

Deep Learning for Predicting Market Excess Returns: The Role of Cross-Sectional Predictors

Junlong Xu * Mengyao Yuan †

Supervisor: Tobias Sichert

Abstract

Our study aggregates firm-level cross-sectional variables into time-series predictors and compares the out-of-sample performance of linear and deep learning models in predicting market excess returns. We find that the deep learning model outperforms linear models by capturing nonlinear structures and interactions often missed by traditional methods. Additionally, our results demonstrate that the cross-sectional predictors provide systematic information that substantially overlaps with the time-series predictors. Within the deep learning framework, variables related to stock trading activity and firms' asset structures are the most influential contributors to model predictions. These findings highlight the potential of using deep learning to extract cross-sectional information in improving the predictability of market premiums.

Keywords: Deep Learning, Neural Networks, Return Predictability, Market Excess Return, Empirical Asset Pricing

*42682@student.hhs.se
†42677@student.hhs.se

Acknowledgments

We would like to express our deepest gratitude to our supervisor, Tobias Sichert, for his invaluable guidance and insight throughout this thesis. His constructive feedback has been instrumental in shaping this research. We also want to thank our friends and family for their unwavering support.

Contents

1 Introduction **4**

2 Literature Review **7**

2.1 Time-series vs. cross-sectional return predictability 7

2.2 The rise of machine learning 9

3 Methodology **11**

3.1 Simple linear regression 11

3.2 Dimension reduction: PCR and PLS 12

3.3 Ridge regression 14

3.4 GLM with group lasso 15

3.5 Neural network 17

3.6 Statistical performance 20

3.7 Variable importance 21

4 Empirical Analysis **23**

4.1 Data and empirical applications 23

4.2 Model performance with validated predictors 25

4.3 Model performance with full cross-sectional predictors 29

4.4 Model performance with cross-sectional and time-series predictors 31

4.5 Variable importance under deep learning 32

4.6 Interaction effects 36

4.7 Economic implications 38

5 Conclusion **40**

References **42**

A Tables and Figures **46**

B Details of Firm-level Predictors **50**

C Algorithms and hyperparameters **54**

D AI disclosure **57**

1 Introduction

The predictability of equity premiums has long been a major issue in the field of financial asset pricing. As far back as 1920, Dow (1920) explored this question by examining the predictive role of dividend ratios. Over the decades, a large literature has extended this research to identify a wide range of time-series and cross-sectional predictors, with the aim of uncovering consistent patterns or signals that could improve the predictability of future stock returns. Valuation ratios (Campbell and Shiller, 1988), liquidity measures (Amihud, 2002), volatility indicators (Guo and Whitelaw, 2006), and macroeconomic factors (Fama and French, 1989), have all been shown to contribute to return predictions. More recently, the emphasis has switched to evaluating aggregated cross-sectional predictors. This emerging line of research investigates whether aggregated firm-level characteristics can capture systematic information relevant to forecasting market returns (Karabiyik and Westerlund, 2021, Engelberg et al., 2023).

However, the real-world application of these predictors is frequently called into question due to weak out-of-sample performance. For example, Welch and Goyal (2008) examined 17 variables commonly considered to be effective predictors of the equity premium and found that few consistently performed well both in and out of sample. Similarly, an empirical study by Engelberg et al. (2023), after analyzing 140 various aggregated cross-sectional predictors, showed limited evidence to support their out-of-sample predictive power. While these empirical studies based on traditional linear models provide valuable insights, the increasing availability of large datasets poses new challenges, such as multicollinearity and overfitting. These issues can severely affect the reliability of predictions, especially when models are not designed to handle high-dimensional data with complex interdependencies.

The rise of machine learning offers promising solutions to these challenges. Advanced methods such as regression trees and neural networks are well-suited to capturing complicated patterns and interactions within high-dimensional data. Neural networks, in particular, have demonstrated superior out-of-sample predictive performance compared to traditional linear models in both time-series and cross-sectional forecasting. For instance, studies by Feng et al. (2018) and Gu et al. (2020) showed that deep learning models generally outperform linear models in predicting equity premiums, especially when accounting for time series predictors.

Despite these advancements, current literature has yet to fully explore whether aggregated cross-sectional variables, in the context of nonlinear models, can be as effective as time series variables in return prediction.

To address this gap, our study focuses on two key questions: (1) Can aggregated cross-sectional characteristics effectively predict market excess returns within a deep learning framework? (2) Do these cross-sectional predictors contain unique systematic information or share overlapping information with time-series macroeconomic variables?

We first construct a time-series sample by aggregating cross-sectional variables obtained from Chen and Zimmermann (2022). For each cross-sectional variable, we compute both equal-weighted and value-weighted averages, resulting in two distinct sets of time-series samples.[1] Each sample consists of 105 variables in total, representing a comprehensive set of firm-level characteristics that have been demonstrated in the academic literature to exhibit cross-sectional return predictability.[2] Of these, 14 variables have already been aggregated in existing studies to forecast market returns. We refer to these as the "validated variables" throughout the paper. The dataset spans monthly observations from December 1953 to December 2023. Consistent with Welch and Goyal (2008), market excess returns are calculated as the continuously compounded return on the S&P 500 index minus the 1-month Treasury bill rate.

In our empirical analysis, we begin with the benchmark prediction using 14 validated variables as the smallest time-series sample and progressively expand the dataset to include the full set of aggregated cross-sectional predictors and time-series macroeconomic variables. We compare predictive performance across a range of methods, including simple linear regression, principal component regression (PCR), partial least squares (PLS), ridge regression, generalized linear models (GLM) with group lasso, and neural networks. Specifically, we use a fixed 50-year moving window to predict market returns over 1-, 3-, and 12-month horizons, and compute their out-of-sample R^2. In addition, we evaluate the importance of each variable by measuring its impact on the model's forecasts. Finally, to intuitively demonstrate the economic value of improved predictability, we calculate the Sharpe ratio improvements achieved by incorporating predictive information into portfolio strategies relative to a buy-and-hold strategy.

[1] We follow Engelberg et al. (2023) to construct aggregated predictors from the cross-sectional data, including both equal-weights and value-weights.

[2] Detailed descriptions of all cross-sectional predictors used in our paper are provided in Appendix B.

Based on the empirical study, our main findings are threefold.

First, we find that deep learning models outperform traditional models in out-of-sample prediction and exhibit improved stability. By examining the relationship between the hidden-layer factors extracted from neural networks and the predictors, as well as the interactions between certain variables, we further confirm that the superior predictive performance of neural networks is related to their ability to capture nonlinear structures and complex interactions within the data.

Second, average cross-sectional predictors prove effective in predicting market excess returns within the deep learning framework, with value-weighted averages beating equal-weighted averages. Incorporating time-series macroeconomic variables yields only marginal improvements in predictive performance, suggesting that cross-sectional predictors already capture overlapping information. The effectiveness of these predictors is reflected in the economic gains realized through trading strategies based on neural network predictions. In particular, these strategies achieve an improvement of over five percentage points in Sharpe ratio relative to a buy-and-hold strategy.

Finally, we examine variable importance within the deep learning models and find that variables associated with stock trading activity and firms' asset structures contribute the most to the model prediction. In addition, variables related to risk, valuation, and profitability also show consistent importance across models, collectively enhancing the model's ability to forecast market returns.

The remainder of this paper is organized as follows: Section 2 briefly reviews the literature on return predictability and machine learning methods. Section 3 details various prediction-related methods. Section 4 describes the data and presents the empirical results, and Section 5 concludes.

2 Literature Review

For over a century, a substantial body of theoretical and empirical studies has sought to uncover and exploit patterns in market and firm-specific characteristics to improve the accuracy of return forecasts. The existing literature predominantly follows two main strands: the development of predictive variables and the advancement of forecasting methodologies. This study builds upon and extends these two lines of research.

2.1 Time-series vs. cross-sectional return predictability

Early studies on return predictability focused on utilizing time-series models to forecast market returns based on macroeconomic and financial indicators. Seminal contributions in this area demonstrated that valuation ratios, such as the dividend-to-price ratio (Campbell and Shiller, 1988) and the book-to-market ratio (Kothari and Shanken, 1997), could predict market returns over a longer horizon, providing evidence of return predictability. Fama and French (1989) extended these insights by connecting stock returns to macroeconomic conditions, particularly through the default and term spreads, which highlighted the cyclical nature of stock returns. Moving into the early 2000s, researchers began to explore more sophisticated predictors. Lettau and Ludvigson (2001) introduced the consumption-wealth ratio (CAY) as a more effective predictor than traditional financial ratios. Additional studies examined predictors like stock variance (Guo and Whitelaw, 2006), cross-sectional premiums (Polk et al., 2006), and corporate issuing activity (Fama and French, 2008). Moreover, bond market variables, such as default spreads and corporate bond yields, were analyzed by Campbell (1987), Hodrick (1992), and Cochrane et al. (2005), confirming their predictive strength. Together, these studies identified a broad array of variables, spanning valuation ratios, bond yields, and macroeconomic indicators, that have been used to forecast stock market returns.

Despite these developments, skepticism regarding the practical utility of time-series predictors has grown. Welch and Goyal (2008) critically evaluated 17 common predictors and found that most failed to outperform simple historical averages in out-of-sample tests. Earlier, Bossaerts and Hillion (1999) raised concerns about overfitting, showing that many predictors did not generalize well outside their sample periods. Cooper and Gulen (2006) further demon-

strated that return predictability was highly sensitive to the choice of variables and model specifications. More recently, Goyal et al. (2024) revisited this issue by testing 29 new variables published in the past two decades and found only a few perform reasonably well both in-sample and out-of-sample. Collectively, these critiques underscore the limitations of traditional time-series forecasting.

Although studies have generally found limited out-of-sample success for time series forecasts, extensive literature supports the predictability of cross-sectional stock returns. Early work by such as Fama and French (1992), introduced a three-factor model incorporating market risk, firm size, and value factors, which outperformed the traditional capital asset pricing model (CAPM). Subsequent research expanded these models: Jegadeesh and Titman (1993) identified the momentum effect, while Fama and French (1996) added profitability and investment factors, significantly enhancing the explanatory power of these models. These findings highlight the persistent impact of firm-specific characteristics on stock returns, and newer studies have reaffirmed this conclusion. For example, McLean and Pontiff (2016) showed that nearly 88% of firm-level predictors still generate significant t-statistics (greater than 1.5) using quintile portfolios over the extended sample period.

However, as with time series forecasts, there has been an evolving discussion about the credibility of cross-sectional return predictability. Lewellen and Shanken (2000)'s work suggested that commonly used valuation ratios, such as dividend-to-price, earning-to-price, and book-to-market ratios, often failed to produce consistent out-of-sample predictive power. Another focus is mainly on the impact of data mining. Linnainmaa and Roberts (2018) showed that most accounting-based anomalies are the result of data mining. In contrast, McLean and Pontiff (2016) analyzed 97 anomalies and found that, at most, only 25% of the observed anomaly returns could be attributed to data mining.

