting in the neural network. The Dropout technique randomly disables a prespecified number of units and their corresponding connections in each layer of the neural network during the training process. This reduces the occurrence of complex co-adaptations between the units on the training data. These co-adaptations generally lead to a close adjustment of the neural network to the training data which compromises its ability to generalize to new data that has not been seen during training. Hence, Dropout mitigates this problem and improves the out-of-sample performance. In our implementation, we randomly disable 20% of the units in each layer during the training process. As a third regularization technique, we adopt early stopping. During each step of an iterative optimization algorithm (e.g., stochastic gradient decent (SGD)) the neural network parameters are adjusted such that the fitting error based on the training data is minimized. While the training error decreases in each iteration, this is not true for the validation error which is used as a proxy for the out-of-sample error. Early stopping keeps track of the validation error and terminates the optimization as soon as the validation error starts to increase.

In order to optimize the DNN-FM, we refer to the adaptive moment estimation algorithm (Adam) introduced by Kingma and Ba (2014) which offers an efficient adaptation of the stochastic gradient decent (SGD) algorithm. More precisely, compared to SGD it provides an adaptive learning rate by using the information of the first and second moments of the gradient.

We estimate the covariance matrix of the residuals of the DNN-FM based on our novel thresholding procedure introduced in Section 4. Specifically, we use $\hat{\theta}_{j,k}\omega_n$ as threshold quantity, where $\hat{\theta}_{j,k}$ is specified in (10) and ω_n is set to $3\sqrt{\frac{\log J}{n}}$. This choice for the threshold is a lower bound enhancing the sparsity in the estimator, such that our threshold error covariance matrix estimator is positive semi-definite in each occasion in the simulation studies and the empirical application.[5]

8 Simulation Evidence

In this section, we present simulation evidence on the finite sample properties of the sparse multilayer neural network in estimating linear and non-linear nonparametric regression models as introduced in (1). In addition, we analyze the precision in estimating the covariance and precision matrix of the underlying data Σ_y and Σ_y^{-1}.

8.1 Monte Carlo designs and Models

For our simulation experiments, we consider the following data generating process (DGP) which is used to evaluate our theoretical findings of Theorems 2, 4 and 5

$$y_{j,i} = \sum_{m=1}^{d} \mathbb{1}_{\{m \text{ is even}\}}\beta_{m,j}X_{m,i} + \mathbb{1}_{\{m \text{ is odd}\}}\beta_{m,j}X_{m,i}^2 + u_{j,i}, \tag{15}$$

where $\mathbb{1}_{\{\cdot\}}$ defines an indicator function that is equal to one if the boolean argument in braces is true. Moreover, the coefficients $\beta_{m,j}$ and the explanatory variables $X_{m,i}$ are drawn from the standard normal distribution, for $m = 1, \cdots d$, $j = 1, \cdots, J$ and $i = 1, \cdots, n$.[6]

For the innovations $u_{j,i}$, we consider two different specifications:

1. For the first specification, we draw $u_{j,i}$, for $j = 1, \cdots, J$, $i = 1, \cdots, n$ from the standard normal distribution. Hence, the innovations are cross-sectionally uncorrelated and the corresponding error covariance matrix Σ_u is diagonal.

2. For the second specification, we allow for cross-sectional correlations between the innovations and

[5] We tried different specifications for ω_n which amount to a lower multiplying constant of the rate $\sqrt{\frac{\log J}{n}}$, i.e., a constant smaller than three, and lead in most of the cases to similar results as our final choice for ω_n. However, in rare occasions a lower threshold quantity resulted in a singular threshold error covariance matrix estimator due to a possibly low sparsity pattern in $\hat{\Sigma}_u^{Th}$. Therefore, we fix the constant to three which leads to a well-defined covariance estimator in all simulations and empirical applications.

[6] We run the same study with fixed coefficients $\beta_{m,j}$ and obtain similar results as in the random coefficients design. Therefore, we omitted these simulation results, however they can be obtained from the authors upon request.

generate those according to the following process which is similar to DGP used in Bai and Liao (2016)

$$\begin{aligned} &u_{1,i} = e_{1,i}, \quad u_{2,i} = e_{2,i} + a_1 e_{1,i}, \quad u_{3,i} = e_{3,i} + a_2 e_{2,i} + b_1 e_{1,i}, \\ &u_{j,i} = e_{j,i} + a_{j-1} e_{j-1,i} + b_{j-2} e_{j-2,i} + c_{j-3} e_{j-3,i}, \qquad \text{for } j = 4, \cdots, J, \end{aligned} \tag{16}$$

where $e_{j,i}$ are iid $N(0,1)$, for $j = 1, \cdots, J$, $i = 1, \cdots, n$ and a_j, b_j, c_j are independently drawn from $0.5\ N(0,1)$, for $j = 1, \cdots, J$. Based on (16), the covariance matrix of the innovations Σ_u is a banded matrix.

Compared to the standard static factor model specification, the process in (15) additionally incorporates non-linearities through the term $\beta_{m,j} X_{m,i}^2$. Hence, we expect that the traditional static factor model estimated with principal component analysis may have greater difficulties in capturing these non-linearities compared to our sparse neural network.

In order to verify the robustness of our simulation results, we consider a second Monte Carlo design which relies on a similar DGP as in Farrell et al. (2021). Specifically, we simulate the data according to the following process

$$Y = \alpha' X + \beta' \psi(X) + u,, \tag{17}$$

where X is drawn from $N(0,1)$ and the idiosyncratic innovations follow the process in (16). Moreover, $\psi(\cdot)$ denotes a non-linear transformation function which incorporates second-degree polynomials and pairwise interactions. Hence, this simulation design extends the non-linear influence of X on the response compared to the first design in (15). The coefficient matrices α and β are both of dimension $d \times J$ and drawn from $U(-1,1)$ and $U(-0.5, 0.5)$, respectively. The time dimension n is set to 60, 120 and 240 for all simulations. Moreover, we consider several dimensions for J and d. Specifically, $J \in \{50, 100, 200\}$ and $d \in \{1, 3, 5, 7\}$. The number of replications is 500.

We compare the precision of our deep neural network factor model (DNN-FM) with methods that are commonly used in the literature. The following models are included in the simulation study.

- SFM-POET: The static factor model with observed factors, where the covariance matrix of the idiosyncratic errors is estimated based on the principal orthogonal complement thresholding method (POET) from Fan et al. (2013).
- DNN-FM: Our non-linear nonparametric factor model estimated based on the s-sparse deep neural network.
- L-LW: The linear shrinkage estimator of Ledoit and Wolf (2003).
- NL-LW: The non-linear shrinkage estimator of Ledoit and Wolf (2017).
- SF-NL-LW: The single factor non-linear shrinkage estimator of Ledoit and Wolf (2017).

8.2 Simulation results

The simulation results for the first Monte Carlo design in (15) with uncorrelated innovations u are illustrated in Tables 1 to 3. Table 1 provides the results of the static linear factor model with observed factors (SFM-POET) and our deep learning factor model (DNN-FM) in estimating the unknown true function $\sum_{m=1}^{d} \mathbb{1}_{\{m \text{ is even}\}} \beta_{m,j} X_{m,i} + \mathbb{1}_{\{m \text{ is odd}\}} \beta_{m,j} X_{m,i}^2$ in (15) which connects the factors X with the response variables Y. The linear (L-LW) and non-linear (NL-LW, SF-NL-LW) shrinkage estimators of Ledoit and

Table 1: Simulation results - First Monte Carlo design, uncorrelated errors
Function estimation

d	n	J	SFM-POET	**DNN-FM**	d	n	J	SFM-POET	**DNN-FM**
1	60	50	7.49	3.35	5	60	50	6.70	2.85
	60	100	8.71	4.04		60	100	7.32	3.42
	60	200	8.37	4.70		60	200	9.09	5.47
	120	50	5.53	2.47		120	50	6.88	2.61
	120	100	8.98	3.84		120	100	7.50	3.39
	120	200	11.30	5.28		120	200	8.17	4.60
	240	50	6.60	2.61		240	50	5.95	2.12
	240	100	8.23	3.61		240	100	6.07	2.31
	240	200	10.00	4.54		240	200	7.32	3.67
3	60	50	5.12	2.08	7	60	50	5.87	2.68
	60	100	9.32	4.36		60	100	7.10	3.72
	60	200	9.51	5.33		60	200	8.57	5.75
	120	50	7.42	2.82		120	50	5.96	2.22
	120	100	7.62	3.06		120	100	7.29	3.50
	120	200	7.96	3.88		120	200	6.88	4.43
	240	50	4.77	1.58		240	50	6.30	2.30
	240	100	7.03	2.83		240	100	7.13	2.95
	240	200	7.50	3.73		240	200	7.87	3.43

Note: The quantities in table relate to the error metric used in Theorem 2 and correspond to the maximum difference between the estimated function and true function ($\sum_{m=1}^{d} \mathbf{1}_{\{m \text{ is even}\}} \beta_{m,j} X_{m,i} + \mathbf{1}_{\{m \text{ is odd}\}} \beta_{m,j} X_{m,i}^2$) in (15), for $j = 1, \cdots, J$.

Wolf (2003) and Ledoit and Wolf (2017), respectively, only provide estimates for the covariance and precision matrix of the returns. For this reason, these methods are not included in Table 1.

The quantities in Table 1 relate to the error metric used in Theorem 2 and correspond to the maximum difference between the estimated and true function. The results indicate that our DNN-FM uniformly provides more precise function estimates compared to the SFM-POET. Consequently, the DNN-FM is better suited for measuring non-linear transformations of the observed factors and complex transferring mechanisms between the factors and the observed variables. As predicted by the theory, the estimation error of the DNN-FM gets closer to zero as n increases, e.g., for $d = 7$ and $J = 200$, the error rate decreases from 5.75 ($n = 60$) to 3.43 ($n = 240$). Hence, the DNN-FM is able to consistently estimate the true underlying function. This result is valid for different dimensions of the number of factors d and number of variables J. Moreover, the error rates of SFM-POET are sensitive to an increase in the number of factors and occasionally rise as d gets large. In contrast to that the error rates of the DNN-FM are in most of the cases unaffected by an increase of d, e.g., for $n = 240, J = 200$, the error rate gradually decreases from 4.54 ($d = 1$) to 3.43 ($d = 7$). This result is in line with the theory in Theorem 2.

Table 2 illustrates the simulation results for estimating the true covariance matrix of the data Σ_y based on the first Monte Carlo design with uncorrelated innovations. The quantities in the table refer to the error metric used in Theorem 4 which corresponds to the maximum matrix norm of the difference between the estimated covariance matrix $\hat{\Sigma}_y$ and Σ_y. The results indicate that our DNN-FM offers in most of the cases the most precise covariance matrix estimates compared to the competing approaches. Specifically, it provides the lowest estimation error and converges faster to zero as n increases, e.g., for $d = 7, J = 200$, the error rates gradually decrease from 1.36 ($n = 60$) to 1.13 ($n = 240$). The only exception is given by the case when the data generating process incorporates one factor ($d = 1$). For this case, the linear shrinkage estimator of Ledoit and Wolf (2003) (L-LW) and the non-linear shrinkage estimators of Ledoit and Wolf (2017) (NL-LW, SF-NL-LW) provide precise estimates of the covariance matrix and lead to lower or similar error rates as DNN-FM. The picture changes as soon as the number of factors increases. While the DNN-FM is unaffected

Table 2: Simulation results - First Monte Carlo design, uncorrelated errors
Covariance matrix estimation

d	n	J	SFM-POET	**DNN-FM**	L-LW	NL-LW	SF-NL-LW
1	60	50	2.48	2.04	1.58	1.51	1.58
	60	100	3.10	2.65	1.85	1.77	1.75
	60	200	3.62	3.28	2.23	2.13	2.04
	120	50	2.58	1.99	1.59	1.55	1.64
	120	100	3.21	2.62	1.89	1.86	1.93
	120	200	3.79	3.28	2.20	2.15	2.21
	240	50	2.64	1.78	1.58	1.57	1.62
	240	100	3.28	2.32	1.91	1.89	1.93
	240	200	3.87	2.89	2.21	2.19	2.38
3	60	50	1.59	1.37	1.80	1.66	1.75
	60	100	1.88	1.69	2.14	1.97	2.00
	60	200	2.20	2.12	2.54	2.35	2.33
	120	50	1.64	1.33	1.70	1.62	1.69
	120	100	1.96	1.64	2.00	1.91	2.00
	120	200	2.29	2.05	2.41	2.30	2.35
	240	50	1.69	1.22	1.65	1.61	1.63
	240	100	2.02	1.48	1.96	1.91	1.94
	240	200	2.34	1.82	2.29	2.23	2.32
5	60	50	1.21	1.09	1.80	1.65	1.72
	60	100	1.45	1.33	2.16	1.97	2.01
	60	200	1.72	1.68	2.53	2.30	2.30
	120	50	1.25	1.05	1.68	1.60	1.67
	120	100	1.49	1.29	2.04	1.95	2.01
	120	200	1.74	1.65	2.38	2.26	2.30
	240	50	1.29	0.98	1.61	1.56	1.60
	240	100	1.54	1.18	1.97	1.93	1.98
	240	200	1.80	1.42	2.30	2.24	2.31
7	60	50	1.06	0.95	1.74	1.58	1.68
	60	100	1.24	1.12	2.10	1.91	1.94
	60	200	1.38	1.36	2.43	2.20	2.18
	120	50	1.05	0.92	1.66	1.57	1.63
	120	100	1.23	1.09	1.96	1.87	1.93
	120	200	1.37	1.32	2.33	2.22	2.23
	240	50	1.09	0.89	1.59	1.55	1.58
	240	100	1.27	1.00	1.88	1.84	1.89
	240	200	1.43	1.13	2.25	2.19	2.25

Note: The quantities in the table refer to the error metric used in Theorem 4 which corresponds to the maximum matrix norm of the difference between the estimated covariance matrix $\hat{\Sigma}_y$ and Σ_y. The deep neural network factor model (DNN-FM) is compared to the static factor model with observed factors and POET estimator from Fan et al. (2013) (SFM-POET), the linear shrinkage estimator of Ledoit and Wolf (2003) (L-LW), the non-linear shrinkage estimator (NL-LW) and the single factor non-linear shrinkage estimator (SF-NL-LW) of Ledoit and Wolf (2017).

by increase of d, i.e., the error rates generally decrease, the error rates of the linear and non-linear shrinkage methods occasionally increase. Moreover, the SFM-POET is uniformly outperformed by our DNN-FM. This result reinforces the conclusion that the rigid linear relationship of the SFM-POET is too restrictive for measuring more complex relations as in (15).

Table 3 provides the results of estimating the true data precision matrix Σ_y^{-1}. The quantities in the table represent the error metric of Theorem 5 which measures the spectral norm of the difference between the estimated and true precision matrix. The results indicate that the DNN-FM consistently estimates the true precision matrix as n increases, e.g., for $d = 7, J = 200$, the error rates rapidly decrease from 2.09 ($n = 60$) to 0.52 ($n = 240$) which is in line with the theory in Theorem 5. Moreover, it generally outperforms SFM-POET across different sample size combinations and number of factors.

Table 3: Simulation results - First Monte Carlo design, uncorrelated errors
Precision matrix estimation

d	n	J	SFM-POET	**DNN-FM**	L-LW	NL-LW	SF-NL-LW
1	60	50	1.78	1.72	20.22	1.77	5.64
	60	100	2.16	2.09	40.59	2.02	4.61
	60	200	2.73	2.64	50.98	1.91	5.35
	120	50	1.26	1.16	11.25	1.19	3.71
	120	100	1.50	1.47	33.63	1.57	4.54
	120	200	1.88	1.89	63.20	1.94	4.15
	240	50	1.05	0.97	5.59	1.15	2.62
	240	100	1.19	1.18	14.16	1.58	3.31
	240	200	1.90	1.50	52.83	2.01	4.56
3	60	50	2.15	1.72	15.91	1.83	11.05
	60	100	2.46	1.65	30.12	2.45	13.47
	60	200	3.08	1.98	48.14	3.36	17.12
	120	50	1.62	1.08	11.70	2.04	8.84
	120	100	1.77	1.14	26.22	2.85	11.45
	120	200	2.10	1.38	52.72	3.91	14.70
	240	50	1.45	0.74	7.58	2.20	6.45
	240	100	1.50	0.82	17.14	3.01	8.88
	240	200	1.64	1.01	44.01	3.98	11.53
5	60	50	2.02	1.73	10.04	2.12	14.47
	60	100	2.22	1.62	19.33	2.84	19.65
	60	200	2.93	1.78	32.66	4.29	27.10
	120	50	1.79	1.20	8.97	2.63	11.79
	120	100	1.77	1.04	19.92	3.71	16.34
	120	200	1.95	1.23	36.85	4.83	22.22
	240	50	1.70	0.80	7.09	3.02	8.76
	240	100	1.67	0.60	16.19	4.16	12.64
	240	200	1.66	0.52	35.45	5.40	17.39
7	60	50	2.16	2.33	5.61	2.21	15.44
	60	100	2.27	2.11	13.11	3.16	24.16
	60	200	2.35	2.09	21.52	4.73	32.15
	120	50	1.90	1.65	5.72	3.04	13.79
	120	100	1.94	1.47	14.59	4.40	20.73
	120	200	1.88	1.51	26.84	5.97	28.33
	240	50	1.81	1.11	5.45	3.73	10.44
	240	100	1.81	0.75	13.62	5.25	15.65
	240	200	1.78	0.52	28.45	8.05	22.28

Note: The quantities in the table represent the error metric of Theorem 5 which measures the spectral norm of the difference between the estimated and true precision matrix. The deep neural network factor model (DNN-FM) is compared to the static factor model with observed factors and POET estimator from Fan et al. (2013) (SFM-POET), the linear shrinkage estimator of Ledoit and Wolf (2003) (L-LW), the non-linear shrinkage estimator (NL-LW) and the single factor non-linear shrinkage estimator (SF-NL-LW) of Ledoit and Wolf (2017).

It is important to note that our DNN-FM is not affected by an increase of the number of included factors, as predicted by our theory. Specifically, the error rates are either hardly changing or decreasing as d increases. E.g., for $n = 240$, $J = 200$, the error rate decreases from 1.50 ($d = 1$) to 0.52 ($d = 7$). In contrast to that the precision of the competing approaches is very much affected by an increase of d and leads to rising error rates, e.g., for $n = 240, J = 200$, the error rates of SFM-POET increase from 1.64 ($d = 3$) to 1.78 ($d = 7$). In addition, the results show that L-LW, NL-LW and SF-NL-LW lead to high error rates in terms of the spectral norm. Only for $d = 1$, the non-linear shrinkage estimator (NL-LW) provides similar results as the DNN-FM. Nevertheless, both NL-LW and SF-NL-LW are inconsistent for different specifications of d.