In recent years, research has gradually started to focus on the relation between cross-sectional predictors and aggregate market returns. As we mentioned in the last section, 14 cross-sectional variables have been validated by aggregating to predict market returns in some published research. Example predictors include leverage (Baker and Wurgler, 2000), accruals (Hirshleifer et al., 2009), and momentum (Moskowitz et al., 2012). Lately, Karabiyik and Westerlund (2021) concluded that simple cross-sectional averages outperform the predictive

performance of estimated principal component factors. In comparison, Engelberg et al. (2023) conducted a comprehensive analysis of 140 firm-level cross-sectional predictors and found weak evidence suggesting that cross-sectional variables contain systematic information relevant to time-series forecasts.

Overall, the question of whether cross-sectional return predictability exists, and how it operates, remains a significant and unresolved debate in finance. However, no studies have yet looked at the performance of aggregated cross-sectional variables under nonlinear models, which is the focus of our paper.

2.2 The rise of machine learning

The aforementioned literature on predicting equity premiums primarily relies on traditional econometric methods. However, when confronted with high-dimensional input variables, these approaches have severe limitations, such as multicollinearity, noise interference, and challenges in variable selection. In particular, regression and portfolio sorting techniques are inadequate for processing the expanding set of predictive variables accumulated in the literature over the past few decades. As a result, there has been a growing interest in machine learning, which offers tools capable of handling complex data structures and extracting meaningful patterns often missed by traditional models.

Machine learning has started to appear frequently in the asset pricing literature. For example, Khandani et al. (2010) and Butaru et al. (2016) used regression trees to predict consumer credit risk, showing the flexibility of tree-based methods in capturing nonlinear patterns. Expanding on the application of tree-based models, Moritz and Zimmermann (2016) utilized these techniques for portfolio sorting, verifying their utility in optimizing asset allocation. Dimension reduction techniques, such as principal component analysis (PCA) used by Kelly et al. (2019) and Giglio and Xiu (2021), have proven effective in estimating and testing factor pricing models. These tools reduce complexity by identifying the most informative factors. Additionally, methods like shrinkage and variable selection have been employed by Kozak et al. (2020) and Freyberger et al. (2020) to approximate stochastic discount factors and nonlinear functions for expected returns. In a related study, Rapach et al. (2013) adopted LASSO to

predict global equity market returns using lagged returns from multiple countries, highlighting the capability of regularization techniques to manage high-dimensional datasets.

Among various machine learning methods, deep learning, namely neural networks, has become widely used in finance. Early studies by Hutchinson et al. (1994) and Yao et al. (2000) demonstrated their potential in predicting option prices. More recently, Sadhwani et al. (2021) applied deep learning to forecast multi-period mortgage risks, including delinquency, foreclosure, and prepayment, while Heaton et al. (2017) developed neural networks for optimizing portfolio selection. Beyond these applications, neural networks have shown promise in both market and cross-sectional stock return forecasting. For example, Chen et al. (2024) employed multiple neural networks, including the long short-term memory (LSTM) network, to estimate an asset pricing model for individual stock returns. Feng et al. (2018) used deep learning models to forecast market excess returns, while Gu et al. (2020) applied these techniques to predict cross-sectional stock returns. Both studies conducted a systematic comparison between linear and nonlinear models, confirming the superior out-of-sample performance of neural networks over traditional methods. Similarly, Bianchi et al. (2021) extended these findings to the bond market, using neural networks to predict bond risk premiums.

Building on insights from the existing literature on return predictability and machine learning, our study aims to fill the gap in the application of average cross-sectional variables under nonlinear models. By leveraging these predictors to forecast market returns in the context of deep learning, we extend the empirical research on market return prediction. Specifically, we systematically explore the effectiveness of neural networks in utilizing aggregated firm characteristics to improve market return forecasts. This approach provides a new perspective on integrating micro-level firm data into macro-level market forecasts, showing that deep learning models can effectively harness the rich information contained in cross-sectional variables.

3 Methodology

This section introduces all methodologies used in our analysis, drawing on the framework of Gu et al. (2020), which provides a foundation for predictive modeling. These methods address a variety of challenges, from handling high-dimensional predictor sets to capturing nonlinear interactions between variables. Each subsection describes a particular statistical model for predicting risk premiums and explains its related settings, such as various regularization techniques and computational algorithms. To ensure consistent evaluation across models, we have a common training goal: minimizing the mean squared prediction error (MSE) to improve prediction accuracy.

As a starting point, we represent the market excess return using an additive prediction error model, expressed in its most general form as follows:

$$R_{t+1} = E_t(R_{t+1}) + \epsilon_{t+1}, \tag{1}$$

where $E_t(R_{t+1})$ denotes the conditional expectation of the market excess return given all available information at time t, and ϵ_{t+1} represents the unpredictable error term. The conditional expectation $E_t(R_{t+1})$ is modeled as a function of X_t:

$$E_t(R_{t+1}) = g^*(X_t). \tag{2}$$

where X_t represents the N-dimensional predictor vector in month $t = 1, \ldots, T$. The function $g^*(\cdot)$ defines the mapping between the predictors and the expected return.

Our primary objective is to accurately estimate $E_t(R_{t+1})$ by using various predictive models, each specifying a distinct functional form for $g^*(\cdot)$. These models aim to capture systematic variation in returns and maximize the out-of-sample explanatory power of the realized R_{t+1}.

3.1 Simple linear regression

We start our research with the ordinary least squares (OLS) regression, one of the most widely used methods in predictive modeling. OLS provides a baseline framework to capture the simple linear relationship between dependent and independent variables. Although its simplicity

makes it interpretable, OLS may exhibit limited predictive performance in high-dimensional data due to issues such as multicollinearity. Nonetheless, we apply this basic model as a comparison point for more sophisticated methods.

The conditional expectation $g^*(\cdot)$ under the OLS regression model is specified as:

$$g(X_{i,t};\theta) = \theta_0 + \sum_{i=1}^{N} X_{i,t}\theta_i, \tag{3}$$

where θ_0 represents the intercept term, and θ_i denotes the coefficients associated with each predictor $X_{i,t}$. This linear functional form assumes that the relationship between predictors and the expected return is additive and linear, without accounting for potential interactions or nonlinear effects among predictors.

The coefficients $\theta = (\theta_0, \theta_1, \ldots, \theta_N)$ are estimated by minimizing the model's MSE, which serves as the loss function:

$$MSE = \frac{1}{T}\sum_{t=1}^{T}(R_{t+1} - \hat{R}_{t+1})^2, \tag{4}$$

where R_{t+1} represents the actual value of market risk premium, and $\hat{R}_{t+1}$ denotes the predicted value at time $t+1$.

3.2 Dimension reduction: PCR and PLS

As discussed in the last subsection, the simple linear model is likely to fail when faced with a large number of predictors. As the number of predictors N approaches the number of observations T, the model tends to fit the noise rather than capture the true signal. This is particularly problematic in return prediction, where the signal-to-noise ratio is inherently low.

To address this challenge, dimension reduction techniques are crucial, as they help reduce the number of estimated parameters while preserving relevant information. Therefore, we introduce two frequently used approaches: principal component regression (PCR) and partial least squares (PLS). Both methods aim to reduce the dimensionality and mitigate strong multicollinearity between predictors, although they differ in principle and application.

PCR is a two-step procedure consisting of principal component analysis (PCA) followed by

a standard predictive regression (Jolliffe, 2003). In the first step, PCA is applied to the predictors, condensing multiple independent variables into a few principal components (PCs) based on their covariance structure. In the second step, these extracted PCs serve as independent variables in an OLS regression to predict the market risk premium. Essentially, PCR regularizes the prediction problem by assigning zero coefficients to low-variance components.

However, a potential limitation of PCR is that it considers only the covariance of predictor matrices when generating PCs, and the target variable is included in the model at the final regression step. As a result, PCR may overlook components that are weakly correlated with other predictors but strongly associated with the response variable. In contrast to PCR, PLS takes into account the covariance between the predictors and the response variable during the construction of components. By aligning the dimension reduction process with the ultimate prediction objective, PLS identifies linear combinations of predictors that maximize their relevance to the target variable.

To apply PCR and PLS, we first convert Equations (1)-(3) into a vectorized form:

$$R = X\theta + E, \tag{5}$$

where R is the $T \times 1$ response vector of R_{t+1}, X is the $T \times N$ matrix of stacked predictors, θ is the parameter vector, and E is the vector of error terms.

Condensing the set of predictors from dimension N to a much smaller number of P, we can now express the forecasting model as:

$$R = (X\Omega_P)\theta_P + \tilde{E}, \tag{6}$$

where Ω_P is the $N \times P$ weight matrix of linear combinations, with columns w_1, w_2,..., w_P. Thus, $X\Omega_P$ is the matrix of predictors after reducing dimension. Likewise, the predictive coefficient θ_P is now a $P \times 1$ vector rather than $N \times 1$.

PCR derives the weight matrix Ω_P recursively. The weights of jth linear combination are

determined by solving:

$$
\begin{aligned}
& w_j = \arg\max_{w} \operatorname{Var}(Xw), \\
& \text{s.t.} \quad w'w = 1, \quad \operatorname{Cov}(Xw, Xw_l) = 0, \quad l = 1, 2, \ldots, j-1.
\end{aligned}
\tag{7}
$$

The objective function above shows that PCR retains the components that explain the maximum variance without considering the target variable at all. Unlike PCR, PLS incorporates prediction objectives directly into the optimization process. To construct the jth component, the weights solve:

$$
\begin{aligned}
& w_j = \arg\max_{w} \operatorname{Cov}^2(R, Xw), \\
& \text{s.t.} \quad w'w = 1, \quad \operatorname{Cov}(Xw, Xw_l) = 0, \quad l = 1, 2, \ldots, j-1.
\end{aligned}
\tag{8}
$$

Some researchers have argued that PLS performs better than PCR due to its improved covariance between the dependent and independent variables. However, there is also concern that such improvement could lead to unintended consequences. For instance, when there is a high covariance between X and R, it may cause the estimated coefficients to deviate from their true values.

For both methods, the number of principal components P is a hyperparameter, which is selected by time series cross-validation.

3.3 Ridge regression

In statistics, approaches to handle high-dimensional data can be categorized in two directions: the first is dimension reduction, as we described in the last subsection, and the second is shrinkage. Shrinkage methods play a critical role in achieving various statistical inference goals like improving numerical stability, enhancing model interpretability, and mitigating overfitting. Frequently used penalized linear models include Lasso, Ridge, and Elastic Net, which shrink the estimated coefficients towards zero by appending a penalty to the original loss function:

$$
\mathcal{L}(\theta; \cdot) = \mathcal{L}(\theta) + \phi(\theta; \cdot). \tag{9}
$$

The penalty function $\phi(\theta;\cdot)$ differs between methods. In our model, we focus on ridge regression (Ridge) proposed by Hoerl and Kennard (1970). Ridge is a popular model in the finance literature. For example, Theoharidis et al. (2023) used Ridge as the benchmark model to address nonlinearities in forecasting inflation, while Kalatzis et al. (2011) applied it to solve multicollinearity issues when analyzing the investment decisions of 373 large Brazilian firms. These studies demonstrate Ridge's versatility in producing stable and reliable coefficient estimates.

Specifically, Ridge uses squared Euclidean distance (or l_2 regularization) in its penalty function, which takes the form:

$$\phi(\theta;\lambda) = \lambda \sum_{j=1}^{N} \theta_j^2, \tag{10}$$

where λ is the l_2 regularization parameter that controls the strength of the penalty term. We identify the optimal value of the regularization parameter λ using a systematic approach that leverages cross-validation.

3.4 GLM with group lasso

A model's forecast error can be decomposed into three parts including approximation error, estimation error, and intrinsic error:

$$R_{t+1} - \hat{R}_{t+1} = \underbrace{g^*(X_t) - g(X_t;\theta)}_{\text{approximation error}} + \underbrace{g(X_t;\theta) - g(X_t;\hat{\theta})}_{\text{estimation error}} + \underbrace{\epsilon_{t+1}}_{\text{intrinsic error}}, \tag{11}$$

where $g^*(X_t)$ is the true model, $g(X_t;\theta)$ is the model specified by researchers, and $g(X_t;\hat{\theta})$ is the fitted model.

When the true model is complex and nonlinear, relying on a linear model for estimation will artificially introduce approximation errors due to model misspecification. These errors can be mitigated by employing more flexible model specifications that better capture the underlying complexities. Estimation errors, arising from sampling variation, are data-dependent. While increasing the number of observations can reduce these errors, it is often beyond the econometrician's control. Finally, there is an intrinsic error, which cannot be minimized because it is an unpredictable component associated with the randomness of news and markets.

Taking into account the possible nonlinearity between the data, we introduce nonparametric models in the following subsections. The first model we consider is the generalized linear model (GLM), which extends the simple linear model to incorporate potential nonlinear relationships between variables. Thus, GLM serves as the closest nonlinear model to the linear approaches described above. In contrast to the linear model, GLM imposes fewer assumptions on the data, thereby reducing the approximation errors associated with model misspecification. This flexibility makes GLM applicable to a broader range of research contexts.

To introduce nonlinear structures of the original input variables, we apply the K-term spline transformation of order two to X:

$$g(X;\theta,p(\cdot)) = \sum_{i=1}^{N} p(X_i)'\theta_i \tag{12}$$

where $p(\cdot) = (p_1(\cdot), p_2(\cdot), ..., p_K(\cdot))'$ is a vector of basis functions, and $\theta = (\theta_1, \theta_2, ..., \theta_N)$ is a $K \times N$ matrix of parameters. Specifically, $p(\cdot)$ includes terms $(1, X, (X - c_1)^2, (X - c_2)^2, ..., (X - c_{K-2})^2)$, where $c_1, c_2, ..., c_{K-2}$ are knots.

Due to the rapid increase in the number of model parameters caused by series expansion, we apply group lasso penalization to regulate the degrees of freedom. Group lasso is well-suited for spline transformations, as it either selects all K spline terms associated with a given predictor or excludes them entirely. It takes the form:

$$\phi(\theta;\lambda) = \lambda \sum_{j=1}^{N} \left(\sum_{k=1}^{K} \theta_{j,k}^2 \right)^{1/2}, \tag{13}$$

Heavy tails are a common feature of the distribution of asset returns, implying a high probability of extreme events. The convexity of the least squares objective (4) overemphasizes large errors, so outliers are more likely to destabilize the model predictions. To improve robustness against such outliers, we incorporate the Huber loss function, which seamlessly combines squared error for smaller residuals and absolute error for larger residuals. The Huber loss is

defined as follows:

$$H(R-\hat{R},\delta)=\begin{cases}\frac{1}{2}(R-\hat{R})^2 & \text{if } |R-\hat{R}|\leq\delta\\ \delta\left(|R-\hat{R}|-\frac{1}{2}\delta\right) & \text{if } |R-\hat{R}|>\delta\end{cases}, \tag{14}$$

where δ is a tuning parameter that determines the transition point between the two loss functions. Therefore, to optimize GLM, we minimize an objective function, which takes the form of:

$$\mathcal{L}(\theta)=\sum_{t=1}^{T}H(R_{t+1}-g(X_{t+1};\theta,p(\cdot)),\delta)+\phi(\theta;\lambda). \tag{15}$$

3.5 Neural network

Neural networks are known for their flexibility, which comes from their capacity to integrate multiple nested layers of nonlinear predictor interactions. Commonly referred to as "deep learning," this approach represents a type of supervised learning for predicting an output variable Y through input variables X.

Our analysis focuses on traditional "feed-forward" networks or multi-layer perceptrons (MLPs). These networks consist of a minimum of three layers: an input layer comprising raw predictors, a hidden layer for nonlinear transformation, and an output layer that generates an outcome prediction. As in the human brain, the layers of neural networks represent groups of "neurons", with connections between layers serving as "synapses" that facilitate signal transfer between neurons in different layers. Figure 1 illustrates two representative examples.

Consider the left panel of Figure 1 as an example, the number of units in the input layer is corresponding to the dimension of the predictor variable, which is specified as four (represented as $X_1,...,X_4$). The hidden layer, placed between the input and output, contains five neurons. Each neuron receives the signals linearly from the input layer and then transforms it nonlinearly through an activation function f before transmitting it to the next layer. Specifically, the first neuron in the hidden layer transforms inputs into an output as $X_1^{(1)}=f\left(\theta_{1,0}^{(0)}+\sum_{j=1}^{4}X_j\theta_{1,j}^{(0)}\right)$, where θ is a parameter vector including an intercept and one weight parameter per input variable. Finally, the outputs from each neuron are combined linearly to produce the final forecast result $g(X,\theta)=\theta_0^{(1)}+\sum_{j=1}^{5}X_j\theta_j^{(1)}$.

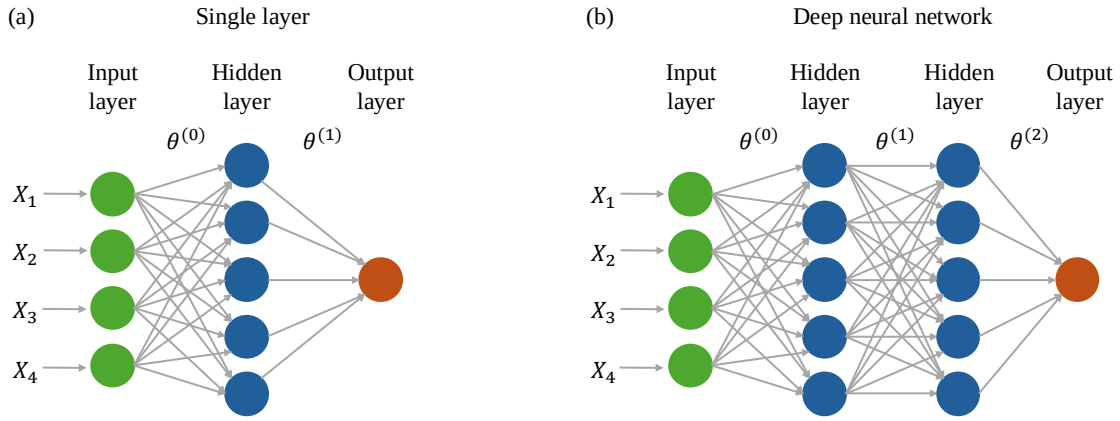

Figure 1. Neural networks.
This figure illustrates two simple neural networks: one with a single hidden layer and the other with two hidden layers. The input layer is represented by green circles, the hidden layers by blue circles, and the output layer by red circles. Each unit in the hidden layers applies a nonlinear activation function to its inputs before passing the output to the next layer. The outputs of the final hidden layer are linearly combined to produce the final prediction.

Deep neural networks incorporate hidden layers between the input and output to capture more complex patterns and enhance predictive performance. The number of hidden layers and the number of neurons in each layer represent the depth and width of the architecture, respectively. When constructing a neural network, multiple design choices need to be made, such as the number of hidden layers, the number of neurons per layer, and which units to connect.

The machine learning literature has extensively explored the effects of network width and depth. Eldan and Shamir (2016) demonstrated that, in standard feed-forward neural networks, increasing the depth by even a single layer can provide exponentially more benefits than expanding the network's width. However, for small datasets, simpler networks with only a few layers and nodes often yield better results. Training deep neural networks is challenging due to the substantial number of parameters, the highly nonconvex nature of the objective function, and the risk of gradient explosion or vanishing during the recursive computation of derivatives.

Selecting network architectures through cross-validation may not be an ideal approach because it disrupts the time sequence and does not accurately reflect the model's performance on future data. Gu et al. (2020) argues that continuously searching for the optimal architec-

ture is impractical. Following their approach, we pre-determine various network structures for comparison, which reflects a reasonable level of deep learning performance.

We select four different neural network architectures, labeled NN1 to NN4. NN1, the most complex architecture, consists of four hidden layers with 32, 16, 8, and 4 neurons, respectively. Next, in decreasing order, NN2 has three hidden layers with 32, 16, and 8 neurons, respectively; NN3 has two hidden layers with 32 and 16 neurons, respectively; and NN4 has only one hidden layer with 32 neurons. The number of neurons in each layer follows the geometric pyramid rule, as outlined by Masters (1993). All architectures are fully connected, meaning each unit receives input from every unit in the preceding layer.

The nonlinear transformation is implemented via an activation function. Common activation functions include sigmoid, softmax, and the rectified linear unit (ReLU). ReLU has gained significant popularity in recent literature due to its simplicity and effectiveness. It is defined as:

$$\text{ReLU}(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{otherwise} \end{cases}. \tag{16}$$

In this study, ReLU is applied uniformly to all nodes throughout the network.

Mathematically, our deep neural network has the following general formula. Let $X_k^{(l)}$ be the output of node k in each layer $l = 1, \ldots, L$. Then, the vector of outputs for this layer is defined as $X^{(l)} = \left(1, X_1^{(l)}, \ldots, X_K^{(l)}\right)'$, where K is the total number of neurons in this layer. The constant (1) is included to integrate bias terms seamlessly into matrix operations without additional computation steps. Thus, to initialize the network, we have the input layer consisting of all raw variables, $X^{(0)} = \left(1, X_1, \ldots, X_N\right)'$. The recursive output of each neuron in the layer $l > 0$ is typically computed as:

$$X_k^{(l)} = \text{ReLU}\left(X^{(l-1)'}\theta_k^{(l-1)}\right)^{3}, \tag{17}$$

with the final prediction

$$g(X, \theta) = X^{(L-1)'}\theta^{(L-1)}. \tag{18}$$

[3]The expanded form is given by $X_k^{(l)} = \text{ReLU}\left(\theta_{k,0}^{(l-1)} + \sum_{j=1}^{K^{(l-1)}} X_j^{(l-1)}\theta_{k,j}^{(l-1)}\right)$.