The simulation results for the first Monte Carlo design in (15) with correlated innovations u are illustrated in Tables 4 to 6. Generally, the error rates for all considered approaches are slightly worse compared

Table 4: Simulation results - First Monte Carlo design, correlated errors
Function estimation

d	n	J	SFM-POET	**DNN-FM**	d	n	J	SFM-POET	**DNN-FM**
1	60	50	9.50	4.72	5	60	50	6.85	3.15
	60	100	9.04	5.29		60	100	9.02	5.05
	60	200	10.55	6.97		60	200	9.78	6.49
	120	50	6.42	3.61		120	50	5.87	2.46
	120	100	6.99	4.25		120	100	7.07	3.60
	120	200	11.67	6.72		120	200	7.46	4.63
	240	50	6.45	3.34		240	50	6.39	2.48
	240	100	8.33	4.48		240	100	7.83	3.49
	240	200	9.95	5.48		240	200	7.38	4.07
3	60	50	6.50	3.06	7	60	50	6.03	2.92
	60	100	9.62	4.72		60	100	8.57	4.86
	60	200	7.35	5.03		60	200	8.13	5.49
	120	50	6.51	2.79		120	50	6.84	2.75
	120	100	8.57	3.94		120	100	6.76	3.17
	120	200	8.03	4.57		120	200	7.51	5.06
	240	50	7.00	2.82		240	50	5.40	2.03
	240	100	7.63	3.41		240	100	6.56	3.06
	240	200	7.84	4.22		240	200	8.23	4.06

Note: The quantities in table relate to the error metric used in Theorem 2 and correspond to the maximum difference between the estimated function and true function $(\sum_{m=1}^{d} \mathbf{1}_{\{m \text{ is even}\}}\beta_{m,j}X_{m,i} + \mathbf{1}_{\{m \text{ is odd}\}}\beta_{m,j}X_{m,i}^2)$ in (15), for $j = 1, \cdots, J$.

to the first design with uncorrelated innovations which is to be expected due to the more complex data generating process. Nevertheless, the overall conclusion is qualitatively similar to the one that we obtain with uncorrelated innovations. Specifically, the DNN-FM consistently determines the true unknown functional form in (15) and provides consistent estimates for the corresponding covariance and precision matrices as n increases. Specifically, Table 4 shows e.g., for $d = 7, J = 200$ a decrease in the error rates of the DNN-FM from 5.49 to 4.06 as n increases from 60 to 240 which confirms a consistent estimation of the unknown functional form. The results in Tables 4 and 6 lead to a similar conclusion for the covariance and precision matrix estimation. More precisely, for $d = 7, J = 200$, and an increase of n from 60 to 240, the error rates of the DNN-FM diminish from 1.33 to 1.10 and 2.01 to 0.81, for the covariance and precision matrix estimators, respectively. Moreover, the error rates of the DNN-FM are generally unaffected by an increase of the number of included factors d. Especially, for the precision matrix estimation this is not the case for the competing approaches. Specifically, Table 6 shows e.g., for $n = 240, J = 100$, an increase in the error rates of SFM-POET from 1.45 ($d = 1$) to 1.76 ($d = 7$) and NL-LW deteriorates from 5.69 ($d = 1$) to 7.80 ($d = 7$). In contrast to that the convergence of the DNN-FM is stable with respect to d, i.e., the error rates are 1.39 for $d = 1$ and decreases to 1.00 for $d = 7$.

Compared to the experiment with uncorrelated innovations, where the NL-LW method lead to similar error rates in the precision matrix estimation as our DNN-FM, for $d = 1$, the precision of NL-LW is largely distorted if the errors contain cross-sectional correlations. It is important to note that the advantage in estimation precision of the DNN-FM compared to the competing approaches is even more pronounced with correlated innovations. This leads to the conclusion that our newly developed robust estimator for the covariance matrix of the innovations, provided in Section 4, is well suited for capturing remaining cross-sectional correlations in the errors.

In order to verify the robustness of the results from the first simulation design, we analyze in the following the Monte Carlo results for the second simulation design in (17) which incorporates linear, as well as non-linear effects and pairwise interactions in X on the dependent variable Y. The results are provided in the

Table 5: Simulation results - First Monte Carlo design, correlated errors
Covariance matrix estimation

d	n	J	SFM-POET	**DNN-FM**	L-LW	NL-LW	SF-NL-LW
1	60	50	2.14	1.74	1.39	1.27	1.31
	60	100	2.59	2.25	1.60	1.52	1.70
	60	200	3.18	2.89	1.92	1.77	1.86
	120	50	2.21	1.70	1.34	1.28	1.29
	120	100	2.68	2.21	1.59	1.53	1.57
	120	200	3.27	2.85	1.93	1.80	1.99
	240	50	2.25	1.54	1.34	1.30	1.31
	240	100	2.73	1.98	1.58	1.54	1.54
	240	200	3.33	2.53	1.93	1.82	1.84
3	60	50	1.45	1.28	1.64	1.60	1.65
	60	100	1.74	1.57	2.01	1.84	1.96
	60	200	2.01	1.95	2.29	2.18	2.40
	120	50	1.53	1.24	1.57	1.53	1.55
	120	100	1.81	1.54	1.89	1.86	1.89
	120	200	2.08	1.89	2.17	2.14	2.25
	240	50	1.57	1.14	1.51	1.51	1.51
	240	100	1.85	1.39	1.84	1.78	1.79
	240	200	2.14	1.68	2.13	2.11	2.13
5	60	50	1.14	1.04	1.67	1.49	1.53
	60	100	1.38	1.27	2.02	1.85	1.88
	60	200	1.57	1.55	2.32	2.17	2.21
	120	50	1.16	1.00	1.58	1.50	1.51
	120	100	1.40	1.24	1.92	1.82	1.84
	120	200	1.60	1.52	2.21	2.15	2.18
	240	50	1.21	0.94	1.55	1.49	1.49
	240	100	1.45	1.12	1.86	1.81	1.80
	240	200	1.64	1.30	2.14	2.14	2.16
7	60	50	1.00	0.92	1.70	1.51	1.56
	60	100	1.17	1.08	2.03	1.85	1.88
	60	200	1.36	1.33	2.44	2.19	2.19
	120	50	1.00	0.90	1.60	1.48	1.49
	120	100	1.18	1.06	1.89	1.84	1.86
	120	200	1.35	1.30	2.31	2.18	2.20
	240	50	1.03	0.85	1.55	1.47	1.48
	240	100	1.21	0.96	1.84	1.82	1.82
	240	200	1.39	1.10	2.22	2.17	2.17

Note: The quantities in the table refer to the error metric used in Theorem 4 which corresponds to the maximum matrix norm of the difference between the estimated covariance matrix $\hat{\Sigma}_y$ and Σ_y. The deep neural network factor model (DNN-FM) is compared to the static factor model with observed factors and POET estimator from Fan et al. (2013) (SFM-POET), the linear shrinkage estimator of Ledoit and Wolf (2003) (L-LW), the non-linear shrinkage estimator (NL-LW) and the single factor non-linear shrinkage estimator (SF-NL-LW) of Ledoit and Wolf (2017).

Tables 7 to 9. The simulation results are qualitatively similar to the ones that we obtain for the first Monte Carlo design. Specifically, Table 7 shows that the DNN-FM consistently estimates the true functional form $\alpha' X + \beta' \psi(X)$ in (17) as n increases. Moreover, it generally provides more precise estimates than the SFM-POET, showing that the DNN-FM is better suited to capture non-linear transformations in the observed factors.

Table 7 illustrates the error rates in terms of the maximum matrix norm of the different approaches in estimating the true covariance matrix. The results are similar to the first simulation design. In fact, the DNN-FM offers consistent estimates of the covariance matrix and uniformly outperforms the SFM-POET for different combinations of d, n and J. Moreover, it leads to more precise estimates compared to the linear and non-linear shrinkage estimators of Ledoit and Wolf (2003), for $d > 1$. For $d = 1$, L-LW achieves a similar

Table 6: Simulation results - First Monte Carlo design, correlated errors
Precision matrix estimation

d	n	J	SFM-POET	**DNN-FM**	L-LW	NL-LW	SF-NL-LW
1	60	50	1.71	1.85	10.41	3.50	12.02
	60	100	1.94	1.99	19.72	2.61	4.75
	60	200	2.15	2.14	32.37	2.15	4.33
	120	50	1.44	1.52	7.07	4.22	6.40
	120	100	1.61	1.69	17.02	6.11	22.51
	120	200	1.77	1.82	37.78	2.88	4.18
	240	50	1.32	1.21	4.47	3.92	4.99
	240	100	1.45	1.39	10.27	5.69	8.52
	240	200	1.71	1.70	30.75	17.82	34.06
3	60	50	2.28	2.15	9.41	4.11	8.26
	60	100	2.33	2.15	18.75	3.98	8.87
	60	200	2.65	2.13	32.05	3.66	9.52
	120	50	1.69	1.76	7.41	5.40	8.19
	120	100	1.83	1.76	16.15	5.57	15.98
	120	200	2.08	1.77	35.47	4.66	9.02
	240	50	1.54	1.43	4.92	5.32	6.87
	240	100	1.65	1.52	11.09	8.65	10.91
	240	200	1.81	1.64	29.22	7.55	24.69
5	60	50	2.04	2.17	6.92	4.53	8.72
	60	100	2.29	1.99	13.75	4.80	11.32
	60	200	2.64	1.89	23.30	4.73	13.17
	120	50	1.69	1.56	5.80	5.84	8.74
	120	100	1.84	1.53	13.23	5.75	11.82
	120	200	1.99	1.44	27.06	5.66	12.11
	240	50	1.62	1.24	4.41	6.19	7.82
	240	100	1.68	1.29	10.02	7.76	11.36
	240	200	1.74	1.23	24.28	7.09	15.67
7	60	50	2.17	2.37	5.36	5.84	9.07
	60	100	2.26	2.14	10.60	5.60	12.63
	60	200	2.62	2.01	16.78	5.72	15.43
	120	50	1.83	1.70	4.95	6.42	9.42
	120	100	1.84	1.55	10.74	6.54	11.71
	120	200	1.96	1.59	20.61	6.62	14.19
	240	50	1.74	1.23	3.95	6.58	8.37
	240	100	1.76	1.00	9.10	7.80	11.24
	240	200	1.75	0.81	20.12	10.50	17.64

Note: The quantities in the table represent the error metric of Theorem 5 which measures the spectral norm of the difference between the estimated and true precision matrix. The deep neural network factor model (DNN-FM) is compared to the static factor model with observed factors and POET estimator from Fan et al. (2013) (SFM-POET), the linear shrinkage estimator of Ledoit and Wolf (2003) (L-LW), the non-linear shrinkage estimator (NL-LW) and the single factor non-linear shrinkage estimator (SF-NL-LW) of Ledoit and Wolf (2017).

precision as our DNN-FM and in most cases offers better performance compared to the remaining competing methods. Furthermore, we observe that the error rates of the DNN-FM are not affected if the number of factors increases which is in accordance with our theory. The simulation results for estimating the precision matrix are outlined in Table 9. The error rates in the terms of the spectral norm are in most of the cases lower for the DNN-FM, followed by the SFM-POET. Moreover, the DNN-FM provides consistent estimates of the precision matrix. Compared to the first simulation design there is a slight difference concerning the sensitivity to an increase of the number of factors. Specifically, the error rates of the DNN-FM slightly rise as d increases, especially if the number of temporal observations is small. This might be due to the DGP used for the second simulation design in (17) which by incorporating pairwise interactions in X is not perfectly conforming with the model setting of the DNN-FM. Moreover, the sensitivity to an increase in d vanishes as

Table 7: Simulation results - Second Monte Carlo design, correlated errors
Function estimation

d	n	J	SFM-POET	**DNN-FM**	d	n	J	SFM-POET	**DNN-FM**
1	60	50	6.21	5.32	5	60	50	5.39	3.87
	60	100	7.94	7.06		60	100	6.30	4.58
	60	200	9.06	8.83		60	200	6.72	5.81
	120	50	5.97	5.31		120	50	5.30	3.54
	120	100	6.94	6.28		120	100	5.75	4.54
	120	200	8.51	8.34		120	200	6.99	6.02
	240	50	5.64	4.97		240	50	4.88	3.22
	240	100	7.23	6.40		240	100	5.74	4.35
	240	200	8.62	8.19		240	200	6.75	6.16
3	60	50	5.58	4.07	7	60	50	5.60	4.03
	60	100	6.35	5.20		60	100	6.77	5.16
	60	200	8.01	7.09		60	200	7.77	6.79
	120	50	5.10	3.70		120	50	4.98	3.38
	120	100	6.61	5.15		120	100	5.78	4.57
	120	200	7.11	6.19		120	200	7.19	6.47
	240	50	4.99	3.61		240	50	4.80	3.29
	240	100	5.81	4.51		240	100	5.63	4.35
	240	200	6.89	6.04		240	200	7.06	6.19

Note: The quantities in table relate to the error metric used in Theorem 2 and correspond to the maximum difference between the estimated function $\hat{\alpha}'X + \hat{\beta}'\hat{\psi}(X)$ and true function $\alpha'X + \beta'\psi(X)$ in (17).

soon as the relative difference between the number of variables J and d gets larger (i.e., $n = 240, J = 200$) which is in line with our theory. Nevertheless, the estimation of the competing approaches is still largely affected by increase of d.

Table 8: Simulation results - Second Monte Carlo design, correlated errors
Covariance matrix estimation

d	n	J	SFM-POET	**DNN-FM**	L-LW	NL-LW	SF-NL-LW
1	60	50	0.82	0.66	0.71	0.62	0.74
	60	100	0.90	0.76	0.83	0.78	0.94
	60	200	0.97	0.87	0.93	0.96	1.12
	120	50	0.80	0.56	0.55	0.50	0.58
	120	100	0.87	0.64	0.63	0.63	0.71
	120	200	0.94	0.73	0.67	0.80	0.86
	240	50	0.80	0.49	0.43	0.40	0.45
	240	100	0.87	0.57	0.49	0.50	0.57
	240	200	0.93	0.62	0.52	0.64	0.66
3	60	50	0.68	0.62	0.88	0.73	0.83
	60	100	0.75	0.67	1.06	0.89	1.03
	60	200	0.85	0.76	1.20	0.99	1.20
	120	50	0.62	0.55	0.77	0.69	0.74
	120	100	0.68	0.60	0.91	0.82	0.89
	120	200	0.75	0.67	1.01	0.91	1.04
	240	50	0.60	0.50	0.71	0.67	0.69
	240	100	0.66	0.54	0.81	0.78	0.81
	240	200	0.73	0.57	0.93	0.87	0.94
5	60	50	0.63	0.59	1.00	0.82	0.91
	60	100	0.70	0.62	1.15	0.95	1.07
	60	200	0.79	0.66	1.32	1.10	1.23
	120	50	0.51	0.53	0.90	0.81	0.85
	120	100	0.58	0.55	1.02	0.92	0.97
	120	200	0.63	0.57	1.16	1.06	1.12
	240	50	0.49	0.49	0.84	0.80	0.82
	240	100	0.55	0.50	0.95	0.90	0.93
	240	200	0.60	0.49	1.09	1.04	1.08
7	60	50	0.59	0.58	0.96	0.78	0.88
	60	100	0.68	0.58	1.16	0.95	1.06
	60	200	0.77	0.59	1.36	1.12	1.22
	120	50	0.47	0.51	0.86	0.77	0.81
	120	100	0.52	0.50	1.04	0.94	0.98
	120	200	0.57	0.51	1.21	1.11	1.15
	240	50	0.43	0.42	0.80	0.76	0.78
	240	100	0.47	0.45	0.96	0.91	0.93
	240	200	0.51	0.43	1.15	1.10	1.12

Note: The quantities in the table refer to the error metric used in Theorem 4 which corresponds to the maximum matrix norm of the difference between the estimated covariance matrix $\hat{\Sigma}_y$ and Σ_y. The deep neural network factor model (DNN-FM) is compared to the static factor model with observed factors and POET estimator from Fan et al. (2013) (SFM-POET), the linear shrinkage estimator of Ledoit and Wolf (2003) (L-LW), the non-linear shrinkage estimator (NL-LW) and the single factor non-linear shrinkage estimator (SF-NL-LW) of Ledoit and Wolf (2017).

Table 9: Simulation results - Second Monte Carlo design, correlated errors
Precision matrix estimation

d	n	J	SFM-POET	**DNN-FM**	L-LW	NL-LW	SF-NL-LW
1	60	50	1.30	1.22	2.50	2.98	5.52
	60	100	1.42	1.30	3.73	1.44	2.00
	60	200	1.48	1.31	4.81	1.21	1.91
	120	50	1.14	0.99	2.16	2.82	3.14
	120	100	1.23	1.07	3.88	7.87	9.63
	120	200	1.27	1.16	6.07	1.46	1.80
	240	50	1.08	0.79	1.58	2.35	2.46
	240	100	1.16	0.88	3.08	3.56	3.91
	240	200	1.20	0.99	6.54	9.70	19.11
3	60	50	1.77	1.88	2.33	2.18	2.99
	60	100	1.88	1.90	3.48	1.97	2.66
	60	200	1.97	1.75	4.64	1.85	2.74
	120	50	1.52	1.53	2.03	2.79	3.09
	120	100	1.64	1.64	3.67	3.10	3.07
	120	200	1.72	1.62	5.90	2.08	2.61
	240	50	1.41	1.29	1.57	2.58	2.77
	240	100	1.53	1.45	3.17	3.88	4.01
	240	200	1.62	1.49	6.66	3.95	5.73
5	60	50	1.95	2.30	2.10	2.59	3.97
	60	100	2.08	2.15	2.75	2.40	2.97
	60	200	2.16	1.88	3.45	2.45	3.16
	120	50	1.60	1.83	1.79	2.78	3.03
	120	100	1.75	1.83	2.80	3.10	3.69
	120	200	1.84	1.67	4.22	2.59	3.05
	240	50	1.46	1.55	1.40	2.48	2.63
	240	100	1.61	1.62	2.59	3.21	3.69
	240	200	1.70	1.50	5.04	3.86	4.13
7	60	50	1.96	2.40	1.94	2.84	3.17
	60	100	2.17	2.21	2.50	2.63	3.13
	60	200	2.24	1.83	3.00	2.82	3.45
	120	50	1.56	1.85	1.69	2.66	2.88
	120	100	1.75	1.81	2.41	5.20	6.90
	120	200	1.84	1.59	3.28	3.17	3.55
	240	50	1.40	1.54	1.33	2.27	2.35
	240	100	1.58	1.58	2.11	3.77	3.96
	240	200	1.68	1.43	3.68	4.19	3.85

Note: The quantities in the table represent the error metric of Theorem 5 which measures the spectral norm of the difference between the estimated and true precision matrix. The deep neural network factor model (DNN-FM) is compared to the static factor model with observed factors and POET estimator from Fan et al. (2013) (SFM-POET), the linear shrinkage estimator of Ledoit and Wolf (2003) (L-LW), the non-linear shrinkage estimator (NL-LW) and the single factor non-linear shrinkage estimator (SF-NL-LW) of Ledoit and Wolf (2017).