We estimate the weight parameters θ of the deep learning model by minimizing a penalized l_2 objective function of the prediction errors. A notable advantage of training a neural network is the ability to update model parameters jointly at each optimization step. Given the computational demands of brute-force optimization, we use the commonly used stochastic gradient descent (SGD) algorithm in TensorFlow for training.

Due to the highly nonlinear and parametric nature of neural networks and to mitigate the risk of overfitting, we incorporate additional regularization techniques alongside the l_2 penalization of the weight parameters in our estimation: dropout, batch normalization, early stopping, and ensembles.[4]

Dropout (Srivastava et al., 2014) randomly removes units and their connections in the neural network with a given probability p, effectively preventing co-adaptation among units. Next, batch normalization (Ioffe and Szegedy, 2015) addresses internal covariate shifts by standardizing the inputs to each hidden layer, which not only stabilizes the training process but also accelerates convergence. To further control overfitting, early stopping halts training if the loss on the validation sample has not improved. Finally, we use an ensemble approach[5] (Hansen and Salamon, 1990) that constructs the final prediction by averaging forecasts from networks initialized with different random seeds. This reduces the prediction variance, as the stochastic nature of optimization can lead to varying forecasts across different seeds.

3.6 Statistical performance

Following Campbell and Thompson (2008), we compare the predictions from each method to the historical mean of market excess returns. Specifically, we calculate the out-of-sample R^2 as

$$R^2_{\text{oos}} = 1 - \frac{\sum_{t=0}^{T} \left(R_{t+1} - \hat{R}_{t+1} \right)^2}{\sum_{t=0}^{T} \left(R_{t+1} - \bar{R}_{t+1} \right)^2}, \tag{19}$$

where $\hat{R}_{t+1}$ is the forecast of a predictive model, and $\bar{R}_{t+1}$ is the prediction based on the historical mean. A positive R^2_{oos} means that the predictive model achieves a lower mean squared

[4]See Appendix C for relevant algorithms.

[5]Due to computational constraints, the parameter tuning and ensemble methods applied in this study are not exhaustive. As a result, our findings represent a lower bound of the potential performance of neural networks.

prediction error than the historical average.

Although R^2_{oos} tends to be small for many predictive models, Campbell and Thompson (2008) argue that even a marginally positive R^2_{oos} can still hold economic significance in the context of predicting portfolio returns. Specifically, they show that the Sharpe ratio (SR^*) achieved by an active investor leveraging predictive information (as captured by R^2_{oos}) surpasses the Sharpe ratio (SR) of a buy-and-hold strategy. The relationship is given by the formula:

$$SR^* = \sqrt{\frac{SR^2 + R^2_{\text{oos}}}{1 - R^2_{\text{oos}}}}, \tag{20}$$

where SR is defined as:

$$SR = \frac{R_P - R_f}{\sigma_P}. \tag{21}$$

Here, R_P represents the portfolio's average return, R_f denotes the risk-free rate, and σ_P is the standard deviation of the portfolio's excess returns.

To assess the statistical significance of differences in out-of-sample predictive performance between models, we implement the Diebold and Mariano (1995) test. This test is suitable for pairwise comparisons of forecast errors, particularly when the time-series dependence in returns is relatively weak. Formally, let $\hat{e}_{1,t}$ and $\hat{e}_{2,t}$ denote the forecast errors of two models at time t. The loss differential is defined as $d_t = \hat{e}^2_{1,t} - \hat{e}^2_{2,t}$. Then, we compute the Diebold-Mariano test statistic as:

$$DM = \frac{\bar{d}}{\sigma(\bar{d})}, \tag{22}$$

where $\bar{d}$ and $\sigma(\bar{d})$ are the sample mean and Newey-West standard error of d_t, respectively. Under the null hypothesis $H_0 : E[d_t] = 0$, the test determines whether the predictive accuracies of the two models are statistically equivalent.

3.7 Variable importance

Machine learning models are growing in complexity and predictive accuracy. However, as they become more like "black boxes", it is more challenging to interpret how they arrive at specific outcomes. We aim to identify which variables significantly influence our model's predictions of market returns. To achieve this, we utilize SHapley Additive exPlanations (SHAP), a measure

grounded in cooperative game theory that quantifies the contribution of each predictor to the output of the model (Scott et al., 2017).

SHAP assigns each predictor a Shapley value, which represents its average marginal contribution to the model's prediction. This method ensures that the sum of all variable contributions is equal to the model's prediction, providing a consistent measure of variable importance. Mathematically, the Shapley value ϕ_i for an input variable X_i is defined as:

$$\phi_{val}(x_i) = \sum_{S \subseteq X \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left[val(S \cup \{i\}) - val(S) \right], \tag{23}$$

where $S \subseteq X \setminus \{i\}$ represents all possible subsets of features excluding X_i, and $val(S)$ denotes the model's prediction using only the subset S of features. The expression $val(S \cup \{i\})$ is the prediction when feature X_i is added to subset S. Thus, the difference $\Delta val(i, S) = val(S \cup \{i\}) - val(S)$ captures the marginal contribution of feature X_i. The normalization term $\frac{|S|!(N-|S|-1)!}{N!}$ adjusts for the number of possible subsets, ensuring that the contributions are weighted across all potential variable combinations.

4 Empirical Analysis

Our empirical analysis is structured around two primary research questions. The first question examines whether average cross-sectional characteristics can effectively predict market excess returns within a deep learning framework. The second question explores whether these cross-sectional predictors contain unique systematic information or share overlapping information with time-series macroeconomic variables.

In this section, we first describe the data used in our analysis. Then, we present the empirical results across various settings, followed by a comprehensive discussion of our findings.

4.1 Data and empirical applications

Our dataset consists of two main components: aggregated firm-level characteristics and time-series macroeconomic variables. It spans 70 years, from December 1953 to December 2023, with a monthly frequency. This period covers multiple market cycles, ensuring that the analysis captures variations in market dynamics and reflects diverse economic conditions.

The first component of the dataset comprises aggregated predictors derived from firm-level characteristics. These predictive signals are sourced from Chen and Zimmermann (2022),[6] a dataset of 209 predictors capturing a broad spectrum of firm-specific attributes, including but not limited to profitability, valuation, growth, and risk-related measures. We retain 105 variables that are consistently available throughout the study period and have shown return predictability in the academic literature. Among these, 14 predictors have been validated in prior studies for their effectiveness in forecasting market returns. Detailed descriptions of all firm-level characteristics are provided in Appendix B.

To convert the firm-level characteristics into time-series predictors, we compute equal-weighted and value-weighted averages for each characteristic. Specifically, the equal-weighted average assigns the same weight to each firm, providing an aggregate measure that represents the average attribute across all firms. In contrast, the value-weighted average weights each firm by its market capitalization,[7] thereby emphasizing the contributions of larger firms to the

[6]Data is downloaded from Andrew Y. Chen's website (https://www.openassetpricing.com/data/).

[7]We obtain price and outstanding share data from the Center for Research in Security Prices (CRSP) to calculate the market capitalization for each company.

aggregate measure. These aggregated factors serve as key inputs to the predictive models.

Time series predictors		**Validated cross-sectional predictors**	
Variable	**Description**	**Variable**	**Description**
b/m	Book-to-market ratio	Accruals	Accruals
d/e	Dividend payout ratio	AssetGrowth	Asset growth
d/p	Dividend-price ratio	BidAskSpread	Bid-ask spread
d/y	Dividend-yield ratio	BMdec	Book-to-market ratio
dfy	Default yield spread	EP	Earnings-price ratio
dfs	Default return spread	Illiquidity	Amihud's illiquidity measure
e/p	Earnings-price ratio	IndIPO	Initial public offerings
infl	Consumer price index	Investment	Investment-to-revenue ratio
ltr	Long-term rate	Leverage	Market leverage
ntis	Net issues	LRreversal	Long-term reversal
svar	Stock variance	Mom12m	12-month momentum
tbl	Treasury bill rate	ShareRepurchase	Share repurchases
tms	Term spread	ShareVol	Share turnover
		zerotrade6M	Zero trading days (6 months)

Table 1. List of predictors used in the analysis.
Validated predictors are those aggregated and applied in forecasting market returns in published studies.

The second component of the dataset includes macroeconomic variables obtained from Welch and Goyal (2008).[8] In particular, the market excess return is calculated as the continuously compounded return on the S&P 500 index, inclusive of dividends, minus the 1-month Treasury bill rate. Additionally, we construct 13 time series predictors based on the definitions provided in this source. A complete list of these time-series predictors, along with validated cross-sectional predictors, is presented in Table 1.

All methods we consider are designed to approximate the predictive model defined in Equation (2). As discussed in Welch and Goyal (2008), while the optimal choice of period for estimating regression models may be unclear, it is crucial to have a sufficient amount of initial data at the beginning of the evaluation period to obtain reliable estimates. In our model training setup, we use a fixed moving window (50 years) for model estimation and next-period prediction. There are multiple splitting schemes to consider, but the trade-off between the size of the training and validation samples is ultimately an empirical question (Bianchi et al., 2021). We

[8]Data is downloaded from Amit Goyal's website (https://sites.google.com/view/agoyal145).

fix the proportion of data used for training and validation to 90% and 10% of the in-sample data, respectively. In consideration of robustness, we predict returns over three different horizons: one, three, and twelve month returns.

4.2 Model performance with validated predictors

We start our analysis with a set of 14 validated cross-sectional predictors shown in Table 1, which have been aggregated to forecast market returns in academic literature. These predictors serve as the baseline for evaluating the performance of various models, including both linear and nonlinear methods we described in Section 2.

Figure 2 presents the comparison of out-of-sample prediction performance for all models, considering both equal-weighted and value-weighted averages of validated predictors. First, by comparing Panel A and Panel B, we find that value-weighted predictors exhibit stronger predictive power, as R^2_{oos} for models using value-weighted predictors are generally higher than that obtained with equal-weighted predictors. Notably, models such as PCR, PLS, and Ridge benefit significantly from value-weighted predictors, showing better predictions than simple historical averages for longer forecast horizons. This improved performance suggests that value-weighted predictors are more effective at capturing systematic market trends, potentially due to their emphasis on larger firms with a more substantial impact on market dynamics.

We focus on the case of value-weighted predictors with a 1-month forecast horizon. Of all the models, OLS performs the worst, yielding an R^2_{oos} of -4.94%. This poor performance is expected because the lack of any regularization makes OLS subject to in-sample overfitting. In addition, OLS is ill-suited to handle high-dimensional input data owing to the presence of multicollinearity. In contrast, regularizing the linear model through dimensionality reduction techniques can significantly improve the prediction of market returns. By extracting principal components, PCR and PLS increase the R^2_{oos} to -0.15% and -0.87%, respectively, bringing their predictive performance close to the level of simple historical averages. Another penalized model that we use in this analysis is Ridge, which applies l_2 regularization to shrink the impact of some variables. It achieves a small but positive R^2_{oos} of 0.08% and demonstrates relatively stable performance across different forecasting horizons.