9 Empirical Application

In order to verify the practical validity of the DNN factor model, we investigate its efficiency for estimating high-dimensional empirical portfolios.

In an out-of-sample portfolio forecasting experiment, we compare the performance of the global minimum variance portfolio (GMVP) strategy based on the covariance matrix estimated by the sparse multilayer neural network as illustrated in Section 3 with popular alternative portfolio strategies commonly used in the literature. As we are mainly interested in analyzing the empirical quality of the covariance matrix estimator, we concentrate on the estimation of GMVP weights which are solely a function of the covariance matrix of the asset returns.

9.1 Data and illustration of the forecasting experiment

For our analysis we use the excess returns of stocks of the S&P 500 index which were constituents of the index on December 31, 2021. The excess returns are constructed by subtracting the corresponding one-month Treasury bill rate from the asset returns. We compare the forecasting results for asset returns on monthly frequency, where we use the prices available at the end of the corresponding month to calculate the returns.

For our forecasting experiment, we consider two different time periods.

1. Period 1: January 1986 - December 2019: The first sample period yields $n = 407$ monthly return observations and 227 assets are available for the entire sample.

2. Period 2: January 1986 - December 2021: The second period yields $n = 431$ return observations for 227 assets. Compared to the first sample period, the second period incorporates the COVID-19 Crisis 2020/21 and allows us to analyze the effect of this turbulent episode on the forecasting performance of the considered methods.

We conduct an out-of-sample forecasting experiment based on a rolling window approach with an in-sample size of ten years which corresponds to $n_I = 120$ monthly observations. Specifically, at any investment date h, we use for the most recent 120 monthly returns for the estimation. Based on the estimated portfolio weights $\hat{\omega}_h$, we compute the out-of-sample portfolio return for period $h+1$ as $\hat{r}^{pf}_{h+1} = \hat{\omega}'_h r_{h+1}$. In the following step, we shift the in-sample window by one observation, estimate the portfolio weights $\hat{\omega}_{h+1}$ for investment period $h+1$ and compute the out-of-sample portfolio return $\hat{r}^{pf}_{h+1}$. This procedure is repeated until we reach the end of the sample. Hence, for the first period we obtain a series of $n - n_I = 287$ out-of-sample portfolio returns, whereas for the second sample period we retain 311 out-of-sample portfolio returns. All portfolios are updated on a monthly basis.

This result is used to calculate the average out-of-sample portfolio return and variance as follows

$$\hat{\mu} = \frac{1}{n - n_I} \sum_{h=n_I}^{n-1} \hat{r}^{pf}_{h+1} \quad \text{and} \quad \hat{\sigma}^2 = \frac{1}{n - n_I - 1} \sum_{h=n_I}^{n-1} \left(\hat{r}^{pf}_{h+1} - \hat{\mu} \right)^2$$

In order to evaluate the performance of the approaches for different asset dimensions, we consider the following portfolio sizes: $J \in \{50, 100, 200\}$. Specifically, at the investment date h, we select the largest J assets, as measured by their market value which have a complete return history over the most recent n_I periods, similar to Ledoit and Wolf (2017).

In addition, we investigate the portfolio performance in the presence of transaction costs. Following Li

(2015), we calculate the out-of-sample excess returns with transaction costs according to

$$\hat{r}^{pf,tc}_{h+1} = \hat{\omega}'_h r_{h+1} - c\left(1 + \hat{\omega}'_h r_{h+1}\right) \sum_{j=1}^{J} |\hat{\omega}_{h+1,j} - \hat{\omega}^{+}_{h,j}|,$$

where $\hat{\omega}^{+}_{h,j} = \hat{\omega}_{h,j}(1 + r_{h+1,j})/(1 + r^{pf}_{h+1})$ is the portfolio weight of the j-th asset before rebalancing and c are the proportional transaction costs which we set to 50 basis points per transaction as adopted by DeMiguel et al. (2007). Based on the previous definitions, we define the mean and variance of the excess portfolio returns with transaction costs as

$$\hat{\mu}^{tc} = \frac{1}{n - n_I} \sum_{h=n_I}^{n-1} \hat{r}^{pf,tc}_{h+1} \quad \text{and} \quad \hat{\sigma}^{2,tc} = \frac{1}{n - n_I - 1} \sum_{h=n_I}^{n-1} \left(\hat{r}^{pf,tc}_{h+1} - \hat{\mu}^{tc}\right)^2.$$

Moreover, we determine the portfolio turnover by

$$\text{PT} = \frac{1}{n - n_I} \sum_{h=n_I}^{n-1} \sum_{j=1}^{J} |\hat{\omega}_{h+1,j} - \hat{\omega}^{+}_{h,j}|.$$

To evaluate the portfolio performance of each method, we concentrate on the annualized out-of-sample standard deviation (SD), average return (AV) and Sharpe ratio (SR). Moreover, we analyze the average out-of-sample Portfolio Turnover (PT). SD corresponds to $\hat{\sigma}, \hat{\sigma}^{tc}$, AV is $\hat{\mu}, \hat{\mu}^{tc}$, denoting the out-of-sample results without transaction costs, and with transaction costs, respectively. SR is calculated by AV/SD.

In the following, we provide an overview of the approaches which we incorporate in the empirical study:

- EW: The equally weighted portfolio.
- DNN-FM: Our non-linear s-sparse deep neural network factor model. In order to implement our method, we use the Fama-French three factors by Fama and French (1993) as observed factors.
- POET: The principal orthogonal complement thresholding covariance matrix estimator from Fan et al. (2013), with latent factors and the number of factors is estimated based on the IC_1 criterion by Bai and Ng (2002).
- FF3F: The Fama-French three factor model introduced by Fama and French (1993).
- L-LW: The linear shrinkage estimator of Ledoit and Wolf (2003).
- NL-LW: The non-linear shrinkage estimator of Ledoit and Wolf (2017).
- SF-NL-LW: The single factor non-linear shrinkage estimator of Ledoit and Wolf (2017).

9.2 Out-of-sample portfolio results

The annualized out-of-sample portfolio results for the first sample period which excludes the COVID-19 Crisis 2020/21 are illustrated in Table 10. Our DNN-FM provides the lowest out-of-sample standard deviation if the asset dimension J is smaller than the in-sample size n_I of 120 months. For $J = 200$ the linear and non-linear shrinkage estimators of Ledoit and Wolf (2003), Ledoit and Wolf (2017) lead to slightly lower portfolio standard deviations. Given the results of our theory and the simulation results, this outcome is anticipated, as DNN-FM offers the best performance when $n_I > J$. If we consider the out-of-sample SR without transaction costs, the DNN-FM outperforms the competing approaches for large portfolio dimensions, i.e., $J \geq 100$. For

Table 10: Out-of-sample portfolio application results for the first sample

Period 1: January 1986 - December 2019
Without transaction costs

Model	EW	**DNN-FM**	POET	FF3F	L-LW	NL-LW	SF-NL-LW
$J = 50$							
SD	0.143	0.136	0.143	0.142	0.145	0.146	0.144
AV	0.067	0.075	0.067	0.068	0.081	0.082	0.083
SR	0.464	0.555	0.467	0.476	0.556	0.560	0.576
PT	0.071	0.175	0.075	0.083	0.378	0.374	0.375
$J = 100$							
SD	0.144	0.133	0.145	0.140	0.140	0.141	0.139
AV	0.069	0.077	0.064	0.070	0.057	0.061	0.059
SR	0.479	0.582	0.446	0.503	0.409	0.430	0.426
PT	0.073	0.243	0.121	0.098	0.537	0.516	0.535
$J = 200$							
SD	0.145	0.138	0.142	0.140	0.132	0.134	0.132
AV	0.086	0.117	0.082	0.089	0.091	0.089	0.090
SR	0.592	0.852	0.577	0.636	0.689	0.666	0.680
PT	0.066	0.437	0.235	0.107	0.546	0.483	0.470

Period 1: January 1986 - December 2019
With transaction costs

Model	EW	**DNN-FM**	POET	FF3F	L-LW	NL-LW	SF-NL-LW
$J = 50$							
SD	0.143	0.136	0.143	0.142	0.145	0.146	0.144
AV	0.062	0.065	0.062	0.063	0.058	0.059	0.060
SR	0.435	0.478	0.436	0.441	0.399	0.406	0.420
$J = 100$							
SD	0.144	0.132	0.144	0.140	0.140	0.141	0.139
AV	0.064	0.062	0.057	0.065	0.025	0.029	0.027
SR	0.449	0.472	0.396	0.467	0.179	0.210	0.194
$J = 200$							
SD	0.145	0.137	0.142	0.140	0.132	0.133	0.132
AV	0.082	0.091	0.068	0.082	0.058	0.060	0.061
SR	0.565	0.663	0.476	0.591	0.440	0.448	0.466

Note: The deep neural network factor model (DNN-FM) is compared to the equally weighted portfolio (EW), the POET estimator of Fan et al. (2013) (POET), the three factor model of Fama and French (1993) (FF3F), the linear shrinkage estimator of Ledoit and Wolf (2003) (L-LW), the non-linear shrinkage estimator (NL-LW) and the single factor non-linear shrinkage estimator (SF-NL-LW) of Ledoit and Wolf (2017).

low asset dimensions ($J = 50$) it leads to a similar performance as L-LW, NL-LW and SF-NL-LW in terms of SR. However, it is important to note that the linear and non-linear shrinkage estimators generate the highest portfolio turnovers across the considered approaches. When transaction costs are taken into account, the DNN-FM offers the highest SR for all portfolio dimensions. This is mainly due to high out-of-sample returns and a relative low portfolio turnover which is for low dimensions only slightly higher compared to methods with less complex model structures as FF3F and POET. The general better performance of the DNN-FM compared to FF3F testifies the advantage of measuring non-linear relations in high-dimensional portfolios based on deep neural networks compared to models which solely allow for capturing linear effects.

The annualized results for the second sample period which incorporates the COVID-19 Crisis are reported in Table 11. Overall the results are very similar to the ones obtained for the first sample period without COVID-19 Crisis. However, as anticipated, the recent turbulent period in 2020/21 leads to a general increase

Table 11: Out-of-sample portfolio application results for the second sample

Period 2: January 1986 - December 2021
Without transaction costs

Model	EW	**DNN-FM**	POET	FF3F	L-LW	NL-LW	SF-NL-LW
$J = 50$							
SD	0.147	0.140	0.147	0.145	0.148	0.150	0.148
AV	0.076	0.088	0.078	0.078	0.097	0.099	0.099
SR	0.520	0.632	0.528	0.534	0.654	0.659	0.665
PT	0.072	0.180	0.083	0.084	0.383	0.376	0.377
$J = 100$							
SD	0.147	0.137	0.146	0.143	0.143	0.145	0.142
AV	0.078	0.088	0.076	0.079	0.073	0.078	0.073
SR	0.528	0.638	0.517	0.553	0.509	0.536	0.517
PT	0.073	0.250	0.135	0.101	0.544	0.555	0.569
$J = 200$							
SD	0.150	0.144	0.144	0.144	0.135	0.136	0.135
AV	0.093	0.120	0.085	0.095	0.099	0.098	0.098
SR	0.619	0.836	0.589	0.659	0.735	0.720	0.724
PT	0.067	0.449	0.245	0.112	0.557	0.485	0.471

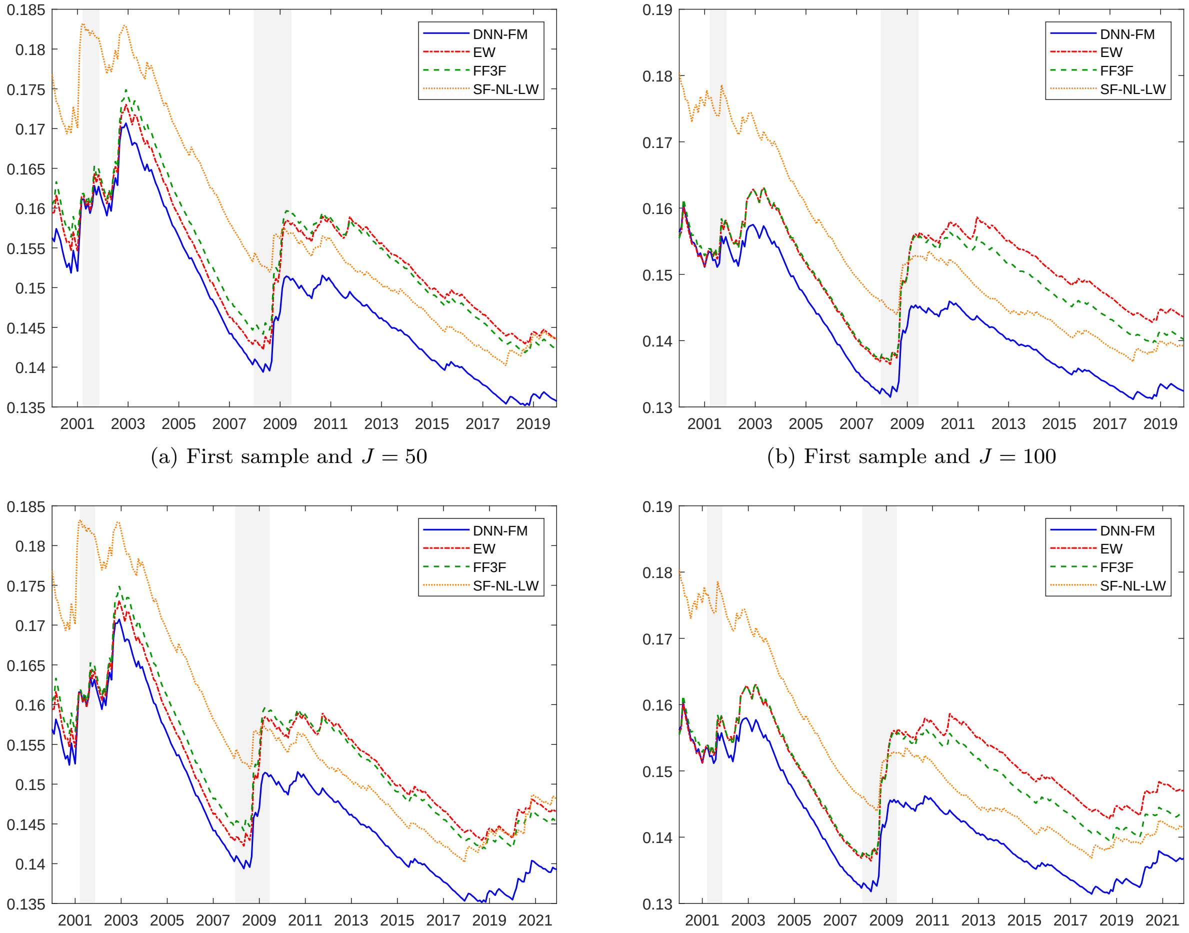
Period 2: January 1986 - December 2021
With transaction costs

Model	EW	**DNN-FM**	POET	FF3F	L-LW	NL-LW	SF-NL-LW
$J = 50$							
SD	0.147	0.139	0.147	0.145	0.148	0.150	0.148
AV	0.072	0.077	0.073	0.073	0.074	0.076	0.076
SR	0.490	0.555	0.495	0.499	0.499	0.508	0.512
$J = 100$							
SD	0.147	0.137	0.146	0.143	0.142	0.143	0.141
AV	0.073	0.072	0.067	0.073	0.040	0.044	0.039
SR	0.498	0.529	0.462	0.511	0.280	0.306	0.275
$J = 200$							
SD	0.150	0.143	0.144	0.143	0.134	0.136	0.135
AV	0.089	0.093	0.070	0.088	0.065	0.069	0.069
SR	0.593	0.650	0.486	0.613	0.487	0.505	0.514

Note: The deep neural network factor model (DNN-FM) is compared to the equally weighted portfolio (EW), the POET estimator of Fan et al. (2013) (POET), the three factor model of Fama and French (1993) (FF3F), the linear shrinkage estimator of Ledoit and Wolf (2003) (L-LW), the non-linear shrinkage estimator (NL-LW) and the single factor non-linear shrinkage estimator (SF-NL-LW) of Ledoit and Wolf (2017).

in out-of-sample portfolio volatility for all considered methods and portfolio dimensions. Nevertheless, the DNN-FM leads to the lowest SD for $n_I > J$ which is even more pronounced compared to the first sample period. This indicates that the DNN-FM can compensate the large volatility during the COVID-19 Crisis. Moreover, when transaction costs are taken into account it leads to the highest SR for all portfolio dimensions.

In order to verify the robustness of our results and to analyze the evolution of the out-of-sample portfolio standard deviations during crisis periods, we also consider the forecasting results for sub-periods. Specifically, we evaluate the performance of the considered methods for a gradual increase of the out-of-sample period. The results are illustrated in Figure 2, where panels (a) and (b) refer to the first sample without COVID-19 Crisis and panels (c) and (d) correspond to the second sample including the COVID-19 Crisis. The outcome at a specific period t incorporates the out-of-sample returns until t (e.g., the SD in December 2011 incorporates the out-of-sample returns from January 1996 until December 2011). The graphs illustrate that



(a) First sample and $J = 50$

(b) First sample and $J = 100$

(c) Second sample and $J = 50$

(d) Second sample and $J = 100$

Figure 2: SD for different sub-periods

the advantage of the DNN-FM in lowest portfolio SD compared to the competing approaches are especially pronounced during and after the Great Financial Crisis 2008/09 and the COVID-19 Crisis 2020/21. These results confirm that our deep learning method is well suited for capturing high volatilities during turbulent episodes. We chose FF3F instead of POET for the linear factor model method, and SF-NL-LW for the shrinkage methods instead of L-LW and NL-LW, since FF3F and SF-NL-LW generally performed better among the competing methods in terms of SD in Tables 10 and 11.

10 Conclusions

In this paper, we contribute to the theoretical understanding of the advantages in predictive power of modern deep neural networks. Specifically, we analyze the large sample properties of feedforward multilayer neural networks (multilayer perceptron) in multivariate nonparametric regression models. Our theoretical elaboration generalizes and extends the deep learning results of Schmidt-Hieber (2020). In particular, we relax the distributional assumption on the innovations in Schmidt-Hieber (2020) from Gaussian errors to subgaussian noise which enhances the scope of our theoretical results in deep neural networks. Moreover, we provide uniform results on the expected estimation risk that are novel to the deep learning literature.

We adopt the deep neural network to an additive model setting and apply the framework to factor models for finance applications. The deep neural network factor model (DNN-FM) allows to measure flexible linear and non-linear interactions between the considered time series and underlying factors. We develop a novel data-dependent estimator of the covariance matrix of the residuals in the DNN-FM which is necessary to construct a robust estimator for the data covariance matrix. The estimator is based on a flexible adaptive thresholding method which is robust to outliers in the innovations. We prove that the adaptive thresholding estimator is consistent under the l_2-norm. In addition, we provide consistency results of the corresponding covariance and precision matrix estimators based on the sparse DNN-FM. It is important to note that the convergence rates of the estimators are unaffected by the number of included factors. Hence, compared to the traditional factor models that impose a rigid linear relationship between the factors and considered variables, our deep neural network factor model considerably enhances the model flexibility.

In the Monte Carlo study, we analyze the finite sample properties of our DNN-FM using various simulation designs. The results confirm our large sample findings. Specifically, the DNN-FM consistently estimates the true underlying non-linear functional form which connects the observed factors with the considered time series, as the number of temporal periods increases. Moreover, the corresponding covariance and precision matrix estimators based on the DNN-FM are consistent as well. The simulation results further verify that the convergence rates of the DNN-FM estimators are independent of the number of incorporated factors. In contrast to that, competing methods are more sensitive to the design of the data generating process and their error rates are negatively affected if the number of included factors increases.

In an out-of-sample portfolio application, we investigate the efficiency of the DNN-FM in predicting high-dimensional empirical portfolios in a global minimum variance portfolio setting and compare the performance to alternative approaches that are commonly used in the literature. The forecasting results show that our DNN-FM leads to the lowest out-of-sample portfolio standard deviation when the number of periods is larger than the number of assets. At the same time, it provides a low portfolio turnover. This results in the highest out-of-sample Sharpe ratio across different portfolio sizes compared to all alternative estimators when transaction costs are taken into account. Moreover, the results illustrate that the advantage of the DNN-FM in terms of lowest portfolio standard deviation is especially pronounced during volatile periods, such as during the COVID-19 Crisis.

Acknowledgments

For helpful comments on an earlier draft of the paper we would like to thank Yuan Liao.

Appendix

A Proofs

We start with the proof of Theorem 1.

Proof of Theorem 1.

(i) The proof of Theorem 1(i) depends on two crucial steps. Step 1 is an oracle inequality, and Step 2 is the smooth function approximation by s-sparse deep neural networks.

In the following, we outline the steps, and then show the proof. Step 1 is subdivided into two parts. Step 1a will be our oracle inequality with subgaussian noise. Step 1a consists of Lemma A.1-A.3. Our Lemma A.1 is the subgaussian counterpart of Lemma C.1 in the Supplement of Schmidt-Hieber (2020), and our Lemma A.2 provides a bound for the noise term which is described in (A.1) below. Lemma A.2 is the subgaussian counterpart of inequality (II) on p. 10 in the Supplement of Schmidt-Hieber (2020). Our Lemma A.3 is the subgaussian counterpart of Lemma 4 in Schmidt-Hieber (2020). The proofs for gaussian noise in Schmidt-Hieber (2020) do not carry over in a simple way to the subgaussian case, hence we provide Lemma A.1-A.3 here. Step 1b provides an upper bound for the covering numbers for the functions in the sparse neural network. Step 2 is a function approximation result that directly carries over from Schmidt-Hieber (2020) with a minor extension.

STEP 1a. Before we start with the proof, we should note that the main technical issue is to obtain a bound for

$$E\left[\frac{2}{n}\sum_{i=1}^{n} u_{j,i}\hat{f}_j(X_i)\right]. \tag{A.1}$$

We are interested in the following term which will be crucial for the bound in (A.1). Let M_j be a positive integer for each $j = 1, \cdots, J$. For $m = 1, \cdots, M_j$, $j = 1, \cdots, J$

$$\eta_{m,j} := \frac{\sum_{i=1}^{n} u_{j,i}[f_{m,j}(X_i) - f_{0,j}(X_i)]}{n^{1/2}\|f_{m,j}(X_i) - f_{0,j}(X_i)\|_n}, \tag{A.2}$$

which is shown on p. 13 in the Supplement of Schmidt-Hieber (2020). $f_{m,j}(.)$ is defined as an approximation for the estimator, where

$$\|\hat{f}_j(X_i) - f_{m,j}(X_i)\|_\infty \le \delta, \tag{A.3}$$

for $j = 1, \cdots, J, m = 1, \cdots, M_j$, with $\delta > 0$. For our purposes, we will rewrite (A.2) as

$$\eta_{m,j} = \frac{\sum_{i=1}^{n} u_{j,i}[f_{m,j}(X_i) - f_{0,j}(X_i)]}{\sqrt{\sum_{i=1}^{n}(f_{m,j}(X_i) - f_{0,j}(X_i))^2}} = \sum_{i=1}^{n} u_{j,i}\Delta_{m,j}(X_i), \tag{A.4}$$

where

$$\Delta_{m,j}(X_i) := \frac{f_{m,j}(X_i) - f_{0,j}(X_i)}{\sqrt{\sum_{i=1}^{n}[f_{m,j}(X_i) - f_{0,j}(X_i)]^2}},$$

See that, for each m, j

$$\sum_{i=1}^{n}\Delta_{m,j}(X_i)^2 = \frac{\sum_{i=1}^{n}[f_{m,j}(X_i) - f_{0,j}(X_i)]^2}{\sum_{i=1}^{n}[f_{m,j}(X_i) - f_{0,j}(X_i)]^2} = 1. \tag{A.5}$$

Note that given $u_{j,i}$ as zero-mean, with variance 1, and iid subgaussian errors, which are independent of X_i across $i = 1, \cdots, n$, $E\eta_{m,j} = 0$, $var\eta_{m,j} = 1$, for each $m = 1, \cdots, M_j$. Set the Orlicz norm for the errors

$$\max_{1\le j\le J}\|u_{j,i}\|_{\psi_2} = C_\psi < \infty.$$

Now we state the following Lemma. Lemma A.1 is for subgaussian noise, and extends from the Gaussian error assumption in Lemma C.1 of Schmidt-Hieber (2020).

Lemma A.1. *Under Assumptions 1-2, for each $j = 1, \cdots, J$*

$$E\max_{1\le m\le M_j}\eta_{m,j}^2 \le C_1\log M_j + 2,$$

with $C_1 = \max(3, C_\psi^2/c), c > 0$, and c is a positive constant.

Proof of Lemma A.1. Define $Z_j := \max_{1\leq m\leq M_j} \eta^2_{m,j}$. For a given j we have

$$Z_j \leq \sum_{m=1}^{M_j} \eta^2_{m,j}, \tag{A.6}$$

and

$$EZ_j \leq \sum_{m=1}^{M_j} E\eta^2_{m,j} = M_j, \tag{A.7}$$

via zero mean and unit variance for $\eta_{m,j}$. We show the proof for $M_j \geq 4$. The proof for $M_j \leq 3$ is a simple algebraic inequality which does not use the distribution of $u_{j,i}$, and is shown in Schmidt-Hieber (2020), also will be shown at the end of the proof here. Note that for $t > 0$

$$P[\eta^2_{1,j} \geq t] = P[|\eta_{1,j}| \geq \sqrt{t}] = 2P[\eta_{1,j} \geq \sqrt{t}].$$

For any $T_j > 0$

$$\begin{aligned} EZ_j &= \int_0^\infty P[Z_j \geq t]dt \leq T_j + \int_{T_j}^\infty P[Z_j \geq t]dt \\ &\leq T_j + M_j \int_{T_j}^\infty P[\eta^2_{1,j} \geq t]dt, \end{aligned} \tag{A.8}$$

where we use Lemma 1.2.1-Integral identity of Vershynin (2019), since Z_j is nonnegative for the equality in (A.8), and we use the union bound for the last inequality and (A.6). In (A.8) consider by the definition of $\eta_{1,j}$ in (A.4)

$$\int_{T_j}^\infty P[\eta^2_{1,j} \geq t]dt = \int_{T_j}^\infty P[|\eta_{1,j}| \geq \sqrt{t}]dt = \int_{T_j}^\infty P[|\sum_{i=1}^n u_{j,i}\Delta_{1,j}(X_i)| \geq \sqrt{t}]dt. \tag{A.9}$$

Now we use the General Hoeffding inequality, Theorem 2.6.3 of Vershynin (2019), since conditional on X_i $u_{j,i}\Delta_{1,j}(X_i)$ is subgaussian, and $u_{j,i}$ is independent from $\Delta_{1,j}(X_i)$ across i, for a given j. For a positive constant $c > 0$, we have

$$P\left[|\sum_{i=1}^n u_{j,i}\Delta_{1,j}(X_i)| \geq \sqrt{t}\right] \leq 2\exp\left(\frac{-ct}{C_\psi^2 \|\Delta_{1,j}(X_i)\|_2^2}\right) = 2\exp\left(\frac{-ct}{C_\psi^2}\right),$$

by (A.5) for the last equality. Now substitute this last inequality in (A.8)

$$\begin{aligned} EZ_j &\leq T_j + 2M_j \int_{T_j}^\infty \exp\left(\frac{-ct}{C_\psi^2}\right) dt \\ &= T_j + 2M_j \frac{C_\psi^2}{c} \exp\left(\frac{-cT_j}{C_\psi^2}\right) \\ &= \left(\frac{C_\psi^2}{c}\right) \log M_j + 2M_j \frac{C_\psi^2}{c}\frac{1}{M_j} = \frac{C_\psi^2}{c}(\log M_j + 2), \end{aligned} \tag{A.10}$$

where we use $T_j = \frac{C_\psi^2}{c} \log M_j$ for the second equality. Combine (A.10) with the case of $1 \leq M_j \leq 3$ on p. 14 in the proof of Lemma C.1 in the Supplement of Schmidt-Hieber (2020) which is $EZ_j \leq M_j \leq 3\log M_j + 1$

to have the desired result.

Q.E.D.

Now we provide another lemma, that will bound the noise given Lemma A.1. This is an extension of the noise bound (II) on p. 10 in the Supplement of Schmidt-Hieber (2020), from gaussian error to subgaussian error. Note that we set $M_j = N_{n,j}(\delta, \mathcal{F}_j, \|.\|_\infty)$ which are the covering numbers for the s-sparse deep neural network $\mathcal{F}_j$. We condition on $\log M_j \leq n$. The case when $\log M_j \geq n$ will be discussed at the end of proof of Theorem 1. First, we set the in-sample prediction error as:

$$\hat{R}_n(\hat{f}_j(X_i), f_{0,j}(X_i)) := E\left[\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i) - f_{0,j}(X_i))^2\right]. \tag{A.11}$$

The following noise bound is the extension of (II) on p. 10 in the Supplement of Schmidt-Hieber (2020), from gaussian noise to subgaussian noise. The proof technique for Lemma A.2 is the same as in Schmidt-Hieber (2020), given our new Lemma A.1. To simplify the notation we set $N_{n,j} := N_{n,j}(\delta, \mathcal{F}_j, \|.\|_\infty)$.

Lemma A.2. *Under Assumptions 1-2, with* $\log N_{n,j} \leq n$, *for all* $j = 1, \cdots, J$

$$\left|E\left[\frac{2}{n}u_{j,i}\hat{f}_j(X_i)\right]\right| \leq (2 + 2\sqrt{C_1 + 2})\delta + 2\sqrt{\frac{\hat{R}_n(\hat{f}_j(X_i), f_{0,j}(X_i))(C_1 \log N_{n,j} + 2)}{n}}.$$

Proof of Lemma A.2. Note that by (C.5) in the Supplement Appendix of Schmidt-Hieber (2020) we obtain the first inequality below

$$\begin{aligned}\left|E\left[\frac{2}{n}u_{j,i}\hat{f}_j(X_i)\right]\right| &\leq 2\delta + \frac{2}{n^{1/2}}E\left[(\|\hat{f}_j(X_i) - f_{0,j}(X_i)\|_n + \delta)|\eta_{m,j}|\right] \\ &\leq 2\delta + \frac{2}{n^{1/2}}(\hat{R}_n(\hat{f}_j(X_i), f_{0,j}(X_i))^{1/2} + \delta)\sqrt{C_1 \log N_{n,j} + 2}, \end{aligned} \tag{A.12}$$

where for the second inequality we use the Cauchy-Schwartz inequality, with $E\eta_{m,j}^2 \leq E\max_{1\leq m\leq M_j}\eta_{m,j}^2$, and Lemma A.1. Note that since we assume $\log N_{n,j} \leq n$,

$$\frac{2}{n^{1/2}}\delta\sqrt{C_1 \log N_{n,j} + 2} \leq 2\delta\sqrt{C_1 + 2}. \tag{A.13}$$

Then by (A.12) and (A.13) we obtain

$$\left|E\left[\frac{2}{n}u_{j,i}\hat{f}_j(X_i)\right]\right| \leq (2 + 2\sqrt{C_1 + 2})\delta + 2\sqrt{\frac{\hat{R}_n(\hat{f}_j(X_i), f_{0,j}(X_i))(C_1 \log N_{n,j} + 2)}{n}}. \tag{A.14}$$

Q.E.D.

Now we provide an oracle inequality which is the extension of Lemma 4 of Schmidt-Hieber (2020) from Gaussian noise to subgaussian noise. Moreover, we extend the result to maximum of J functions.

Lemma A.3. *Under Assumptions 1,2, and*

$$\{f_{0,j}(.)\} \cup \mathcal{F}_j \subset \{f_j(.) : [0,1]^d \to [-F, F]\}, \textit{ for some} \quad F \geq 1,$$

with covering numbers

$$\mathcal{N}_{n,j} \geq 3 \quad \textit{for all } \delta \in (0,1],$$

then for each $j = 1, \cdots, J$

$$\max_{1\leq j\leq J} R(\hat{f}_j, f_{0,j}) \leq 4 \left[\max_{1\leq j\leq J} \inf_{f_{0,j}\in\mathcal{F}_j} E[(f_j(X) - f_{0,j}(X))^2] \right.$$
$$\left. + \frac{4F^2}{n}[C_2 \max_{1\leq j\leq J} \log N_{n,j} + 74] + C_3\delta F \right],$$

for constants $C_2 \geq 18, C_3 \geq 58$. *These last two constants are derived from the constant* $C_1 \geq 3$, *and explained in the proof.*

Remark. Note that in subgaussian case of Lemma 4 of Schmidt-Hieber (2020), he has 18 in front of the covering numbers, since $C_1 \geq 3$ we have $C_2 := 2C_1 + 12 \geq 18$. Our second constant in the second term is 74 and larger than 72 in Lemma 4 of Schmidt-Hieber (2020), so there is a price to pay for a more general result in our case, but these differences will not matter when $n \to \infty$. Also in front of δF, Lemma 4 of Schmidt-Hieber (2020) has 32 as a contant, and we have at least 58. Clearly the Orlicz norm plays a key role in our bound, since $C_1 = \max(3, \frac{C_\psi^2}{c})$, and C_ψ represents maximum of the Orlicz norm (ψ_2) of the noise. Also we impose $\epsilon = 1$ in Lemma 4 of Schmidt-Hieber (2020) to simplify the notation.

Proof of Lemma A.3. Our proof of Lemma A.3 has the structure of (III) on p. 11 in the Supplement of Schmidt-Hieber (2020). The main difference arises from our results in Lemma A.1-A.2. We start with the case of $\log N_{n,j} \leq n$, for $j = 1, \cdots, J$. The other case will be discussed at the end. By the definition of $\hat{f}_j(.)$

$$E\left[\frac{1}{n}\sum_{i=1}^{n}(Y_{j,i} - \hat{f}_j(X_i))^2\right] \leq E\left[\frac{1}{n}\sum_{i=1}^{n}(Y_{j,i} - f_j(X_i))^2\right]. \tag{A.15}$$

Since $X_i \equiv X$, we have

$$E\|f_j(X_i) - f_{0,j}(X_i)\|_n^2 = E[f_j(X) - f_{0,j}(X)]^2. \tag{A.16}$$

Also by assumption

$$Eu_{j,i}f_j(X_i) = 0. \tag{A.17}$$

By the model, $Y_{j,i}$ in (1)

$$E\left[\frac{1}{n}\sum_{i=1}^{n}(u_{j,i} - (\hat{f}_j(X_i) - f_{0,j}(X_i)))^2\right] \leq E\left[\frac{1}{n}\sum_{i=1}^{n}(u_{j,i} - (f_j(X_i) - f_{0,j}(X_i)))^2\right].$$

Then by rearranging, simplifying and using (A.11), (A.16) and (A.17) we obtain

$$\hat{R}_n(\hat{f}_j(X_i), f_{0,j}(X_i)) \leq E\left[\frac{2}{n}\sum_{i=1}^{n}u_{j,i}\hat{f}_j(X_i)\right] + E[f_j(X) - f_{0,j}(X)]^2. \tag{A.18}$$

By Lemma A.2

$$\begin{aligned}\hat{R}_n(\hat{f}_j(X_i), f_{0,j}(X_i)) &\leq E[f_j(X) - f_{0,j}(X)]^2 \\ &+ (2 + 2\sqrt{C_1 + 2})\delta + 2\sqrt{\frac{\hat{R}_n(\hat{f}_j(X_i), f_{0,j}(X_i))(C_1 \log N_{n,j} + 2)}{n}}.\end{aligned}$$

For positive real numbers, a, b, d set

$$a = \hat{R}_n(\hat{f}_j(X_i), f_{0,j}(X_i)),\ b = \sqrt{(C_1 \log N_{n,j} + 2)/n},$$

$$d = E[f_j(X) - f_{0,j}(X)]^2 + (2 + 2\sqrt{C_1 + 2})\delta.$$

If $|a| \leq 2\sqrt{a}b + d$ then $a \leq 2d + 4b^2$ which implies, with $F \geq 1$

$$\hat{R}_n(\hat{f}_j(X_i), f_{0,j}(X_i)) \leq 2\left[\inf_{f_j \in \mathcal{F}_j} E[f_j(X) - f_{0,j}(X)]^2\right] + (2 + 2\sqrt{C_1 + 2})2\delta + F^2 \frac{4(C_1 \log N_{n,j} + 2)}{n}. \quad \text{(A.19)}$$

Next, we use the upper bound in the inequality (I) on p. 10 in the Supplement of Schmidt-Hieber (2020), which uses only the iid structure of the data Y_i, X_i across i, but relates the risk to its empirical version

$$R(\hat{f}_j(X), f_{0,j}(X)) \leq 2\left[\hat{R}_n(\hat{f}_j(X_i), f_{0,j}(X_i)) + \frac{2F^2}{n}(12 \log N_{n,j} + 70) + 26\delta F\right].$$

Taking maximum over $j = 1, \cdots, J$ and using (A.19) we get

$$\begin{aligned}\max_{1 \leq j \leq J} R(\hat{f}_j(X), f_{0,j}(X)) &\leq 4[\max_{1 \leq j \leq J} \inf_{f_j \in \mathcal{F}_j} E[f_j(X) - f_{0,j}(X)]^2] \\ &+ [26 + (8 + 8\sqrt{C_1 + 2})]\delta F + 4F^2 \frac{(12 + 2C_1)\max_{1 \leq j \leq J} \log N_{n,j} + 74}{n}.\end{aligned}$$

Set $C_2 := 12 + 2C_1, C_3 \geq 58 \geq 26 + (8 + 8\sqrt{C_1 + 2})$, with $C_1 \geq 3$. (Also we have an empirical minimizer as estimator so in the proof of Lemma 4 of Schmidt-Hieber (2020), his term $\Delta_n = 0$).