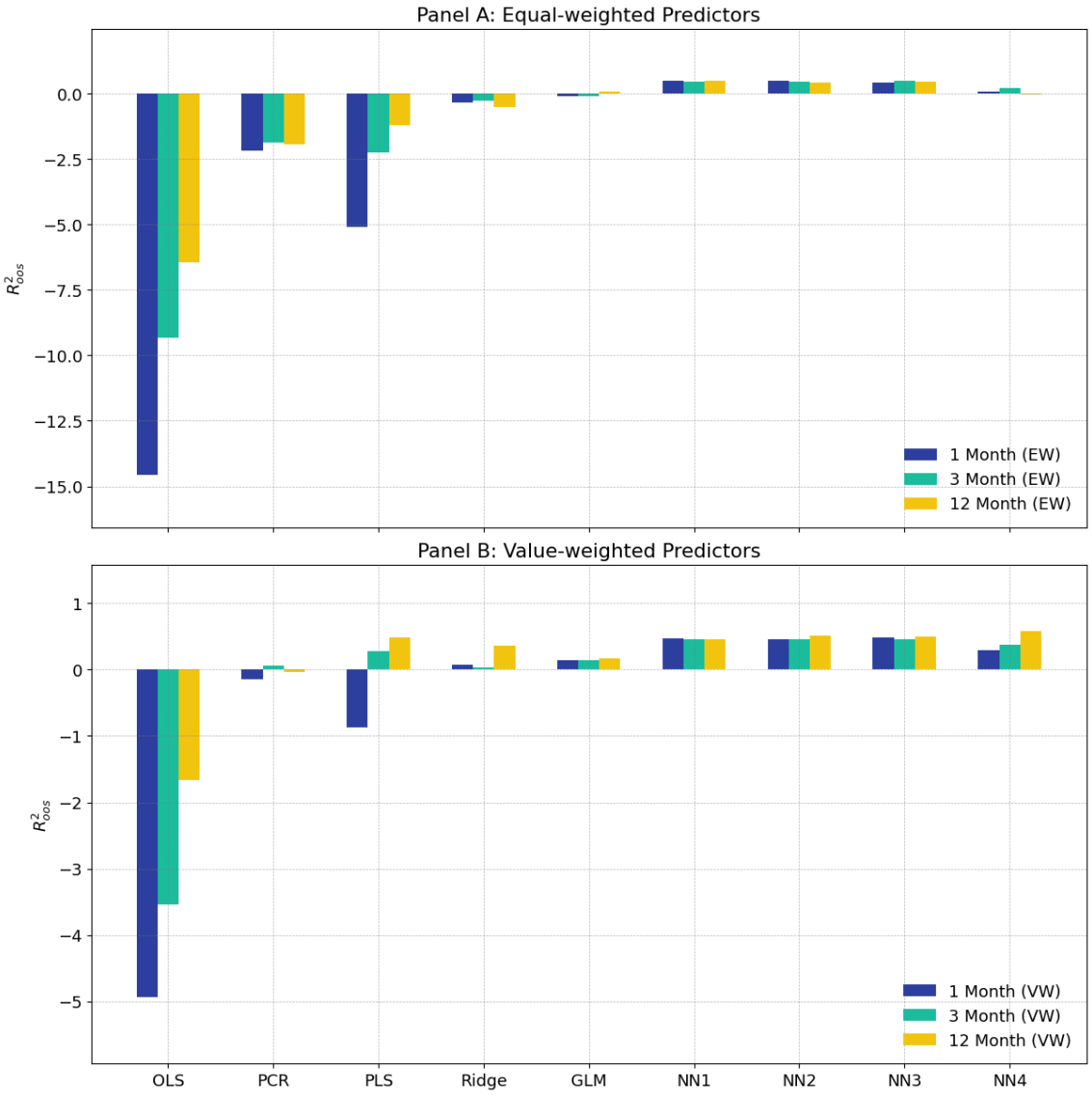

Figure 2. Out-of-sample prediction performance with 14 validated predictors.
Panel A presents the results using equal-weighted average cross-sectional predictors, while Panel B presents the results for value-weighted average cross-sectional predictors. The models evaluated include OLS, PCR, PLS, Ridge, GLM with group lasso, and neural networks (NN1-NN4). The bars represent the percentage out-of-sample R^2, with higher bars indicating better predictive performance.

The generalized linear model with group lasso outperforms other linear models, achieving an out-of-sample R^2_{oos} of 0.14%. The application of spline transformations, which introduce nonlinearities into the model, led to an observable improvement in predictive accuracy. This improvement strongly indicates the presence of nonlinear relationships among the input variables that linear models fail to capture. These results underscore the importance of accounting for nonlinearity in modeling complex data, as it allows the model to better align with the underlying data structure and capture more nuanced patterns.

Neural networks are the best-performing models. The R^2_{oos} values of NN1, NN2, and NN3 are close, at 0.47%, 0.46%, and 0.48%, respectively, much higher than those of other models. However, the single-layer neural network performs slightly worse, achieving an R^2_{oos} of 0.29%. This can be attributed to the fact that the final hidden layer of NN4 contains 32 neurons, far exceeding the 14 input variables, and the training of the single-layer network does not sufficiently learn the hidden nonlinear relationships. These findings indicate that simply increasing the number of neurons in a single layer, without additional layers, is insufficient to model the underlying complexity of the data. Furthermore, the minimal performance differences among the deeper networks suggest that increasing depth beyond a certain point provides limited additional benefit, likely because the dataset's complexity is already adequately captured.

The R^2_{oos} comparison for predicting market returns at 3- and 12-month horizons generally mirrors the results for monthly returns but shows a greater magnitude, except in the cases of ridge regression and neural networks. The predictive performance of neural networks over a longer horizon is nearly unchanged or weakly improved. This stability indicates that deep learning models may capture more than just short-term risk fluctuations, suggesting their potential to account for longer-term return dynamics.

Why do neural networks have better predictive performance? Our conjecture is that the deep learning models capture the nonlinear pattern among the average cross-sectional predictors. Taking NN1 as an example, Figure 3 shows the relationship between the final hidden layer factors and the four different types of predictors (AssetGrowth, EP, Illiquidity, and Mom12m). The existence of strong nonlinearity implies that the neural network is able to learn nonlinear structures in the data.

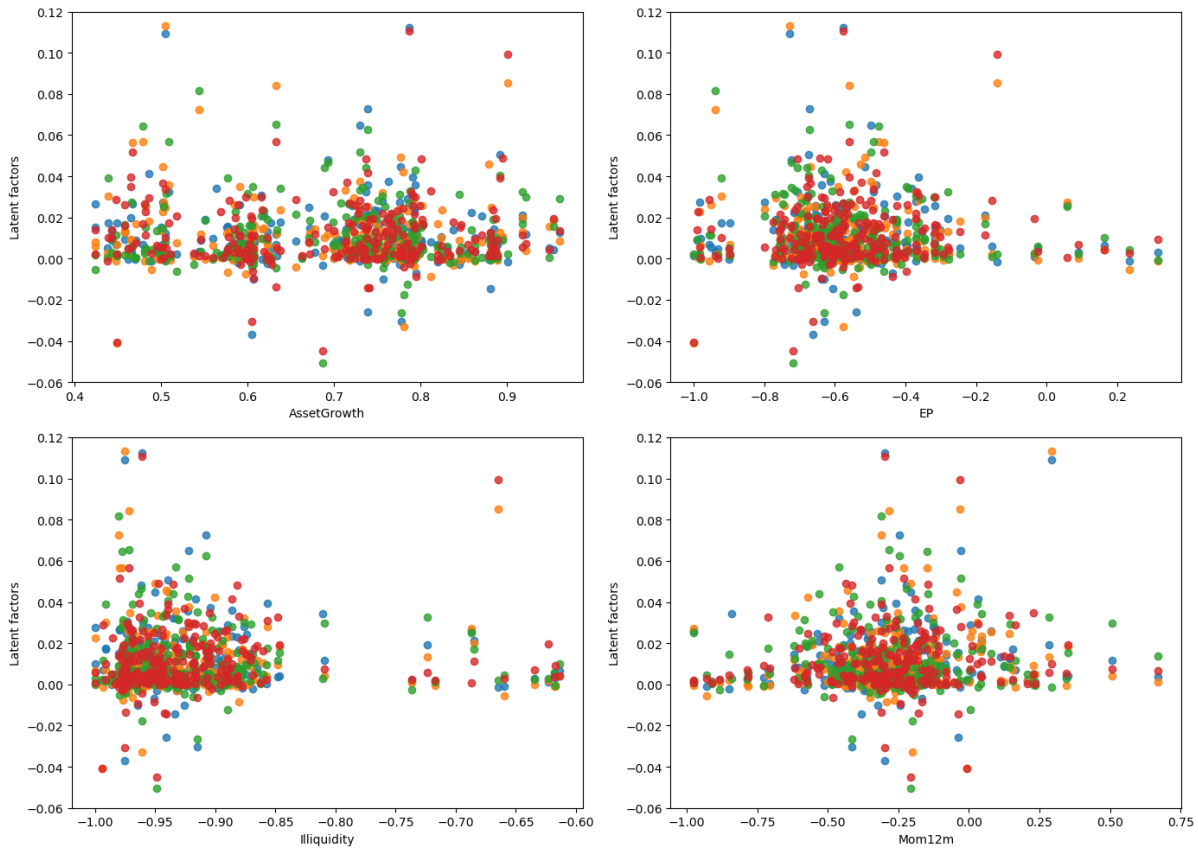

Figure 3. Latent factors against predictive variables.
We generate 4 latent factors by fitting a four layer deep learning model (32-16-8-4). The plot shows the latent factors against different variables, including AssetGrowth, EP, Illiquidity, and Mom12m. Each color indicates one latent factor.

4.3 Model performance with full cross-sectional predictors

Building on the benchmark performance of 14 validated variables, we find that average cross-sectional predictors outperform simple historical averages within a neural network framework. This initial evidence suggests that cross-sectional variables may contain complex systematic information. To further explore this question, we extend our set of predictors to include a total of 105 average cross-sectional features, which allows us to examine whether additional variables provide incremental information.

Figure 4 presents the histogram of R^2_{oos} for all models. Consistent with the benchmark results, the value-weighted predictors continue to outperform the equal-weighted predictors in forecasting the market risk premium. However, the predictive performance of both value-weighted and equal-weighted predictors deteriorates across OLS, PCR, and PLS models when the full set of cross-sectional variables is considered. For instance, in monthly predictions using value-weighted predictors, the R^2_{oos} for PCR and PLS drop to -7.20% and -4.31%, respectively. This decline in performance is mainly attributed to the introduction of noise and redundant information in high-dimensional data, which increases model complexity and the risk of overfitting. Dimension reduction techniques like PCR and PLS face challenges in isolating effective signals when the feature space is expanded significantly.