Case of $\log N_{n,j} \geq n$ is shown in Schmidt-Hieber (2020) and will be repeated here with added uniformity over j. Since

$$\max_{1 \leq j \leq J} R(\hat{f}_j(X), f_{0,j}(X)) \leq 4F^2,$$

upper bound here follows in that case.

Q.E.D.

Step 1b. Next, we need to find upper bounds on the covering numbers. First, Lemma 5 of Schmidt-Hieber (2020) with Remark 1 and proof of Theorem 2 in Schmidt-Hieber (2020) puts an upper bound on $\log N_{n,j}$ in our Lemma A.3 and it is not related to $u_{j,i}$ data distribution, and the upper bound in Theorem 2 of Schmidt-Hieber (2020) is the same here. Let $C_4 > 0$ be a positive constant

$$\begin{aligned}\max_{1 \leq j \leq J} \log N_{n,j} &\leq C_4 \max_{1 \leq j \leq J}\left[(s_j + 1)\log(n(s_j + 1)^L d)\right] \\ &= C_4(\bar{s} + 1)\log(n(\bar{s} + 1)^L d),\end{aligned}$$

where we use $p_0 = d, p_{L+1} = 1$ in our notation, where p_0 is the input dimension, and p_{L+1} is the output dimension in Schmidt-Hieber (2020). We extend the result of Schmidt-Hieber (2020) to uniform over $j =$

$1, \cdots, J$ functions. Then

$$\begin{aligned} \frac{\max_{1 \leq j \leq J} \log N_{n,j}}{n} &\leq \frac{C_4(\bar{s}+1)\log[n(\bar{s}+1)^L d]}{n} = \frac{C_4(\bar{s}+1)\log[\frac{n^L}{n^{L-1}}(\bar{s}+1)^L \frac{d^L}{d^{L-1}}]}{n} \\ &= \frac{C_4(\bar{s}+1)L\log[\frac{n}{n^{1-1/L}}(\bar{s}+1)\frac{d}{d^{1-1/L}}]}{n}. \end{aligned} \tag{A.20}$$

Then by Assumption 4, $\bar{s} \leq C_4 n\phi_n \log n$ we have

$$\bar{s}+1 \leq C' n\phi_n \log n. \tag{A.21}$$

where C' is another positive constant. Next, we have

$$n^{\frac{1}{L}}\phi_n \log n = n^{\frac{1}{L}} n^{-2\beta_h^*/(2\beta_h^*+t_h)} \log n \to 0,$$

by Assumption 4, $L \geq \sum_{i=0}^{h} log_2(4t_h \cup 4\beta_h) log_2 n \geq 2$. Since d is constant and using (A.21) and the last result above, with a large positive constant C'',

$$\log[\frac{n}{n^{1-1/L}}(\bar{s}+1)\frac{d}{d^{1-1/L}}] \leq \log[(n)C' n^{1/L} d^{1/L}\phi_n \log n] \leq C'' \log n. \tag{A.22}$$

So substitute (A.21)-(A.22) into (A.20)

$$\frac{\max_{1 \leq j \leq J} \log N_{n,j}}{n} \leq C'' L\phi_n \log^2 n. \tag{A.23}$$

Step 2. Next, we consider function approximation. Since

$$\inf_{f_j \in \mathcal{F}_j} E[f_j(X) - f_{0,j}(X)]^2] \leq \inf_{f_j^* \in \mathcal{F}_j} \|f_j^*(X) - f_0(X)\|_\infty^2,$$

by (26) of Schmidt-Hieber (2020) with C''' a positive constant

$$\inf_{f_j^* \in \mathcal{F}_j} \|f_j^*(X) - f_0(X)\|_\infty^2 \leq C'''\phi_n,$$

we get

$$\max_{1 \leq j \leq J} \inf_{f_j^* \in \mathcal{F}_j} \|f_j^*(X) - f_0(X)\|_\infty^2 \leq C'''\phi_n, \tag{A.24}$$

since right hand side of the previous inequality, the one before (A.24) does not depend on j. Now combine (A.23)(A.24) in Lemma A.3 to have the desired result, with $\delta = 1/n$, and $F \geq 1$ and function have same smoothness parameters and dimensions of β_h, t_h regardless of $j = 1, \cdots, J$. C replaces C'', C'''.

(ii) Apply (A.20)-(A.24) to the right side of (A.19) to get the desired result, or by using X_i being iid, and $X \equiv X_i$ for all $i = 1, \cdots, n$, and Theorem 1(i)

$$\max_{1 \leq j \leq J} E\left[\frac{1}{n}\sum_{i=1}^{n}[\hat{f}_j(X_i) - f_{0,j}(X_i)]^2\right] = \max_{1 \leq j \leq J} E\left[[\hat{f}_j(X) - f_{0,j}(X)]^2\right] \leq C\phi_n L \log^2 n.$$

Q.E.D.

Proof of Theorem 2. This is not a trivial extension from Theorem 1 since we have J functions to estimate, and it is not obvious which concentration inequality and how to use them in the proof. Here our maximal inequality result in (A.28) can also apply to other estimation problems in high dimensions.

Since for all $i = 1, \cdots, n, j = 1, \cdots, J$

$$|\hat{f}_j(X_i) - f_{0,j}(X_i)| \leq 2F,$$

we have by (2.11) of Wainwright (2019) or one-sided version of p. 454 of Wainwright (2019), with $t > 0$

$$P\left(\frac{1}{n}\sum_{i=1}^{n}[\hat{f}_j(X_i) - f_{0,j}(X_i)]^2 - E[\hat{f}_j(X_i) - f_{0,j}(X_i)]^2 \geq t\right) \leq \exp(\frac{-nt^2}{32F^4}).$$

Using the union bound, with $t_1 > 0$

$$P\left(\max_{1\leq j\leq J}\left[\frac{1}{n}\sum_{i=1}^{n}[\hat{f}_j(X_i) - f_{0,j}(X_i)]^2 - E[\hat{f}_j(X_i) - f_{0,j}(X_i)]^2\right] \geq t_1\right) \leq \exp(\log J - \frac{-nt_1^2}{32F^4}). \qquad \text{(A.25)}$$

Choose $t_1 = (\sqrt{32}F^2)\sqrt{t^2 + \log J/n}$ with choice of $t = c_1\sqrt{\log J/n}$ in (A.25)

$$P\left(\max_{1\leq j\leq J}\left[\frac{1}{n}\sum_{i=1}^{n}[\hat{f}_j(X_i) - f_{0,j}(X_i)]^2 - E[\hat{f}_j(X_i) - f_{0,j}(X_i)]^2\right] \geq \sqrt{32}F^2\sqrt{(c_1^2+1)\log J/n}\right) \leq \frac{1}{J^{c_1^2}}. \qquad \text{(A.26)}$$

Then note that by defining $Z_j := \sum_{i=1}^{n}[\hat{f}_j(X_i) - f_{0,j}(X_i)]^2, Z_j \geq 0$, we see that

$$\max_{1\leq j\leq J}[Z_j - EZ_j] \geq \max_{1\leq j\leq J}[Z_j - \max_{1\leq j\leq J} EZ_j] = \max_{1\leq j\leq J} Z_j - \max_{1\leq j\leq J} EZ_j. \qquad \text{(A.27)}$$

(A.27) implies that we can change the left side term of the probability in (A.26) and get the following maximal inequality

$$P\left(\max_{1\leq j\leq J}\frac{1}{n}\sum_{i=1}^{n}[\hat{f}_j(X_i) - f_{0,j}(X_i)]^2 - \max_{1\leq j\leq J}\frac{1}{n}\sum_{i=1}^{n}E[\hat{f}_j(X_i) - f_{0,j}(X_i)]^2 \geq \sqrt{32}F^2\sqrt{(c_1^2+1)\log J/n}\right) \leq \frac{1}{J^{c_1^2}}. \qquad \text{(A.28)}$$

So, use Theorem 1(ii) in the second term on the left side of the probability in (A.28) with $r_{n1} := Cn^{-2\beta/(2\beta+1)}\log^3 n, r_{n2} := \sqrt{32}F^2\sqrt{(c_1^2+1)\log J/n}$ definitions

$$P[\max_{1\leq j\leq J}\frac{1}{n}\sum_{i=1}^{n}[\hat{f}_j(X_i) - f_{0,j}(X_i)]^2 \geq r_{n1} + r_{n2}] \leq \frac{1}{J^{c_1^2}}.$$

Q.E.D.

Proof of Lemma 1.

(i) Since $u_{j,i}$ are independent sub Gaussian and multiple of sub Gaussian random variable are sub exponential $u_{j,i}u_{k,i}$, and centered version is also subexponential by p. 31-32 of Vershynin (2019), then we can benefit from Corollary 2.8.3 of Vershynin (2019), and providing a union bound

$$P\left(\max_{1\leq j\leq J}\max_{1\leq k\leq J}|\frac{1}{n}\sum_{i=1}^{n}u_{j,i}u_{k,i}-Eu_{j,i}u_{k,i}|\geq t\right)\leq J^2 2\exp\left(-c\min(\frac{t^2}{C_\psi^4},\frac{t}{C_\psi^2})n\right),$$

with C as a positive constant. Then the right side upper bound probability simplifies

$$J^2 2\exp\left(-c\min(\frac{t^2}{C_\psi^4},\frac{t}{C_\psi^2})n\right)=\exp\left(\log 2J^2-c\min(\frac{t^2}{C_\psi^4},\frac{t}{C_\psi^2})n\right).$$

Set $t=C\sqrt{\log J/n}$, with sufficiently large n, ($n^{1/2}\geq(C/C_\psi^2)(\log J)^{1/2}$) we have $t^2/C_\psi^4\leq t/C_\psi^2$, we only analyze

$$J^2 2\exp\left(-c\min(\frac{t^2}{C_\psi^4},\frac{t}{C_\psi^2})n\right)=\exp\left(\log 2J^2-c\frac{t^2 n}{C_\psi^4}\right).$$

Set $c_2: cC^2/C_\psi^4-2>0$ to have

$$\log 2+2\log J-c\frac{C^2}{C_\psi^4}\log J=\log(\frac{2}{J^{c_2}}).$$

This provides the desired result.

Q.E.D.

(ii) First, $u_{j,i}u_{k,i}-Eu_{j,i}u_{k,i}$ is subexponential random variables across $i=1,\cdots,n$ by Lemma 2.7.7 and p. 32 of Vershynin (2019). Then $|u_{j,i}u_{k,i}-Eu_{j,i}u_{k,i}|$ is subexponential by Proposition 2.7.1 of Vershynin (2019), and by p. 32 of Vershynin (2019) $|u_{j,i}u_{k,i}-Eu_{j,i}u_{k,i}|-E|u_{j,i}u_{k,i}-Eu_{j,i}u_{k,i}|$ is subexponential. Then apply Lemma 1(i) proof above.

Q.E.D.

Proof of Lemma 2. (i) Clearly, $u_{j,i}-\hat{u}_{j,i}=\hat{f}_j(X_i)-f_{0,j}(X_i)$. Then we simplify the problem as

$$\max_{1\leq j\leq J}\frac{1}{n}\sum_{i=1}^{n}(u_{j,i}-\hat{u}_{j,i})^2=\max_{1\leq j\leq J}\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i)-f_{0,j}(X_i))^2. \tag{A.29}$$

Apply Theorem 2 to get the result.

Q.E.D.

(ii) Use triangle inequality by seeing $\hat{u}_{j,i}=\hat{u}_{j,i}-u_{j,i}+u_{j,i}$, and in the same way for $\hat{u}_{k,i}$, and Cauchy-Schwartz inequality for the second inequality below

$$\begin{aligned}\max_{1\le j\le J}\max_{1\le k\le J}|\frac{1}{n}\sum_{i=1}^n(\hat{u}_{j,i}\hat{u}_{k,i}-u_{j,i}u_{k,i})| &\le \max_{1\le j\le J}\max_{1\le k\le J}|\frac{1}{n}\sum_{i=1}^n(\hat{u}_{j,i}-u_{j,i})(\hat{u}_{k,i}-u_{k,i})|\\ &+ 2\max_{1\le j\le J}\max_{1\le k\le J}|\frac{1}{n}\sum_{i=1}^n u_{j,i}(\hat{u}_{k,i}-u_{k,i})|\\ &\le \max_{1\le j\le J}\frac{1}{n}\sum_{i=1}^n(\hat{u}_{j,i}-u_{j,i})^2\\ &+ 2\sqrt{\max_{1\le j\le J}\frac{1}{n}\sum_{i=1}^n u_{j,i}^2}\sqrt{\max_{1\le j\le J}\frac{1}{n}\sum_{i=1}^n(\hat{u}_{j,i}-u_{j,i})^2}.\end{aligned}$$

Then apply Lemma 1(i), Assumption 1, and Lemma 2(i) to get the result, since $\max_{1\le j\le J} Eu_{j,i}^2 \le C < \infty$, and $a_n^2 + 2Ca_n \le C_* a_n$ with Assumption 7(i).

Q.E.D.

(iii) By triangle inequality

$$\begin{aligned}\max_{1\le j\le J}\max_{1\le k\le J}|\frac{1}{n}\sum_{i=1}^n\hat{u}_{j,i}\hat{u}_{k,i}-Eu_{j,i}u_{k,i}| &\le \max_{1\le j\le J}\max_{1\le k\le J}|\frac{1}{n}\sum_{i=1}^n(\hat{u}_{j,i}\hat{u}_{k,i}-u_{j,i}u_{k,i})|\\ &+ \max_{1\le j\le J}\max_{1\le k\le J}|\frac{1}{n}\sum_{i=1}^n(u_{j,i}u_{k,i}-Eu_{j,i}u_{k,i})|.\end{aligned}$$

Then Lemma 1(i), and Lemma 2(ii) above provides the result with Assumption 7(i) and $C_{**} \ge 3C+1$.

Q.E.D.

The proof of Theorem 3 uses the following lemma and this will be used to in robust adaptive thresholding covariance matrix estimator.

Lemma A.4. *Under Assumptions 1,2,5-7, with $c_1 > 0, c_2 > 0$ both positive constants*

$$P\left(C_L \le \min_{1\le j\le J, 1\le k\le J}\hat{\theta}_{j,k} \le \max_{1\le j\le J, 1\le k\le J}\hat{\theta}_{j,k} \le C_u\right) \ge 1 - O(\frac{1}{J^{\min(c_1,c_2)}}),$$

with $C_U := \max_{1\le j\le J,1\le k\le J}|\sigma_{j,k}| + \max_{1\le j\le J}\sigma_{j,j} < \infty$, and $C_L := \min_{1\le j\le J,1\le k\le J} E|u_{j,i}u_{k,i}-\sigma_{j,k}| = c > 0$.

Proof of Lemma A.4.

Define the following events, that we condition our proof, and $C'' > 0$ is a positive constant

$$\mathcal{A}_1 := \{\max_{1\le k\le J}\max_{1\le j\le J}|\hat{\sigma}_{j,k}-\sigma_{j,k}| \le C''(\sqrt{\frac{\log J}{n}} + a_n)\},$$

$$\mathcal{A}_2 := \{\max_{1\le j\le J}\frac{1}{n}\sum_{i=1}^n(\hat{u}_{j,i}-u_{j,i})^2 \le a_n^2\}.$$

$$\mathcal{A}_3 := \{\max_{1\le j\le J}|\frac{1}{n}\sum_{i=1}^n u_{j,i}^2 - \sigma_{j,j}| \le C\sqrt{\log J/n}\}.$$

$$\mathcal{A}_4 := \{ \max_{1\le j\le J} \max_{1\le k\le J} \left| \frac{1}{n}\sum_{i=1}^n |u_{j,i}u_{k,i} - \sigma_{j,k}| - E|u_{j,i}u_{k,i} - \sigma_{j,k}| \right| \le C\sqrt{\frac{\log J}{n}} \}.$$

and then we will relax the condition and get the probabilistic result. We start with the upper bound. By using the triangle inequality

$$\begin{aligned} \max_{1\le j\le J}\max_{1\le k\le J} \hat{\theta}_{j,k} &= \max_{1\le j\le J}\max_{1\le k\le J} \frac{1}{n}\sum_{i=1}^n |\hat{u}_{j,i}\hat{u}_{k,i} - \hat{\sigma}_{j,k}| \le \max_{1\le j\le J}\max_{1\le k\le J} \frac{1}{n}\sum_{i=1}^n |\hat{u}_{j,i}\hat{u}_{k,i} - \sigma_{j,k}| \\ &+ \max_{1\le j\le J}\max_{1\le k\le J} \frac{1}{n}\sum_{i=1}^n |\sigma_{j,k} - \hat{\sigma}_{j,k}| \\ &= \max_{1\le j\le J}\max_{1\le k\le J} \frac{1}{n}\sum_{i=1}^n |\hat{u}_{j,i}\hat{u}_{k,i} - \sigma_{j,k}| + \max_{1\le j\le J, 1\le k\le J} |\sigma_{j,k} - \hat{\sigma}_{j,k}|. \end{aligned} \tag{A.30}$$

Then condition on the event $\mathcal{A}_1$, and use it in second term on the right side of (A.30)

$$\max_{1\le j\le J, 1\le k\le J} |\sigma_{j,k} - \hat{\sigma}_{j,k}| \le C''(\sqrt{\log J/n} + a_n) = o(1), \tag{A.31}$$

where the asymptotic negligibility is by Assumption 7. Consider the first term on the right side of (A.30).

$$\begin{aligned} \max_{1\le j\le J}\max_{1\le k\le J} \frac{1}{n}\sum_{i=1}^n |\hat{u}_{j,i}\hat{u}_{k,i} - \sigma_{j,k}| &\le \max_{1\le j\le J}\max_{1\le k\le J} \frac{1}{n}\sum_{i=1}^n |(\hat{u}_{j,i} - u_{j,i})(\hat{u}_{k,i} - u_{k,i})| \\ &+ \max_{1\le j\le J}\max_{1\le k\le J} \frac{1}{n}\sum_{i=1}^n |(\hat{u}_{j,i} - u_{j,i})(u_{k,i})| \\ &+ \max_{1\le j\le J}\max_{1\le k\le J} \sum_{i=1}^n |(u_{j,i})(\hat{u}_{k,i} - u_{k,i})| \\ &+ \max_{1\le j\le J}\max_{1\le k\le J} \frac{1}{n}\sum_{i=1}^n |u_{j,i}u_{k,i} - \sigma_{j,k}|. \end{aligned} \tag{A.32}$$

Consider the first term on the right side of (A.32) via Cauchy-Schwartz inequality

$$\max_{1\le j\le J}\max_{1\le k\le J} \sum_{i=1}^n |(\hat{u}_{j,i} - u_{j,i})(\hat{u}_{k,i} - u_{k,i})| \le \sqrt{\max_{1\le j\le J} \frac{1}{n}\sum_{i=1}^n (\hat{u}_{j,i} - u_{j,i})^2}\sqrt{\max_{1\le k\le J} \frac{1}{n}\sum_{i=1}^n (\hat{u}_{k,i} - u_{k,i})^2} \le a_n^2, \tag{A.33}$$

by event $\mathcal{A}_2$. Consider the second term on the right side of (A.32)

$$\begin{aligned} \max_{1\le j\le J}\max_{1\le k\le J} \frac{1}{n}\sum_{i=1}^n |(\hat{u}_{j,i} - u_{j,i})(u_{k,i})| &\le \sqrt{\max_{1\le j\le J} \frac{1}{n}\sum_{i=1}^n (\hat{u}_{j,i} - u_{j,i})^2}\sqrt{\max_{1\le k\le J} \frac{1}{n}\sum_{i=1}^n u_{k,i}^2} \\ &\le a_n \left\{ \max_{1\le k\le J} \sigma_{k,k} + C\sqrt{\frac{\log J}{n}} \right\}^{1/2}, \end{aligned} \tag{A.34}$$

where we use Cauchy-Schwartz for the first inequality, and we use events $\mathcal{A}_2 \cap \mathcal{A}_3$. Same analysis applies to third term on the right side of (A.32). Consider the fourth term on the right side of (A.32)

$$\max_{1\le j\le J}\max_{1\le k\le J}\frac{1}{n}\sum_{i=1}^{n}|u_{j,i}u_{k,i}-\sigma_{j,k}| \le \max_{1\le j\le J}\max_{1\le k\le J}\frac{1}{n}\sum_{i=1}^{n}|u_{j,i}u_{k,i}| + \max_{1\le j\le J,1\le k\le J}|\sigma_{j,k}|. \tag{A.35}$$

Use Cauchy-Schwartz inequality

$$\begin{aligned}\max_{1\le j\le J}\max_{1\le k\le J}\frac{1}{n}\sum_{i=1}^{n}|u_{j,i}u_{k,i}| &\le \sqrt{\max_{1\le j\le J}\frac{1}{n}\sum_{i=1}^{n}u_{j,i}^2}\sqrt{\max_{1\le k\le J}\frac{1}{n}\sum_{i=1}^{n}u_{k,i}^2}\\ &\le [\max_{1\le j\le J}\sigma_{j,j}+C\sqrt{\frac{\log J}{n}}]^{1/2}[\max_{1\le k\le J}\sigma_{k,k}+C\sqrt{\frac{\log J}{n}}]^{1/2} \qquad (A.36)\end{aligned}$$

via event $\mathcal{A}_3$. Combine (A.33)-(A.36) in (A.32) on $\mathcal{A}_1\cap\mathcal{A}_2\cap\mathcal{A}_3$

$$\begin{aligned}\max_{1\le j\le J,1\le k\le J}\frac{1}{n}\sum_{i=1}^{n}|\hat{u}_{j,i}\hat{u}_{k,i}-\sigma_{j,k}| &\le a_n^2+2a_n[\max_{1\le j\le J}\sigma_{j,j}+C\sqrt{\frac{\log J}{n}}]^{1/2}+\max_{1\le j\le J,1\le k\le J}|\sigma_{j,k}|\\ &+ [\max_{1\le j\le J}\sigma_{j,j}+C\sqrt{\frac{\log J}{n}}]^{1/2}[\max_{1\le k\le J}\sigma_{k,k}+C\sqrt{\frac{\log J}{n}}]^{1/2}. \qquad (A.37)\end{aligned}$$

Since $a_n = o(1)$ by Assumption 7

$$\max_{1\le j\le J,1\le k\le J}\hat{\theta}_{j,k} \le \max_{1\le j\le J,1\le k\le J}|\sigma_{j,k}| + \max_{1\le j\le J}\sigma_{j,j} + o(1). \tag{A.38}$$

See that by defining $C_U := \max_{1\le j\le J,1\le k\le J}|\sigma_{j,k}| + \max_{1\le j\le J}\sigma_{j,j} < \infty$ and using Lemma 1-2 we have

$$P(\mathcal{A}_1\cap\mathcal{A}_2\cap\mathcal{A}_3) \ge 1-\frac{3}{J^{c_1}}-\frac{4}{J^{c_2}}.$$

We analyze the lower bound in our problem. We condition on the events $\mathcal{A}_1,\mathcal{A}_2,\mathcal{A}_3,\mathcal{A}_4$. Then we relax this assumption at the end of the proof here. We start with the following triangle inequality and use $\hat{\theta}_{j,k}$ definition

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}|u_{j,i}u_{k,i}-\sigma_{j,k}| &\le \frac{1}{n}\sum_{i=1}^{n}|u_{j,i}u_{k,i}-\hat{u}_{j,i}\hat{u}_{k,i}| + \frac{1}{n}\sum_{i=1}^{n}|\hat{u}_{j,i}\hat{u}_{k,i}-\hat{\sigma}_{j,k}| + \frac{1}{n}\sum_{i=1}^{n}|\hat{\sigma}_{j,k}-\sigma_{j,k}|\\ &\le \frac{1}{n}\sum_{i=1}^{n}|u_{j,i}u_{k,i}-\hat{u}_{j,i}\hat{u}_{k,i}| + \hat{\theta}_{j,k} + \max_{1\le j\le J,1\le k\le J}|\hat{\sigma}_{j,k}-\sigma_{j,k}|. \qquad (A.39)\end{aligned}$$

The term on the left side of (A.39) is lower bounded by

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}|u_{j,i}u_{k,i}-\sigma_{j,k}| &\ge E|u_{j,i}u_{k,i}-\sigma_{j,k}| - C\sqrt{\frac{\log J}{n}}\\ &\ge c - C\sqrt{\frac{\log J}{n}}, \qquad (A.40)\end{aligned}$$

where the first inequality is by event $\mathcal{A}_4$ and iid nature of errors, and the second inequality is by Assumption 7. Then analyze the right side of (A.39) by event $\mathcal{A}_1$

$$\max_{1\le j\le J,1\le k\le J}|\hat{\sigma}_{j,k}-\sigma_{j,k}| \le C''(\sqrt{\frac{\log J}{n}}+a_n). \tag{A.41}$$

Consider the first term on the right side of (A.39) by triangle inequality and (A.33)-(A.34) on $\mathcal{A}_2 \cap \mathcal{A}_3$

$$\begin{aligned}
\max_{j,k} \frac{1}{n} \sum_{i=1}^n |u_{j,i} u_{k,i} - \hat{u}_{j,i} \hat{u}_{k,i}| &\leq \max_{j,k} \frac{1}{n} \sum_{i=1}^n |(\hat{u}_{j,i} - u_{j,i})(\hat{u}_{k,i} - u_{k,i})| + \max_{j,k} \frac{1}{n} \sum_{i=1}^n |(\hat{u}_{j,i} - u_{j,i})(u_{k,i})| \\
&+ \max_{j,k} \frac{1}{n} \sum_{i=1}^n |(u_{j,i})(\hat{u}_{k,i} - u_{k,i})| \\
&\leq a_n^2 + a_n \{ \max_{1 \leq j \leq J} \sigma_{j,j} + C \sqrt{\log J/n} \}^{1/2} \\
&+ a_n \{ \max_{1 \leq k \leq J} \sigma_{k,k} + C \sqrt{\log J/n} \}^{1/2}. \qquad (A.42)
\end{aligned}$$

Combine (A.40)-(A.42) in (A.39),

$$\begin{aligned}
\min_{1 \leq j \leq J} \min_{1 \leq k \leq J} \hat{\theta}_{j,k} &\geq c - C\sqrt{\frac{\log J}{n}} - C'' a_n - C'' \sqrt{\log J/n} - a_n^2 \\
&- a_n \{ \max_{1 \leq j \leq J} \sigma_{j,j} + C \sqrt{\log J/n} \}^{1/2} - a_n \{ \max_{1 \leq k \leq J} \sigma_{k,k} + C \sqrt{\log J/n} \}^{1/2} \\
&= c - o(1),
\end{aligned}$$

by $a_n = o(1)$, $\sqrt{\log J/n} = o(1)$ by Assumption 7. Since we condition on $\cap_{l=1}^4 \mathcal{A}_l$

$$P(\cap_{l=1}^4 \mathcal{A}_l) \geq 1 - \frac{3}{J^{c_1}} - \frac{6}{J^{c_2}},$$

by Lemmata 1-2.

Q.E.D.

Proof of Theorem 3.

(i) Define $b_n := C''(\sqrt{\frac{\log J}{n}} + a_n)$, with $CC_L > 2C'' > 0$ we get

$$C_L \omega_n > 2b_n, \qquad (A.43)$$

by (11).

Next define

$$\mathcal{B}_1 := \{ \max_{1 \leq k \leq J} \max_{1 \leq j \leq J} |\hat{\sigma}_{j,k} - \sigma_{j,k}| \leq C''(\sqrt{\frac{\log J}{n}} + a_n) \},$$

$$\mathcal{B}_2 := \{ \min_{1 \leq k \leq J} \min_{1 \leq j \leq J} \hat{\theta}_{j,k} \omega_n > 2b_n \},$$

$$\mathcal{B}_3 := \{ \max_{1 \leq k \leq J} \max_{1 \leq j \leq J} \hat{\theta}_{j,k} \leq C_U \}.$$

Define also the event $E : \{\cap_{l=1}^3 \mathcal{B}_l\}$. Note that under E, with b_n definition

$$|\hat{\sigma}_{j,k}| \geq \omega_n \hat{\theta}_{j,k} \quad \textit{implies} \quad |\sigma_{j,k}| \geq b_n, \qquad (A.44)$$

since

$$b_n + |\sigma_{j,k}| \geq |\hat{\sigma}_{j,k}| \geq \omega_n \hat{\theta}_{j,k} > 2b_n.$$

and

$$|\hat{\sigma}_{j,k}| < \omega_n \hat{\theta}_{j,k} \quad \textit{implies} \quad |\sigma_{j,k}| < b_n + C_U \omega_n, \qquad (A.45)$$

via

$$\omega_n C_U \geq \omega_n \hat{\theta}_{j,k} > |\hat{\sigma}_{j,k}| > |\sigma_{j,k}| - b_n.$$

Let $\|A\|_{l_2}$ be the spectral norm of matrix A.Then

$$\begin{aligned}
\|\hat{\Sigma}_u^{Th} - \Sigma_u\|_{l_2} &\leq \max_{1\leq j\leq J} \sum_{k=1}^{J} |\hat{\sigma}_{j,k} \mathbb{1}_{\{|\hat{\sigma}_{j,k}|\geq \omega_n \hat{\theta}_{j,k}\}} - \sigma_{j,k}| \\
&\leq \max_{1\leq j\leq J} \sum_{k=1}^{J} |\hat{\sigma}_{j,k} - \sigma_{j,k}| \mathbb{1}_{\{|\hat{\sigma}_{j,k}|\geq \omega_n \hat{\theta}_{j,k}\}} \\
&+ \max_{1\leq j\leq J} \sum_{k=1}^{J} |\sigma_{j,k}| \mathbb{1}_{\{|\hat{\sigma}_{j,k}|< \omega_n \hat{\theta}_{j,k}\}} \\
&\leq \max_{1\leq j\leq J} \sum_{k=1}^{J} |\hat{\sigma}_{j,k} - \sigma_{j,k}| \mathbb{1}_{\{|\sigma_{j,k}|\geq b_n\}} + \max_{1\leq j\leq J} \sum_{j=1}^{J} |\sigma_{j,k}| \mathbb{1}_{\{|\sigma_{j,k}|< b_n + C_U \omega_n\}} \\
&\leq b_n s_n + (b_n + C_U \omega_n) s_n \leq (C_L + C_U)\omega_n s_n = C_2 \omega_n s_n
\end{aligned}$$

where the first inequality is due to $\|A\|_{l_2} \leq \|A\|_{l_\infty}$ for matrix norms, with A being a symmetric matrix, and the third inequality is by (A.44)(A.45), and the fourth inequality is by $\mathcal{B}_1$ and the sparsity definition, and the last inequality is due to (A.43). The last equality is $C_2 := C_L + C_U$.

Note that by (A.43)

$$P(\mathcal{B}_2) = P(\min_{1\leq j\leq J, 1\leq k\leq J} \hat{\theta}_{j,k}\omega_n > 2b_n) = P(\min_{1\leq j\leq J, 1\leq k\leq J} \hat{\theta}_{j,k} > \frac{2b_n}{\omega_n}) \geq P(\min_{1\leq j\leq J, 1\leq k\leq J} \hat{\theta}_{j.k} > C_L).$$

Then by Lemma A.4, and Lemma 2(iii) we have

$$P[E] \geq 1 - O(\frac{1}{J^{\min(c_1,c_2)}}).$$

Q.E.D.

(ii) Given $Eigmin(\Sigma_u) \geq c > 0$, and (i), the proof follows exactly as in proof of Theorem 2.1(ii) of Fan et al. (2011).

Q.E.D.

Proof of Lemma 3.

There are three steps in this proof.

Step 1.

We start with definitions. From the definition of the covariance matrix, we can simplify the estimator for the covariance matrix for function of factors

$$\hat{\Sigma}^f = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(X_i)\hat{f}(X_i)' - \bar{f}(X_i)\bar{f}(X_i)',$$

where (j,k) th element can be written as

$$\hat{\Sigma}^f_{j,k} = \frac{1}{n}\sum_{i=1}^{n} \hat{f}_j(X_i)\hat{f}_k(X_i) - \bar{f}_j(X_i)\bar{f}_k(X_i), \tag{A.46}$$

with $\bar{f}_j(X_i) := \frac{1}{n}\sum_{i=1}^n \hat{f}_j(X_i), \bar{f}_k(X_i) := \frac{1}{n}\sum_{i=1}^n \hat{f}_k(X_i)$. Now define the infeasible covariance matrix estimator for function of factors

$$\bar{\Sigma}^f := \frac{1}{n}\sum_{i=1}^n [f_0(X_i) - \bar{f}_0(X_i)][f_0(X_i) - \bar{f}_0(X_i)]' = \frac{1}{n}\sum_{i=1}^n f_0(X_i)f_0(X_i)' - \bar{f}_0(X_i)\bar{f}_0(X_i)', \tag{A.47}$$

with $\bar{f}_0(X_i) := \frac{1}{n}\sum_{i=1}^n f_0(X_i) : J \times 1$, and

$$f_0(X_i) := (f_{0,1}(X_i), \cdots, f_{0,j}(X_i), \cdots, f_{0,J}(X_i))' : \quad J \times 1.$$

The (j,k) th element of $\bar{\Sigma}^f$ is

$$\bar{\Sigma}^f_{j,k} := \frac{1}{n}\sum_{i=1}^n f_{0,j}(X_i)f_{0,k}(X_i) - \bar{f}_{0,j}(X_i)\bar{f}_{0,k}(X_i),$$

with $\bar{f}_{0,j}(X_i) := \frac{1}{n}\sum_{i=1}^n f_{0,j}(X_i)$, and $\bar{f}_{0,k}$ is defined in the same way, k th asset replacing j th asset.

Step 2.

Now we start with the following triangle inequality.

$$\|\hat{\Sigma}^f - \Sigma^f\|_\infty \leq \|\hat{\Sigma}^f - \bar{\Sigma}^f\|_\infty + \|\bar{\Sigma}^f - \Sigma^f\|_\infty. \tag{A.48}$$

In (A.48) we start with the first term on the right side and use definitions (A.46)(A.47) with triangle inequality

$$\begin{aligned}\|\hat{\Sigma}^f - \bar{\Sigma}^f\|_\infty \quad &\leq \max_{1\leq j\leq J, 1\leq k\leq J}\left|\frac{1}{n}\sum_{i=1}^n \hat{f}_j(X_i)\hat{f}_k(X_i) - \frac{1}{n}\sum_{i=1}^n f_{0,j}(X_i)f_{0,k}(X_i)\right| \\ &+ \max_{1\leq j\leq J, 1\leq k\leq J}\left|(\frac{1}{n}\sum_{i=1}^n \hat{f}_j(X_i))(\frac{1}{n}\sum_{i=1}^n \hat{f}_k(X_i)) - (\frac{1}{n}\sum_{i=1}^n f_{0,j}(X_i))(\frac{1}{n}\sum_{i=1}^n f_{0,k}(X_i))\right|\end{aligned} \tag{A.49}$$

Step 2a.