When using value-weighted predictors, the predictive performance of Ridge, GLM, and neural networks improves relative to the benchmark results. In particular, the Ridge, GLM, and NN4 models exhibit significant gains in R^2_{oos} across nearly all forecast horizons. This improvement likely reflects the ability of Ridge and GLM to handle high-dimensional data through regularization, and NN4's shallow architecture, which effectively leverages the expanded feature set without overfitting. Furthermore, NN4 stands out as the best-performing model, suggesting that shallow neural networks can be highly effective when dealing with a large volume of input features. In comparison, NN1-NN3 models show only a slight improvement in performance, with little to no change in certain cases.

These findings imply that while expanding the set of predictors from 14 to 105 provides some additional information, the incremental gains are limited. The shallow network's ability to extract useful signals with minimal layers further suggests that most of the predictive

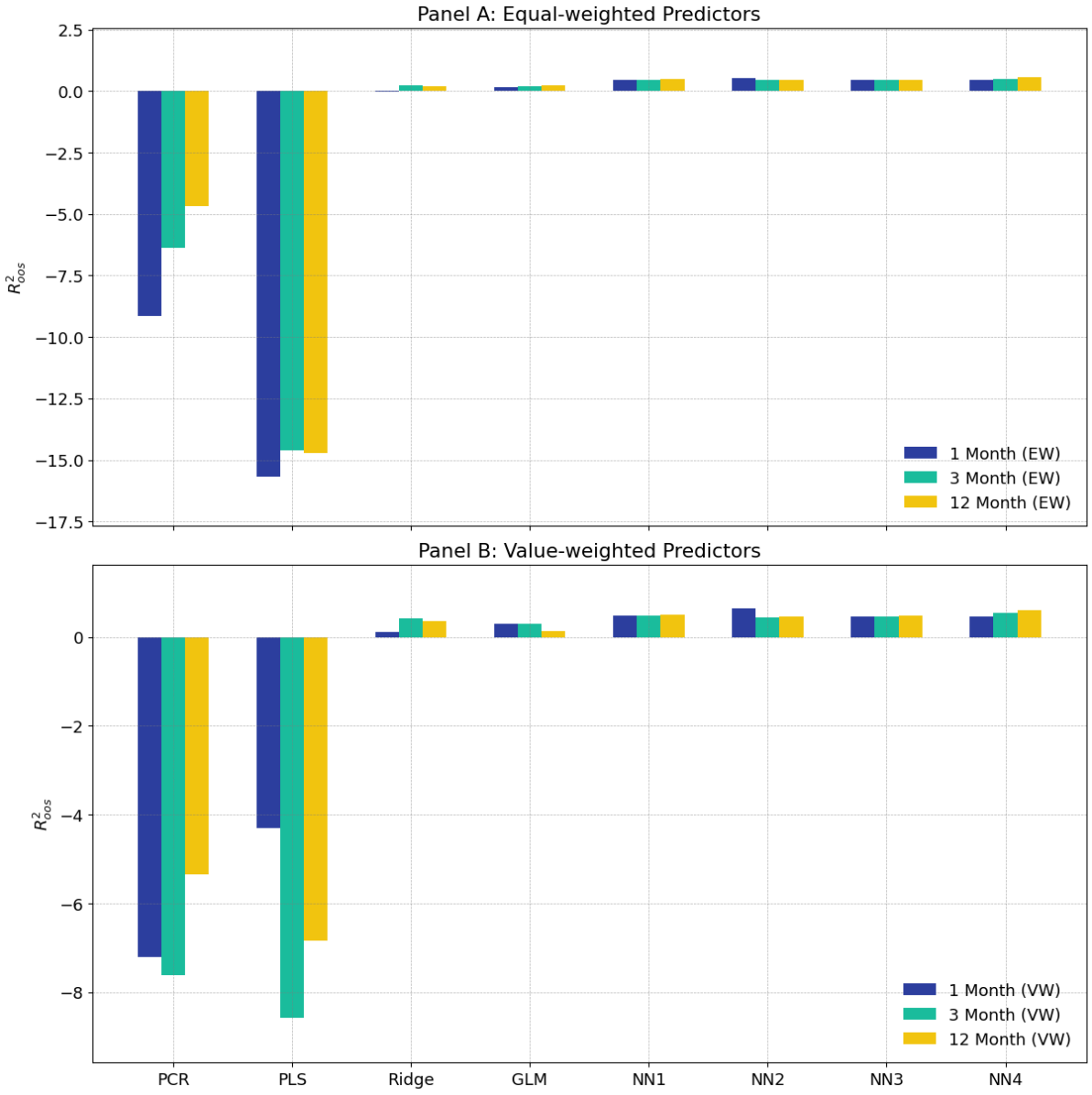

Figure 4. Out-of-sample prediction performance with all predictors.
Panel A presents the results using equal-weighted average cross-sectional predictors, while Panel B presents the results for value-weighted average cross-sectional predictors. The models evaluated include PCR, PLS, Ridge, GLM with group lasso, and neural networks (NN1-NN4). We omit OLS due to its large negative value. The bars represent the percentage out-of-sample R^2, with higher bars indicating better predictive performance.

power is concentrated in a subset of variables, with diminishing returns as more features are added. This highlights the importance of focusing on key variables, as redundant or weakly informative predictions may weaken the model's effectiveness.

	PCR	PLS	Ridge	GLM	NN1	NN2	NN3	NN4
OLS	**3.69**	**3.70**	**4.04**	**4.03**	**4.07**	**4.08**	**4.07**	**4.07**
PCR		0.70	**3.50**	**3.33**	**3.27**	**3.36**	**3.26**	**3.26**
PLS			1.35	1.35	1.27	1.30	1.26	1.27
Ridge				0.48	0.52	0.72	0.49	0.51
GLM					0.32	0.51	0.28	0.30
NN1						0.79	−1.09	−0.43
NN2							−0.91	−0.85
NN3								0.50

Table 2. Pairwise comparison of monthly out-of-sample predictions.
This table presents the Diebold-Mariano test statistics comparing out-of-sample prediction performance of different models using value-weighted predictors. Positive values indicate that the model in the column outperforms the model in the row, and bold values indicate the difference is statistically significant at 5% level or better.

Following Gu et al. (2020), we further implement a pairwise test developed by Diebold and Mariano (1995) to compare the predictive performance of different models. Table 2 reports the test statistics, which confirm our observations from Figure 4. OLS and PCR are the worst-performing models, with their out-of-sample predictions significantly inferior to those of nonlinear models. While the predictive results of neural networks do not differ significantly from those of penalized regressions, the positive test statistics indicate that neural networks still outperform them.

4.4 Model performance with cross-sectional and time-series predictors

While the above results confirm the ability of deep learning models to effectively extract information from cross-sectional variables for predicting market returns, it remains unclear that whether they have included systematic information from time-series macroeconomic variables. To investigate this, we augment the models by including an additional set of 13 time-series predictors.

By comparing Figure 4 and 5, we find that, most models exhibit little to no improvement when combining cross-sectional and time-series variables. In some cases, their performance even declines. This suggests that much of the systematic information contained in time-series variables is already embedded in cross-sectional variables. This result confirms the significant information overlap between these two types of variables, which may stem from the fact that macroeconomic variables affect firm-level characteristics in a consistent manner.

Among all the models tested, only the neural network models (NN1-NN4) demonstrate consistent and robust positive R^2_{oos} across different datasets and forecast horizons. These models remain stable even as the input data changes, highlighting their capacity to effectively extract systematic information from cross-sectional variables while being less sensitive to variations in the dataset compared to traditional linear models.

4.5 Variable importance under deep learning

The expansion of the prediction set does not improve the predictive performance of the model but instead may exacerbate the problem caused by variable redundancy. Therefore, identifying key variables is crucial to effectively forecasting market returns.

We use the SHAP method, as described in Section 3.7, to identify which predictors contribute most significantly to the performance of our deep learning models. For each training sample, the SHAP value quantifies the marginal impact of each feature on the prediction outcome. To summarize variable importance across samples, we compute the average SHAP value for each feature and transform it into a single importance measure.

Figure 6 reports the relative importance of the top 20 value-weighted features across neural networks with varying layer structures, offering a focused view of the most influential predictors.[9] In contrast, Figure 7 provides a comprehensive overview by ranking all features based on their importance. This overall ranking is determined by aggregating the rankings of each feature across different neural networks, where features with higher cumulative importance are placed at the top, and those with lower cumulative importance are positioned at the bottom. The color gradient ranges from light blue to dark blue, where darker shades indicate higher importance values.

[9]The variable importance for equal-weighted averages is provided in Appendix A.