In (A.49) we consider the first right side term, by using $\hat{f}_j(X_i) = [\hat{f}_j(X_i) - f_{0,j}(X_i)] + f_{0,j}(X_i)$, and repeating the same for $\hat{f}_k(X_i)$ and triangle inequality

$$\begin{aligned}\max_{1\leq j\leq J, 1\leq k\leq J} \quad &\left|\frac{1}{n}\sum_{i=1}^n \hat{f}_j(X_i)\hat{f}_k(X_i) - \frac{1}{n}\sum_{i=1}^n f_{0,j}(X_i)f_{0,k}(X_i)\right| \\ &\leq \max_{1\leq j\leq J, 1\leq k\leq J}\left|\frac{1}{n}\sum_{i=1}^n (\hat{f}_j(X_i) - f_{0,j}(X_i))(\hat{f}_k(X_i) - f_{0,k}(X_i))\right| \\ &+ \max_{1\leq j\leq J, 1\leq k\leq J}\left|\frac{1}{n}\sum_{i=1}^n (\hat{f}_j(X_i) - f_{0,j}(X_i))(f_{0,k}(X_i))\right| \\ &+ \max_{1\leq j\leq J, 1\leq k\leq J}\left|\frac{1}{n}\sum_{i=1}^n (f_{0,j}(X_i))(\hat{f}_k(X_i) - f_{0,k}(X_i))\right|.\end{aligned} \tag{A.50}$$

Next consider the first right side term in (A.50), by Cauchy-Schwartz inequality

$$\begin{aligned}\max_{1\le j\le J, 1\le k\le J}\left|\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i)-f_{0,j}(X_i))(\hat{f}_k(X_i)-f_{0,k}(X_i))\right| &\le \max_{1\le j\le J}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i)-f_{0,j}(X_i))^2}\\ &\times \max_{1\le k\le J}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_k(X_i)-f_{0,k}(X_i))^2}\\ &= O_p(a_n^2), \qquad (A.51)\end{aligned}$$

where the rate is by Theorem 2, and $a_n^2 := \max(r_{n1}, r_{n2})$. In the same way as in (A.51) via Cauchy-Schwartz inequality, consider the second term on the right side of (A.50)

$$\begin{aligned}\max_{1\le j\le J, 1\le k\le J}\left|\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i)-f_{0,j}(X_i))(f_{0,k}(X_i))\right| &\le \max_{1\le j\le J}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i)-f_{0,j}(X_i))^2}\\ &\times \max_{1\le k\le J}\sqrt{\frac{1}{n}\sum_{i=1}^{n}f_{0,k}(X_i)^2} = O_p(a_n), \qquad (A.52)\end{aligned}$$

where we use Theorem 2 and uniformly bounded functions $f_{0,k}(X_i)$ in Theorem 2 statement for the rate. Next we use the same analysis for the third term on the right side of (A.50) and combine (A.51)(A.52) in (A.50) to have

$$\max_{1\le j\le J, 1\le k\le J}\left|\frac{1}{n}\sum_{i=1}^{n}\hat{f}_j(X_i)\hat{f}_k(X_i)-\frac{1}{n}\sum_{i=1}^{n}f_{0,j}(X_i)f_{0,k}(X_i)\right| = O_p(a_n). \qquad (A.53)$$

Step 2b. We consider the second term on the right side of (A.49). Add and subtract

$$\begin{aligned}\left(\frac{1}{n}\sum_{i=1}^{n}\hat{f}_j(X_i)\right)\left(\frac{1}{n}\sum_{i=1}^{n}\hat{f}_k(X_i)\right) &- \left(\frac{1}{n}\sum_{i=1}^{n}f_{0,j}(X_i)\right)\left(\frac{1}{n}\sum_{i=1}^{n}f_{0,k}(X_i)\right)\\ &= \left(\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i)-f_{0,j}(X_i))\right)\left(\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_k(X_i)-f_{0,k}(X_i))\right)\\ &+ \left(\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i)-f_{0,j}(X_i))\right)\left(\frac{1}{n}\sum_{i=1}^{n}f_{0,k}(X_i)\right)\\ &+ \left(\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_k(X_i)-f_{0,k}(X_i))\right)\left(\frac{1}{n}\sum_{i=1}^{n}f_{0,j}(X_i)\right). \qquad (A.54)\end{aligned}$$

In (A.54) by l_1, l_2 norm inequality we get

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i)-f_{0,j}(X_i)) \le \frac{1}{n}\sum_{i=1}^{n}|(\hat{f}_j(X_i)-f_{0,j}(X_i))| \le \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i)-f_{0,j}(X_i))^2}.$$

The same analysis is applied to $\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_k(X_i)-f_{0,k}(X_i))$. The first term on the right side of (A.54) can be upper bounded as

$$\begin{aligned}\max_{1\leq j\leq J, 1\leq k\leq J}\left|\left(\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i)-f_{0,j}(X_i))\right)\left(\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_k(X_i)-f_{0,k}(X_i))\right)\right| &\leq \max_{1\leq j\leq J}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i)-f_{0,j}(X_i))^2}\\ &\times \max_{1\leq k\leq J}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_k(X_i)-f_{0,k}(X_i))^2}\\ &= O_p(a_n^2), \qquad (A.55)\end{aligned}$$

by Theorem 2. Then in (A.54)

$$\begin{aligned}\max_{1\leq j\leq j, 1\leq k\leq J}\left|\left(\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i)-f_{0,j}(X_i))\right)\left(\frac{1}{n}\sum_{i=1}^{n}f_{0,k}(X_i)\right)\right| &\leq \max_{1\leq j\leq J}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_j(X_i)-f_{0,j}(X_i))^2}\\ &\times \max_{1\leq k\leq J}\left|\frac{1}{n}\sum_{i=1}^{n}f_{0,k}(X_i)\right|\\ &= O_p(a_n), \qquad (A.56)\end{aligned}$$

by Theorem 2 and uniformly bounded $f_{0,k}(X_i)$ by Assumption. Same analysis in (A.56) applies to third term on the right side of (A.54) hence combine (A.55)(A.56) in (A.54)

$$\max_{1\leq j\leq J, 1\leq k\leq J}\left|\left(\frac{1}{n}\sum_{i=1}^{n}\hat{f}_j(X_i)\right)\left(\frac{1}{n}\sum_{i=1}^{n}\hat{f}_k(X_i)\right)-\left(\frac{1}{n}\sum_{i=1}^{n}f_{0,j}(X_i)\right)\left(\frac{1}{n}\sum_{i=1}^{n}f_{0,k}(X_i)\right)\right| = O_p(a_n). \quad (A.57)$$

Use (A.53)(A.57) in (A.49) to have

$$\|\hat{\Sigma}^f-\bar{\Sigma}^f\|_\infty = O_p(a_n). \quad (A.58)$$

Step 2c. In (A.48) consider the second term on the right side

$$\begin{aligned}\|\bar{\Sigma}^f-\Sigma^f\|_\infty &\leq \max_{1\leq j\leq J, 1\leq k\leq J}\left|\frac{1}{n}\sum_{i=1}^{n}f_{0,j}(X_i)f_{0,k}(X_i)-E[f_{0,j}(X_i)f_{0,k}(X_i)]\right|\\ &+ \max_{1\leq j\leq J, 1\leq k\leq J}\left|(\frac{1}{n}\sum_{i=1}^{n}f_{0,j}(X_i))(\frac{1}{n}\sum_{i=1}^{n}f_{0,k}(X_i))-E[f_{0,j}(X_i)]E[f_{0,k}(X_i)]\right|. \quad (A.59)\end{aligned}$$

Take the first term on the right side of (A.59)

$$\max_{1\leq j\leq J, 1\leq k\leq J}\left|\frac{1}{n}\sum_{i=1}^{n}f_{0,j}(X_i)f_{0,k}(X_i)-E[f_{0,j}(X_i)f_{0,k}(X_i)]\right| = O_p(\sqrt{\frac{\log J}{n}}), \quad (A.60)$$

by the proof of Lemma 1 (i) since $f_{0,j}(X_i), f_{0,k}(X_i)$ are uniformly bounded (subgaussian) hence their product is subexponential. Consider the second term on the right side of (A.59) by adding and subtracting, and triangle inequality

$$\begin{aligned}
& \max_{1\le j\le J, 1\le k\le J} \left| (\frac{1}{n}\sum_{i=1}^n f_{0,j}(X_i))(\frac{1}{n}\sum_{i=1}^n f_{0,k}(X_i)) - E[f_{0,j}(X_i)]E[f_{0,k}(X_i)] \right| \\
\le & \max_{1\le j\le J, 1\le k\le J} \left| \left(\frac{1}{n}\sum_{i=1}^n (f_{0,j}(X_i) - E[f_{0,j}(X_i)]) \right) \left(\frac{1}{n}\sum_{i=1}^n (f_{0,k}(X_i) - E[f_{0,k}(X_i)]) \right) \right| \\
+ & \max_{1\le j\le J, 1\le k\le J} \left| \left(\frac{1}{n}\sum_{i=1}^n (f_{0,j}(X_i) - E[f_{0,j}(X_i)]) \right) E[f_{0,k}(X_i)] \right| \\
+ & \max_{1\le j\le J, 1\le k\le J} \left| \left(\frac{1}{n}\sum_{i=1}^n (f_{0,k}(X_i) - E[f_{0,k}(X_i)]) \right) E[f_{0,j}(X_i)] \right|. \qquad (A.61)
\end{aligned}$$

In (A.61) consider the first term on the right side

$$\begin{aligned}
\max_{1\le j\le J, 1\le k\le J} | \left(\frac{1}{n}\sum_{i=1}^n (f_{0,j}(X_i) - E[f_{0,j}(X_i)]) \right) & \times \left(\frac{1}{n}\sum_{i=1}^n (f_{0,k}(X_i) - E[f_{0,k}(X_i)]) \right) | \\
& \le \max_{1\le j\le J} \left| \frac{1}{n}\sum_{i=1}^n (f_{0,j}(X_i) - E[f_{0,j}(X_i)]) \right| \\
& \times \max_{1\le k\le J} \left| \frac{1}{n}\sum_{i=1}^n (f_{0,k}(X_i) - E[f_{0,k}(X_i)]) \right|. \qquad (A.62)
\end{aligned}$$

By Hoeffding inequality in (2.11) of Wainwright (2019), since our functions are uniformly bounded (subgaussian), then taking the union bound with $t_2 = 2F\sqrt{\log 2J/n}$

$$\begin{aligned}
P\left[\max_{1\le j\le J} \left| \frac{1}{n}\sum_{i=1}^n f_{0,j}(X_i) - Ef_{0,j}(X_i) \right| \ge t_2 \right] & \le 2J\exp\left(\frac{-2n}{4F^2}(t_2^2) \right) \\
& = \frac{1}{2J}. \qquad (A.63)
\end{aligned}$$

Second term on the right side of (A.62) is handled in the same way as in (A.63) hence

$$\max_{1\le j\le J, 1\le k\le J} \left| \left(\frac{1}{n}\sum_{i=1}^n (f_{0,j}(X_i) - E[f_{0,j}(X_i)]) \right) \left(\frac{1}{n}\sum_{i=1}^n (f_{0,k}(X_i) - E[f_{0,k}(X_i)]) \right) \right| = O_p\left(\frac{\log J}{n} \right). \qquad (A.64)$$

Since our functions are uniformly bounded we have $E|f_{0,k}(X_i)| \le F$, second term on the right side of (A.61) uses (A.63)

$$\max_{1\le j\le J, 1\le k\le J} \left| \left(\frac{1}{n}\sum_{i=1}^n (f_{0,j}(X_i) - E[f_{0,j}(X_i)]) \right) E[f_{0,k}(X_i)] \right| = O_p\left(\sqrt{\frac{\log J}{n}} \right). \qquad (A.65)$$

Third term on the right side of (A.61) follows the same analysis in (A.65) above hence via second term on the right side of (A.59) and the analysis in (A.62)-(A.65)

$$\max_{1\le j\le J, 1\le k\le J}\left|(\frac{1}{n}\sum_{i=1}^n f_{0,j}(X_i))(\frac{1}{n}\sum_{i=1}^n f_{0,k}(X_i)) - E[f_{0,j}(X_i)]E[f_{0,k}(X_i)]\right| = O_p\left(\sqrt{\frac{\log J}{n}}\right). \tag{A.66}$$

Step 3. Use (A.60) and (A.66) in (A.59) to have

$$\|\bar{\Sigma}^f - \Sigma\|_\infty = O_p\left(\frac{\sqrt{\log J}}{\sqrt{n}}\right). \tag{A.67}$$

Since rate $a_n := \max(\sqrt{r_{n1}}, \sqrt{r_{n2}})$ is always slower than $\sqrt{\log J/n}$ combine (A.58) and (A.67) in (A.48) to have

$$\|\hat{\Sigma}^f - \Sigma\|_\infty = O_p(a_n).$$

Q.E.D.

Proof of Theorem 4. Clearly by definitions of $\hat{\Sigma}_y, \Sigma_y$,

$$\|\hat{\Sigma}_y - \Sigma_y\|_\infty \le \|\hat{\Sigma}^f - \Sigma^f\|_\infty + \|\hat{\Sigma}_u^{Th} - \Sigma_u\|_\infty. \tag{A.68}$$

In the inequality above we analyze the second right side term, using the technique as in the proof of Theorem 3. In that sense we have the following definitions, $\sigma_{j,k} := Eu_{j,i}u_{k,i}$, and by $b_n = C''(\sqrt{\log J/n} + a_n)$

$$\mathcal{B}_1 := \{\max_{1\le k\le J}\max_{1\le j\le J}|\hat{\sigma}_{j,k} - \sigma_{j,k}| \le b_n\},$$

$$\mathcal{B}_2 := \{\min_{1\le k\le J}\min_{1\le j\le J}\hat{\theta}_{j,k}\omega_n > 2b_n\},$$

$$\mathcal{B}_3 := \{\max_{1\le k\le J}\max_{1\le j\le J}\hat{\theta}_{j,k} \le C_U\}.$$

Define also the event $E : \{\cap_{l=1}^3 \mathcal{B}_l\}$. Under event E

$$\begin{aligned}
\|\hat{\Sigma}_u^{Th} - \Sigma_u\|_\infty &= \max_{1\le j\le J}\max_{1\le k\le J}|\hat{\sigma}_{j,k}\mathbb{1}_{\{|\hat{\sigma}_{j,k}|\ge\omega_n\hat{\theta}_{j,k}\}} - \sigma_{j,k}| \\
&\le \max_{1\le j\le J}\max_{1\le k\le J}|\hat{\sigma}_{j,k} - \sigma_{j,k}|\mathbb{1}_{\{|\hat{\sigma}_{j,k}|\ge\omega_n\hat{\theta}_{j,k}\}} \\
&+ \max_{1\le j\le J}\max_{1\le k\le J}|\sigma_{j,k}|\mathbb{1}_{\{|\hat{\sigma}_{j,k}|<\omega_n\hat{\theta}_{j,k}\}} \\
&\le \max_{1\le j\le J}\max_{1\le k\le J}|\hat{\sigma}_{j,k} - \sigma_{j,k}|\mathbb{1}_{\{|\sigma_{j,k}|\ge b_n\}} + \max_{1\le j\le J}\max_{1\le k\le J}|\sigma_{j,k}|\mathbb{1}_{\{|\sigma_{j,k}|<b_n + C_U\omega_n\}} \\
&\le b_n + (b_n + C_U\omega_n) \le (C_L + C_U)\omega_n,
\end{aligned}$$

where the first equality is the definition, and the second inequality is by (A.44)(A.45), and the third inequality is by $\mathcal{B}_1$ definition, and the last inequality is by (A.43). Following the proof of Theorem 3

$$P(E) \ge 1 - O(1/J^{min(c_1,c_2)}),$$

so

$$\|\hat{\Sigma}_u^{Th} - \Sigma_u\|_\infty = O_p(\omega_n).$$

Then using Lemma 3, and since $\omega_n = O(\sqrt{\log J/n} + a_n) = O(a_n)$ we have by the last result in (A.68)

$$\|\hat{\Sigma}_y - \Sigma_y\|_\infty = O_p(\omega_n) = O_p(a_n).$$

Q.E.D.

Consistency of Estimate of Precision Matrix of Returns

This part of the Appendix analyzes the estimate for the precision matrix of the returns based on our sparse deep neural network. We need several Lemmata to begin with. Our results in this part of the paper only allow $J << n$.

Lemma A.5. *Under Assumptions 1,2,5-9*

$$\|(\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f - \Sigma_u^{-1}\Sigma^f\|_{l_2} = O_p(J\omega_n s_n).$$

Proof of A.5. First by adding and subtracting and triangle inequality

$$\begin{aligned} \|(\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f - \Sigma_u^{-1}\Sigma^f\|_{l_2} &\leq \|[(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}][\hat{\Sigma}^f - \Sigma^f]\|_{l_2} \\ &+ \|[(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}]\Sigma^f\|_{l_2} + \|\Sigma_u^{-1}(\hat{\Sigma}^f - \Sigma^f)\|_{l_2}. \end{aligned} \tag{A.69}$$

In the following, we analyze each term separately. See that

$$\begin{aligned} \|[(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}][\hat{\Sigma}^f - \Sigma^f]\|_{l_2} &\leq \|[(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}]\|_{l_2}\|[\hat{\Sigma}^f - \Sigma^f]\|_{l_2} \\ &\leq J\|[(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}]\|_{l_2}\|[\hat{\Sigma}^f - \Sigma^f]\|_\infty \\ &= JO_p(\omega_n s_n)O_p(a_n), \end{aligned} \tag{A.70}$$

by norm inequality that ties the spectral norm to the $\|.\|_\infty$ norm on p. 365 of Horn and Johnson (2013), and by Theorem 3, and Lemma 3. Then in (A.69) consider

$$\begin{aligned} \|[(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}]\Sigma^f\|_{l_2} &\leq \|[(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}]\|_{l_2}\|\Sigma^f\|_{l_2} \\ &= O_p(\omega_n s_n)O(J), \end{aligned} \tag{A.71}$$

by Theorem 3 and Assumption 8 $\|\Sigma^f\|_{l_2} = Eigmax(\Sigma^f) = O(J)$. Next, on the right side of (A.69)

$$\|\Sigma_u^{-1}(\hat{\Sigma}^f - \Sigma^f)\|_{l_2} \leq \|\Sigma_u^{-1}\|_{l_2}\|\hat{\Sigma}^f - \Sigma^f\|_{l_2} \leq J\|\Sigma_u^{-1}\|_{l_2}\|\hat{\Sigma}^f - \Sigma^f\|_\infty = O_p(Ja_n), \tag{A.72}$$

where we use Lemma 3, and norm inequality that ties the spectral norm to the $\|.\|_\infty$ norm on p. 365 of Horn and Johnson (2013), and

$$\|\Sigma_u^{-1}\|_{l_2} = Eigmax(\Sigma_u^{-1}) = \frac{1}{Eigmin(\Sigma_u)} \leq \frac{1}{c} < \infty. \tag{A.73}$$

by Assumption 1. Note that $a_n \to 0$, and $\omega_n s_n \to 0$ by Assumption 9, since $\omega_n s_n = s_n(\sqrt{\log J/n} + a_n) > a_n$. Hence, the rate in (A.71) is the slowest.

Q.E.D.

Lemma A.6. *Under Assumptions 1,2,5-9, with $r_n \to 0$ or $r_n = 0$, with $\delta_n \to \infty$*

(i)

$$\left\| \left[I_J + \Sigma_u^{-1}\Sigma^f \right]^{-1} \right\|_{l_2} = O(\frac{\delta_n}{\delta_n + r_n}) = O(1).$$

(ii)

$$\left\| \left[I_J + (\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f \right]^{-1} \right\|_{l_2} = O_p(\frac{\delta_n}{\delta_n + r_n}) = O_p(1).$$

Proof of Lemma A.6.

(i) We start the proof with the following eigenvalue inequality in Abadir and Magnus (2005) which is on p. 344 as Exercise 12.40a, with a proof, for the symmetric matrices A and B

$$Eigmin(A + B) \geq Eigmin(A) + Eigmin(B). \tag{A.74}$$

Now apply (A.74) with $A = I_J, B = \Sigma_u^{-1}\Sigma^f$ in the first inequality below

$$\begin{aligned} Eigmin(I_J + \Sigma_u^{-1}\Sigma^f) &\geq Eigmin(I_J) + Eigmin(\Sigma_u^{-1}\Sigma^f) \\ &\geq 1 + Eigmin(\Sigma_u^{-1})Eigmin(\Sigma^f) \\ &= 1 + \frac{1}{Eigmax(\Sigma_u)} Eigmin(\Sigma^f) \\ &\geq 1 + \frac{r_n}{C\delta_n}, \end{aligned} \tag{A.75}$$

where the second inequality follows from Fact 10.22.23 in Bernstein (2018) on p. 809, and the remaining derivations follow from Assumptions 8-9. Note that

$$Eigmax[(I_J + \Sigma_u^{-1}\Sigma^f)^{-1}] = \frac{1}{Eigmin[(I_J + \Sigma_u^{-1}\Sigma^f)]} = O(\frac{1}{1 + \frac{r_n}{\delta_n}}) = O(\frac{\delta_n}{\delta_n + r_n}).$$

(ii)

$$\| \left[I_J + (\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f \right] - \left[I_J + \Sigma_u^{-1}\Sigma^f \right] \|_{l_2} = \|(\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f - \Sigma_u^{-1}\Sigma^f\|_{l_2} = O_p(J\omega_n s_n),$$

by Lemma A.5. Then by Assumption 9, $J\omega_n s_n \to 0$, and Assumption 8(ii), $1 + r_n/C\delta_n \to 1$ or exactly 1 depending on r_n, and via Lemma A.1(i) of Fan et al. (2011)

$$\begin{aligned} P \quad & \left[Eigmin(I_J + (\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f) \geq 0.5[1 + r_n/C\delta_n] \right] \\ \geq \quad & P\left[\| \left[I_J + (\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f \right] - \left[I_J + \Sigma_u^{-1}\Sigma^f \right] \|_{l_2} \leq 0.5[1 + r_n/C\delta_n] \right] \\ \geq \quad & 1 - o(1). \end{aligned}$$

So by Lemma A.1(ii) of Fan et al. (2011)

$$\left\| \left[I_J + (\hat{\Sigma}_u^{Th})^{-1} \hat{\Sigma}^f) \right]^{-1} \right\|_{l_2} = O_p(\frac{\delta_n}{\delta_n + r_n}) = O_p(1),$$

since either $\frac{\delta_n}{\delta_n + r_n} = 1$, or $\frac{\delta_n}{\delta_n + r_n} \to 1$ by Assumption 8(ii).

Q.E.D.

Lemma A.7. *Under Assumptions 1,2,5-9*

(i)

$$\|\hat{\Sigma}^f\|_{l_2} = O_p(J).$$

(ii)

$$\|(\hat{\Sigma}_u^{Th})^{-1}\|_{l_2} = O_p(1).$$

Proof of Lemma A.7.

(i)

$$\begin{aligned} \|\hat{\Sigma}^f\|_{l_2} &\leq \|\hat{\Sigma}^f - \Sigma^f\|_{l_2} + \|\Sigma^f\|_{l_2} \\ &\leq [J\|\hat{\Sigma}^f - \Sigma_f\|_\infty] + \|\Sigma^f\|_{l_2} \\ &= O_p(Ja_n) + O(J) = O_p(J), \end{aligned}$$

by p. 365 of Horn and Johnson (2013), tying the $\|.\|_{l_2}$ spectral norm to the $\|.\|_\infty$ norm, and via Lemma 3, and Assumption 9 which implies $a_n \to 0$, by $\omega_n = O(a_n)$ and Assumption 8(ii).

(ii)

$$\begin{aligned} \|(\hat{\Sigma}_u^{Th})^{-1}\|_{l_2} &\leq \|(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}\|_{l_2} + \|\Sigma_u^{-1}\|_{l_2} \\ &= O_p(\omega_n s_n) + O(1) \\ &= O_p(1), \end{aligned}$$

by Theorem 3, Assumption 9 implies $\omega_n s_n \to 0$, and

$$\|\Sigma_u^{-1}\|_{l_2} = Eigmax(\Sigma_u^{-1}) = \frac{1}{Eigmin(\Sigma_u)} \leq \frac{1}{c} < \infty,$$

by Assumption 1. **Q.E.D.**

Proof of Theorem 5

First define

$$\hat{G} := [I_J + (\hat{\Sigma}_u^{Th})^{-1} \hat{\Sigma}^f]^{-1}. \tag{A.76}$$

$$G := [I_J + \Sigma_u^{-1} \Sigma^f]^{-1}. \tag{A.77}$$

Note that by Lemma A.6 and the definitions (A.76), (A.77)

$$\|\hat{G}\|_{l_2} = O_p(1). \tag{A.78}$$

$$\|G\|_{l_2} = O(1). \tag{A.79}$$

Then by the equations (13) and (14) we obtain

$$\|\hat{\Sigma}_y^{-1} - \Sigma_y^{-1}\|_{l_2} \leq \|(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}\|_{l_2} + \|(\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f\hat{G}(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}\Sigma^f G\Sigma_u^{-1}\|_{l_2}. \tag{A.80}$$

First we consider the second term on the right-hand side in (A.80). One issue concerns the simplification of that term such that we can benefit from the previously introduced Lemmata. In that respect, we add and subtract $\Sigma_u^{-1}\hat{\Sigma}^f\hat{G}(\hat{\Sigma}_u^{Th})^{-1}$ and use the triangle inequality to gather

$$\begin{aligned}\|(\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f\hat{G}(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}\Sigma^f G\Sigma_u^{-1}\|_{l_2} &\leq \|[(\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f - \Sigma_u^{-1}\hat{\Sigma}^f]\hat{G}(\hat{\Sigma}_u^{Th})^{-1}\|_{l_2} \\ &+ \|\Sigma_u^{-1}\hat{\Sigma}^f\hat{G}(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}\Sigma^f G\Sigma_u^{-1}\|_{l_2}. \end{aligned} \tag{A.81}$$

Take the second term in (A.81) add and subtract $\Sigma_u^{-1}\Sigma^f\hat{G}(\hat{\Sigma}_u^{Th})^{-1}$ and use the triangle inequality to obtain

$$\begin{aligned}\|\Sigma_u^{-1}\hat{\Sigma}^f\hat{G}(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}\Sigma^f G\Sigma_u^{-1}\|_{l_2} &\leq \|\Sigma_u^{-1}(\hat{\Sigma}^f - \Sigma^f)\hat{G}(\hat{\Sigma}_u^{Th})^{-1}\|_{l_2} \\ &+ \|\Sigma_u^{-1}\Sigma^f\hat{G}(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}\Sigma^f G\Sigma_u^{-1}\|_{l_2}. \end{aligned} \tag{A.82}$$

Take the second term on the right side of (A.82), add and subtract $\Sigma_u^{-1}\Sigma^f\hat{G}\Sigma_u^{-1}$ and via triangle inequality we get

$$\begin{aligned}\|\Sigma_u^{-1}\Sigma^f\hat{G}(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}\Sigma^f G\Sigma_u^{-1}\|_{l_2} &\leq \|\Sigma_u^{-1}\Sigma^f\hat{G}(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}\Sigma^f\hat{G}\Sigma_u^{-1}\|_{l_2} \\ &+ \|\Sigma_u^{-1}\Sigma^f\hat{G}\Sigma_u^{-1} - \Sigma_u^{-1}\Sigma^f G\Sigma_u^{-1}\|_{l_2}. \end{aligned} \tag{A.83}$$

Combine (A.82) and (A.83) in (A.81) which yields

$$\begin{aligned}\|(\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f\hat{G}(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}\Sigma^f G\Sigma_u^{-1}\|_{l_2} &\leq \|[(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}]\hat{\Sigma}^f\hat{G}(\hat{\Sigma}_u^{Th})^{-1}\|_{l_2} \\ &+ \|\Sigma_u^{-1}(\hat{\Sigma}^f - \Sigma^f)\hat{G}(\hat{\Sigma}_u^{Th})^{-1}\|_{l_2} \\ &+ \|\Sigma_u^{-1}\Sigma^f\hat{G}[(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}]\|_{l_2} \\ &+ \|\Sigma_u^{-1}\Sigma^f(\hat{G} - G)\Sigma_u^{-1}\|_{l_2}. \end{aligned} \tag{A.84}$$

In the following, we compute the rates for each term on the right side of (A.84). Consider the first term on the right side of (A.84)

$$\begin{aligned}\|[(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}]\hat{\Sigma}^f\hat{G}(\hat{\Sigma}_u^{Th})^{-1}\|_{l_2} &\leq \|[(\hat{\Sigma}_u^{Th})^{-1} - \Sigma_u^{-1}]\|_{l_2}\|\hat{\Sigma}^f\|_{l_2}\|\hat{G}\|_{l_2}\|(\hat{\Sigma}_u^{Th})^{-1}\|_{l_2} \\ &= O_p(\omega_n s_n)O_p(J)O_p(1)O_p(1) = O_p(J\omega_n s_n) = o_p(1), \end{aligned} \tag{A.85}$$

by Theorem 3, Lemma A.7, (A.78) and the last equality is by Assumption 9. Analyze the second term on the right side of (A.84)

$$\begin{aligned}\|\Sigma_u^{-1}(\hat{\Sigma}^f-\Sigma^f)\hat{G}(\hat{\Sigma}_u^{Th})^{-1}\|_{l_2} &\leq \|\Sigma_u^{-1}\|_{l_2}\|(\hat{\Sigma}^f-\Sigma^f)\|_{l_2}\|\hat{G}\|_{l_2}\|(\hat{\Sigma}_u^{Th})^{-1}\|_{l_2}\\ &\leq \|\Sigma_u^{-1}\|_{l_2}[J\|(\hat{\Sigma}^f-\Sigma^f)\|_\infty]\|\hat{G}\|_{l_2}\|(\hat{\Sigma}_u^{Th})^{-1}\|_{l_2}\\ &= O(1)O_p(Ja_n)O_p(1)O_p(1)=O_p(Ja_n)=o_p(1), \qquad (A.86)\end{aligned}$$

by (A.73), by the inequality tying the spectral norm to the $\|.\|_\infty$ norm in p. 365 of Horn and Johnson (2013), Lemma 3, Lemma A.6(ii)-$\hat{G}$ definition in (A.76) and (A.78), Lemma A.7(ii), and the last equality is by Assumption 9. Consider the third term on the right side of (A.84)

$$\begin{aligned}\|\Sigma_u^{-1}\Sigma^f\hat{G}[(\hat{\Sigma}_u^{Th})^{-1}-\Sigma_u^{-1}]\|_{l_2} &\leq \|\Sigma_u^{-1}\|_{l_2}\|\Sigma^f\|_{l_2}\|\hat{G}\|_{l_2}\|[(\hat{\Sigma}_u^{Th})^{-1}-\Sigma_u^{-1}]\|_{l_2}\\ &= O(1)O(J)O_p(1)O_p(\omega_n s_n)=O_p(J\omega_n s_n)=o_p(1), \qquad (A.87)\end{aligned}$$

by (A.73), Lemma A.6(ii)- $\hat{G}$ definition in (A.76) and (A.78), Theorem 3 with Assumptions 8-9.
The fourth term on the right side of (A.84) is

$$\begin{aligned}\|\Sigma_u^{-1}\Sigma^f(\hat{G}-G)\Sigma_u^{-1}\|_{l_2} &\leq \|\Sigma_u^{-1}\|_{l_2}^2\|\Sigma^f\|_{l_2}\|\hat{G}-G\|_{l_2}\\ &\leq \|\Sigma_u^{-1}\|_{l_2}^2\|\Sigma^f\|_{l_2}\|\hat{G}\|_{l_2}\|G\|_{l_2}\|(\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f-\Sigma_u^{-1}\Sigma^f\|_{l_2}\\ &= O(1)O(J)O_p(1)O(1)O_p(J\omega_n s_n)=O_p(J^2\omega_n s_n)=o_p(1), \qquad (A.88)\end{aligned}$$

where we use the definitions of $\hat{G}$ and G in (A.76) and (A.77) with (A.78) and (A.79)

$$\|\hat{G}-G\|_{l_2}\leq\|\hat{G}\|_{l_2}\|G\|_{l_2}\|(I_J+(\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f)-(I_J+\Sigma_u^{-1}\Sigma^f)\|_{l_2},$$

to get the second inequality on the right side of (A.88). For the rates we use (A.73), Lemma A.5, A.6 and Assumptions 8-9. By the definition of ω_n, we obtain $\omega_n=O(\sqrt{\log J/n}+a_n)=O(a_n)$. Hence, the slowest rate among (A.85)-(A.88) is (A.88). Then by (A.80) we have

$$\begin{aligned}\|\hat{\Sigma}_y^{-1}-\Sigma_y^{-1}\|_{l_2} &\leq \|(\hat{\Sigma}_u^{Th})^{-1}-\Sigma_u^{-1}\|_{l_2}\\ &+ \|(\hat{\Sigma}_u^{Th})^{-1}\hat{\Sigma}^f\hat{G}(\hat{\Sigma}_u^{Th})^{-1}-\Sigma_u^{-1}\Sigma^f G\Sigma_u^{-1}\|_{l_2}\\ &= O_p(\omega_n s_n)+O_p(J^2\omega_n s_n)=O_p(J^2\omega_n s_n)=o_p(1),\end{aligned}$$

by Theorem 3 and (A.88).

Q.E.D.

References

Abadir, K. and J. Magnus (2005). *Matrix Algebra.* Cambridge University Press.

Ao, M., Y. Li, and X. Zheng (2019). Approaching mean-variance efficiency for large portfolios. *Review of Financial Studies 32*, 2499–2540.

Bai, J. and Y. Liao (2016). Efficient estimation of approximate factor models via penalized maximum likelihood. *Journal of Econometrics 191*(1), 1–18.

Bai, J. and S. Ng (2002). Determining the number of factors in approximate factor models. *Econometrica 70*(1), 191–221.

Bauer, B. and M. Kohler (2019). On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *The Annals of Statistics 47*(4), 2261 – 2285.

Bernstein, D. (2018). *Scalar vector, and matrix mathematics, theory facts and formulas.* Princeton University Press.

Bianchi, D., M. Buchner, and A. Tamoni (2021). Bond risk premiums with machine learning. *Review of Financial Studies 34*, 1046–1089.

Callot, L., M. Caner, O. Onder, and E. Ulasan (2021). A nodewise regression approach to estimating large portfolios. *Journal of Business and Economic Statistics 39*, 520–531.

Caner, M., M. Medeiros, and G. Vasconcelos (2022). Sharpe ratio analysis in high dimensions: Residual based nodewise regression in factor models. *Journal of Econometrics Forthcoming.*

Chen, L., M. Pelger, and J. Zhu (2021). Deep learning in asset pricing. arxiv:1904.00745v6, arXiv.

DeMiguel, V., L. Garlappi, and R. Uppal (2007). Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *The review of Financial studies 22*(5), 1915–1953.

Fama, E. F. and K. R. French (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics 33*(1), 3–56.

Fan, J., A. Furger, and D. Xiu (2016). Incorporating global industrial classification standard into portfolio allocation: A simple factor-based large covariance matrix estimator with high frequency data. *Journal of Business and Economic Statistics 34*, 489–503.

Fan, J., T. Ke, Y. Liao, and A. Neuhierl (2022). Structural deep learning in conditional asset pricing. *Princeton University Working Paper.*

Fan, J., Y. Liao, and M. Mincheva (2011). High-dimensional covariance matrix estimation in approximate factor models. *The Annals of Statistics 39*, 3320–3356.

Fan, J., Y. Liao, and M. Mincheva (2013). Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 75*(4), 603–680.

Fan, J., R. Masini, and M. Medeiros (2021). Bridging factor and sparse models. arxiv:2102.11341, arXiv.

Farrell, M. H., T. Liang, and S. Misra (2021). Deep neural networks for estimation and inference. *Econometrica 89*(1), 181–213.

Freyberger, J., A. Neuhierl, and M. Weber (2020). Dissecting characteristics nonparametrically. *Review of Financial Studies 33*, 2326–2377.

Gagliardini, P., E. Ossola, and O. Scaillet (2016). Time-varying risk premium in large cross-sectional equity data sets. *Econometrica 84*, 985–1046.

Gagliardini, P., E. Ossola, and O. Scaillet (2019). A diagnostic criterion for approximate factor structure. *Journal of Econometrics 212*, 503–521.

Gagliardini, P., E. Ossola, and O. Scaillet (2020). Estimation of large dimensional conditional factor models in finance. *Handbook of Econometrics 7A*, 219–282.

Giglio, S. and D. Xiu (2021). Asset pricing with omitted factors. *Journal of Political Economy 129*, 1947–1990.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning. Adaptive Computation and Machine Learning.* MIT press.

Gu, S., B. Kelly, and D. Xiu (2021). Autoencoder asset pricing models. *Journal of Econometrics 222*(1), 429–450.

Horn, R. and C. Johnson (2013). *Matrix Analysis.* Cambridge University Press.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kohler, M. and S. Langer (2021). On the rate of convergence of fully connected deep neural network regression estimates. *Annals of Statistics 49*, 2231–2250.

Ledoit, O, M. and M. Wolf (2003). Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance 10*, 603–621.

Ledoit, O, M. and M. Wolf (2004). A well conditioned estimator for large dimensional covariance matrices. *Journal of Multivariate Analysis 88*, 365–411.

Ledoit, O, M. and M. Wolf (2017). Nonlinear shrinkage of the covariance matrix for portfolio selection: Markowitz meets goldilocks. *Review of Financial Studies 30*, 4349–4388.

Li, J. (2015). Sparse and stable portfolio selection with parameter uncertainty. *Journal of Business & Economic Statistics 33*(3), 381–392.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks 61*, 85–117.

Schmidt-Hieber, J. (2017). Nonparametric regression using deep neural networks with relu activation function. *Available at arXiv:1708.06633v2*.

Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with relu activation. *Annals of Statistics 48*, 1875–1898.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research 15*(1), 1929–1958.

Vershynin, R. (2019). *High Dimensional Probability: An Introduction with Applications in Data Science.* Cambridge University Press.

Wainwright, M. (2019). *High Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press.

Zhong, Q., J. Mueller, and J. Wang (2022). Deep learning for the partially linear cox model. *Annals of Statistics 50*, 1348–1376.