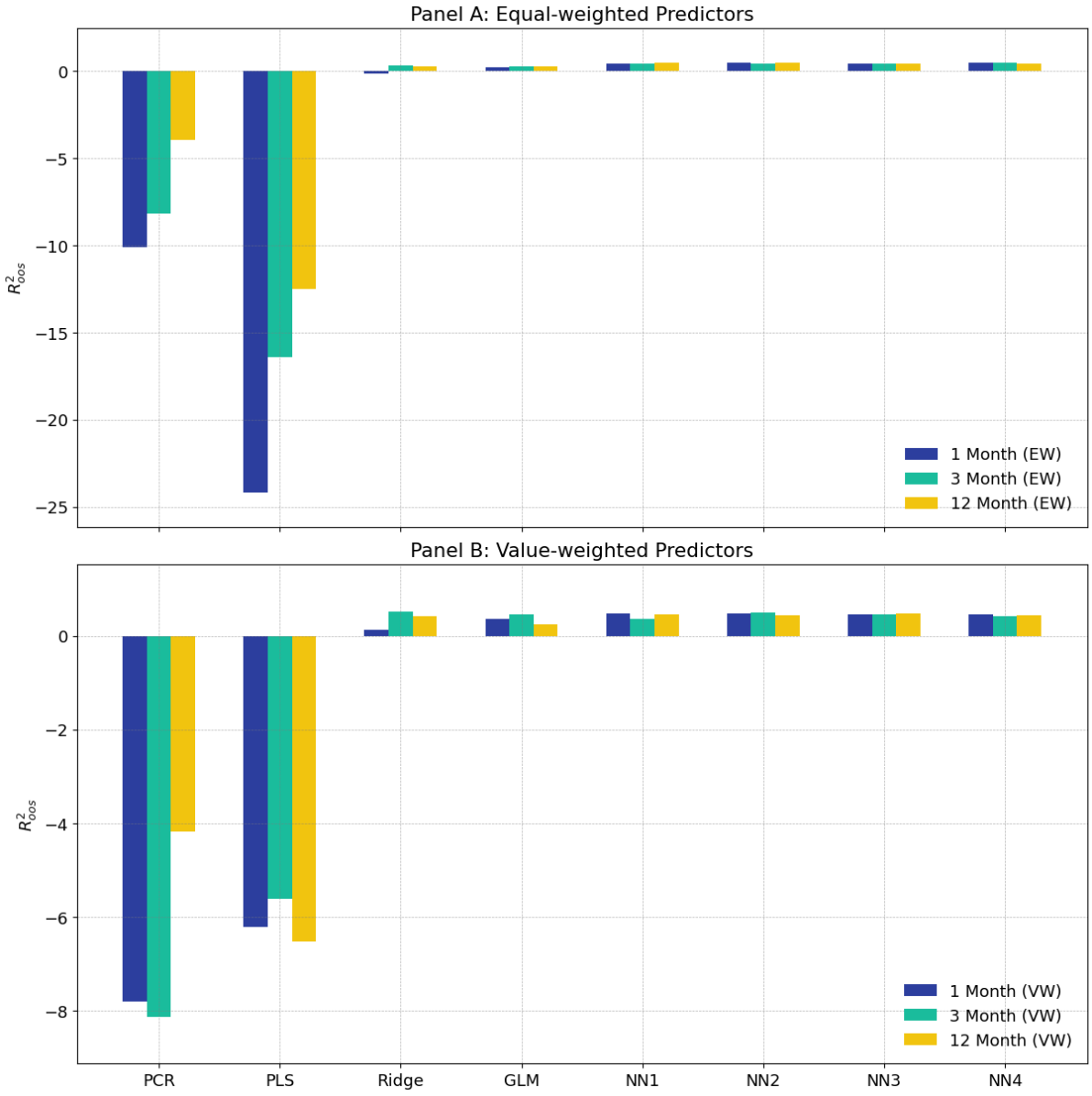

Figure 5. Out-of-sample prediction performance with all predictors.
Panel A presents the results using equal-weighted average cross-sectional predictors, while Panel B presents the results for value-weighted average cross-sectional predictors. The models evaluated include PCR, PLS, Ridge, GLM with group lasso, and neural networks (NN1-NN4). We omit OLS due to its large negative value. The bars represent the percentage out-of-sample R^2, with higher bars indicating better predictive performance.

Comparing the variable importance for NN1-NN4, we find that although the order of the feature importance varies across neural network models, some variables consistently exhibit higher importance.

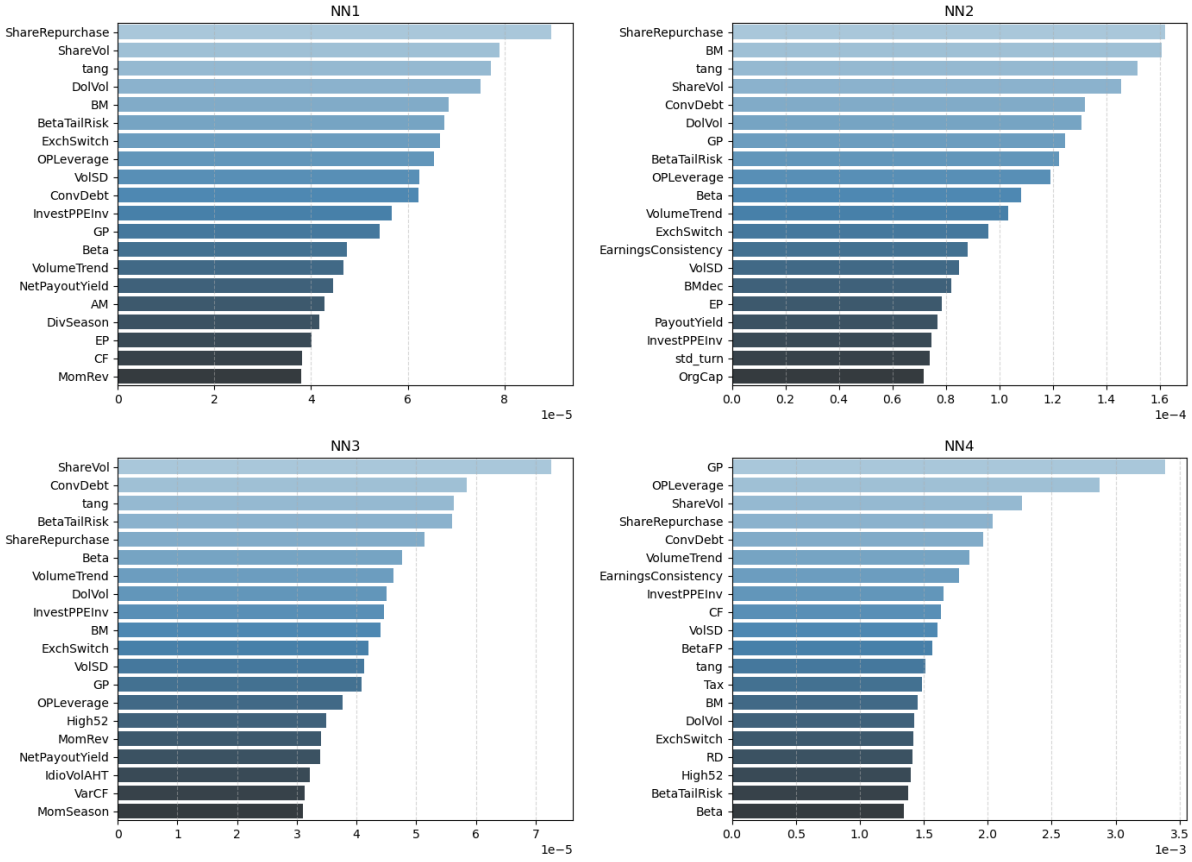

Figure 6. Top 20 most important variables for each model.
This figure shows variable importance for the top-20 most important variables in each neural network, base on the SHAP method.

The first category of variables relates to stock trading activity, including company share repurchase behavior (ShareRepurchase), share volume (ShareVol), past trading volume (DolVol), and trading volume trends (VolumeTrend). In particular, the first two predictors consistently rank among the top five in all neural network models, demonstrating their substantial impact on the prediction of market excess returns. This suggests that trading-related signals provide rich information for forecasting market dynamics, likely due to their ability to capture changes in investor behavior and market liquidity.

The second category contains variables associated with financial leverage and capital structure, such as tangibility (tang), the convertible debt indicator (ConvDebt), and operating lever-



Figure 7. Overall ranking of variables.
Rankings of value-weighted firm-level predictors based on their overall contributions to the model, with darker blue indicating higher importance. Variables with the highest importance are positioned at the top, and those with the lowest at the bottom.

age (OPLeverage). These variables are influential across all four models, with tangibility consistently ranking third in NN1 to NN3. The importance of these variables highlights the ability of firms' capital structure to affect the prediction of market risk premiums.

In addition to the above categories, a few other variables stand out due to their consistent influence across models. These include tail risk beta (BetaTailRisk), which reflects the market sensitivity to extreme downside events; book-to-market ratio (BM), capturing the value premium often associated with undervalued firms; and gross profit to total assets (GP), highlighting firms' operational efficiency. These additional predictors improve the ability of models to forecast market returns by integrating critical aspects of risk exposure, valuation, and profitability, thus contributing to a more comprehensive understanding of the factors driving excess returns.

4.6 Interaction effects

The superior performance of deep learning models supports the benefit of incorporating complex interactions among predictive variables. However, the analysis of interaction effects becomes increasingly challenging given the vast number of potential combinations among a large set of predictors. In our study, we focus on selected pairs of predictors within the NN4 model to identify a handful of potential intersection effects. This allows us to better understand the working mechanism of neural networks. Figure 8 illustrates how the importance of a variable is affected by its interaction with other variables.

More specifically, these four subfigures explore the interaction effects between tangibility and the variables GP, ShareRepurchase, VolumeTrend, and BMdec on the SHAP values. The color gradient represents the level of tangibility, with blue indicating low tangibility and red indicating high tangibility. The differences in SHAP values across varying levels of tangibility suggest that the impact of these characteristics on model predictions is not uniform but rather contingent on the firm's tangibility. For example, in the upper-left subfigure, lower tangibility amplifies the negative contributions of GP, indicating that gross profit to total assets has a more adverse effect in low tangibility scenarios. Similar interactions can be found in the lower-right subfigure with BMdec against tang. In contrast, the lower-left subfigure highlights that lower

tangibility increases the positive impact of volume trends on model predictions, reflecting a stronger momentum effect for firms with lower asset tangibility. Meanwhile, the upper-right subfigure shows that the contribution of ShareRepurchase to the model's predictions becomes increasingly negative as tangibility rises. These observations show that how tangibility acts as a moderator variable, shaping the influence of these predictors on model outcomes.

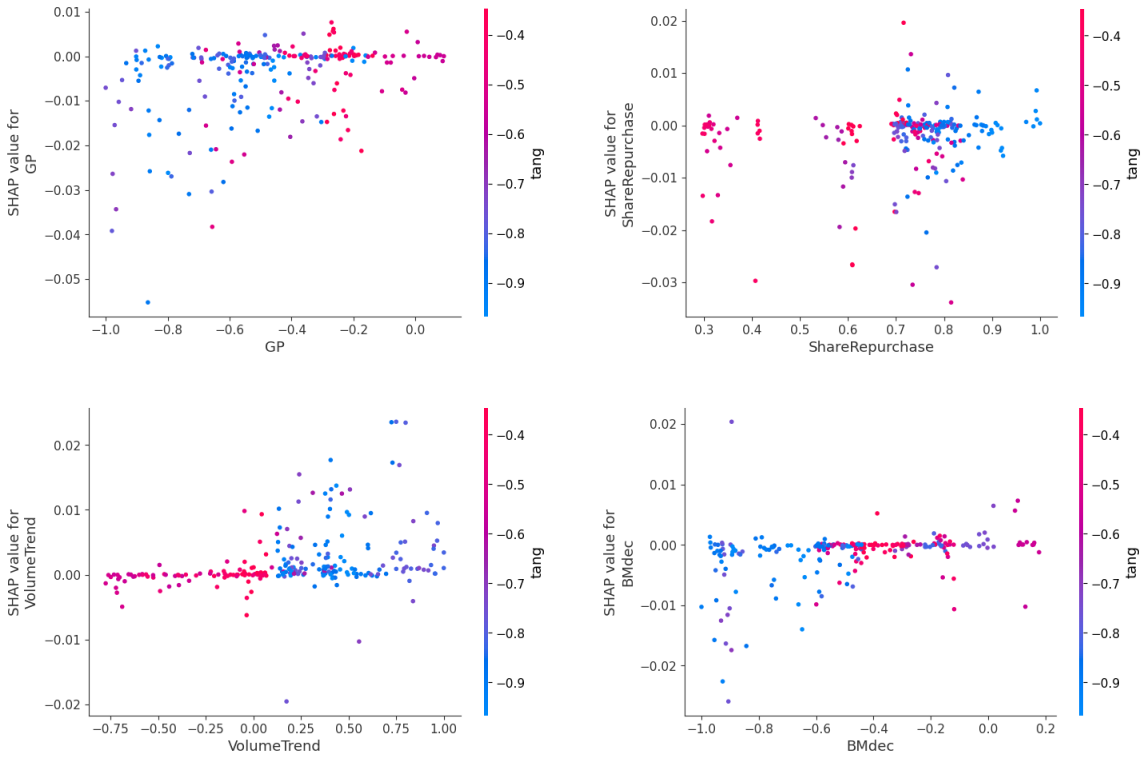

Figure 8. SHAP value and variable intersections.
This figure shows interactions between key variables and tangibility (tang) within NN4 model. The color gradient reflects the effect of tang on the SHAP values for each predictor (GP, ShareRepurchase, VolumeTrend, and BMdec), highlighting variations in their contributions to the model's predictions under different levels of asset tangibility.

4.7 Economic implications

So far, our analysis has focused on the statistical performance of prediction. Next, we assess whether the apparent improvement in forecast accuracy translates into better economic returns relative to the no-predictability scenario. First, we use Equation (21) to convert the out-of-sample R^2 reported in Table A.2 into an improvement in the annualized Sharpe ratio ($SR^* - SR$) relative to a buy-and-hold strategy. Table 3 shows the results. For example, for a mean-variance investor, incorporating the prediction from the NN2 model via a market timing approach can improve the Sharpe ratio of a buy-and-hold strategy from 0.495 to 0.571. In general, when holding market portfolios, deep learning models that exploit information from aggregated cross-sectional predictors can help enhance the Sharpe ratio by more than five percentage points.

	Ridge	**GLM**	**NN1**	**NN2**	**NN3**	**NN4**
Panel A: equal-weighted average cross-sectional predictors						
1 Month	—	0.019	0.055	0.064	0.056	0.056
3 Month	0.031	0.026	0.054	0.055	0.056	0.058
12 Month	0.026	0.028	0.060	0.057	0.057	0.069
Panel B: value-weighted average cross-sectional predictors						
1 Month	0.015	0.037	0.058	0.076	0.056	0.057
3 Month	0.052	0.037	0.058	0.055	0.056	0.065
12 Month	0.044	0.017	0.061	0.057	0.060	0.072

Table 3. Annualized Sharpe ratio improvement.
This table shows the improvement in annualized Sharpe ratio for the 20-year sample of S&P 500, derived from the predictive R^2_{oos} reported in Table A.2. Cases with negative R^2_{oos} are omitted, as they indicate no economic gains from the prediction.

In addition, we adopt the market timing trading strategy proposed by Campbell and Thompson (2008) to assess the economic significance of portfolio predictability. Similarly, we execute the trading strategy based on out-of-sample forecasts, dynamically adjusting positions each month in proportion to the predicted returns. However, unlike their approach, we exclude both leverage and short sales, adhering to a fully-funded and long-only framework. Table 4 reports the annualized Sharpe ratio improvement for this strategy based on various models. The result aligns with models' out-of-sample prediction performance, where trading strategies

based on neural networks outperform other methods. For example, the annualized Sharpe ratio of the NN2-based timing strategy, driven by the equal-weighted cross-sectional predictors, is 9.6% higher than that of a simple buy-and-hold strategy. These results suggest that, although the boost in out-of-sample R^2 is limited, implementing trading strategies based on predictive information still yields economic gains.

	PCR	PLS	Ridge	GLM	NN1	NN2	NN3	NN4
Panel A: equal-weighted average cross-sectional predictors								
1 Month	-0.278	0.023	-0.042	0.019	0.059	0.096	0.064	0.064
3 Month	-0.117	-0.009	0.032	0.024	0.052	0.054	0.058	0.058
12 Month	-0.072	0.003	-0.018	0.009	0.063	0.067	0.063	0.061
Panel B: value-weighted average cross-sectional predictors								
1 Month	-0.281	0.028	0.020	0.045	0.062	0.063	0.064	0.054
3 Month	-0.192	0.084	0.072	0.020	0.070	0.058	0.058	0.046
12 Month	-0.089	-0.106	-0.013	-0.016	0.058	0.063	0.063	0.073

Table 4. Annualized Sharpe ratio improvement of market timing strategies.
This table shows the improvement in the annualized Sharpe ratio for a 20-year sample of S&P 500 achieved by implementing a market timing strategy.

5 Conclusion

A substantial body of literature explores the determinants of equity returns, both from cross-sectional and time-series perspectives. More recently, there has been a growing focus on aggregating firm-level cross-sectional characteristics to predict market returns. Building on these developments, we systematically compare the performance of aggregated cross-sectional predictors in forecasting the market risk premium by using a comprehensive set of both linear and nonlinear models.

Our findings reconfirm the superiority of deep learning models over traditional linear approaches, demonstrating consistent and robust predictive accuracy. Notably, shallow learning networks perform comparably to deep learning networks, though they lack the stability observed in deeper architectures. Moreover, within the framework of the deep learning model, the cross-sectional predictors show better performance compared to simple historical average forecasts, highlighting their capacity to capture systematic information. When incorporating selected time-series macroeconomic variables, we observe negligible improvements in predictive accuracy, suggesting that averaged cross-sectional predictors may share overlapping information with time-series data.

Finally, our analysis reveals that deep learning models identify a handful of dominant predictors. These signals are primarily related to stock trading activity and firms' asset structures, followed by valuation ratios, risk exposures, and profitability measures. These findings emphasize the importance of specific cross-sectional characteristics in forecasting market returns.

Despite these contributions, our study has certain limitations. Limited computational resources restricted our ability to extensively tune hyperparameters and apply ensemble methods for neural networks, which may have constrained the models' performance, especially for neural networks. Additionally, the dataset used in this study includes only variables consistently available throughout the research period, potentially excluding effective predictors that have emerged more recently or are intermittently available. Moreover, while SHAP analysis offers valuable insights into variable importance and partially addresses the interpretability challenges of deep learning models, it remains limited in capturing global patterns and complex variable interactions, leaving room for further exploration.

Future research could explore advanced machine learning techniques, not limited to neural networks, to better capture nonlinear structures and improve predictive performance. Another promising direction is to extend the dataset to include emerging or intermittently available variables, which could enhance the model's adaptability to changing market conditions. Additionally, deeper investigation into variable interactions and their alignment with economic theories could provide a richer understanding of the mechanisms driving their predictive power.

Overall, our study contributes to the empirical understanding of asset pricing by demonstrating the effectiveness of deep learning models and cross-sectional predictors in predicting market risk premiums. By highlighting both their advantages and limitations, we provide a foundation for further exploration of nonlinear models in return forecasting.

References

Amihud, Yakov, "Illiquidity and stock returns: cross-section and time-series effects," *Journal of Financial Markets*, 2002, *5* (1), 31–56.

Baker, Malcolm and Jeffrey Wurgler, "The equity share in new issues and aggregate stock returns," *The Journal of Finance*, 2000, *55* (5), 2219–2257.

Bianchi, Daniele, Matthias Büchner, and Andrea Tamoni, "Bond Risk Premiums with Machine Learning," *The Review of Financial Studies*, February 2021, *34* (2), 1046–1089.

Bossaerts, Peter and Pierre Hillion, "Implementing statistical criteria to select return forecasting models: what do we learn?," *The Review of Financial Studies*, 1999, *12* (2), 405–428.

Butaru, Florentin, Qingqing Chen, Brian Clark, Sanmay Das, Andrew W Lo, and Akhtar Siddique, "Risk and risk management in the credit card industry," *Journal of Banking & Finance*, 2016, *72*, 218–239.

Campbell, John Y, "Stock returns and the term structure," *Journal of Financial Economics*, 1987, *18* (2), 373–399.

Campbell, John Y. and Robert J. Shiller, "The Dividend-Price Ratio and Expectations of Future Dividends and Discount Factors," *The Review of Financial Studies*, 1988, *1* (3), 195–228.

Campbell, John Y and Samuel B Thompson, "Predicting excess stock returns out of sample: Can anything beat the historical average?," *The Review of Financial Studies*, 2008, *21* (4), 1509–1531.

Chen, Andrew Y. and Tom Zimmermann, "Open Source Cross-Sectional Asset Pricing," *Critical Finance Review*, 2022, *27* (2), 207–264.

Chen, Luyang, Markus Pelger, and Jason Zhu, "Deep learning in asset pricing," *Management Science*, 2024, *70* (2), 714–750.

Cochrane, John H et al., "Financial markets and the real economy," *Foundations and Trends® in Finance*, 2005, *1* (1), 1–101.

Cooper, Michael and Huseyin Gulen, "Is time-series-based predictability evident in real time?," *The Journal of Business*, 2006, *79* (3), 1263–1292.

Diebold, Francis X and Roberto S Mariano, "Comparing Predictive Accuracy," *Journal of Business & Economic Statistics*, 1995, *13* (3).

Dow, Charles Henry, *Scientific stock speculation*, Magazine of Wall Street, 1920.

Eldan, Ronen and Ohad Shamir, "The power of depth for feedforward neural networks," in "Conference on learning theory" PMLR 2016, pp. 907–940.

Engelberg, Joseph, R. David McLean, Jeffrey Pontiff, and Matthew C. Ringgenberg, "Do Cross-Sectional Predictors Contain Systematic Information?," *Journal of Financial and Quantitative Analysis*, May 2023, *58* (3), 1172–1201.

Fama, Eugene F and Kenneth R French, "Business conditions and expected returns on stocks and bonds," *Journal of Financial Economics*, 1989, *25* (1), 23–49.

_ **and** _ , "The cross-section of expected stock returns," *The Journal of Finance*, 1992, *47* (2), 427–465.

_ **and** _ , "Multifactor explanations of asset pricing anomalies," *The Journal of Finance*, 1996, *51* (1), 55–84.

_ **and** _ , "Dissecting anomalies," *The Journal of Finance*, 2008, *63* (4), 1653–1678.

Feng, Guanhao, Jingyu He, and Nicholas G. Polson, "Deep Learning for Predicting Asset Returns," *ArXiv*, 2018, *abs/1804.09314*.

Freyberger, Joachim, Andreas Neuhierl, and Michael Weber, "Dissecting characteristics nonparametrically," *The Review of Financial Studies*, 2020, *33* (5), 2326–2377.

Giglio, Stefano and Dacheng Xiu, "Asset pricing with omitted factors," *Journal of Political Economy*, 2021, *129* (7), 1947–1990.

Goyal, Amit, Ivo Welch, and Athanasse Zafirov, "A comprehensive 2022 look at the empirical performance of equity premium prediction," *The Review of Financial Studies*, 2024, *37* (11), 3490–3557.

Gu, Shihao, Bryan Kelly, and Dacheng Xiu, "Empirical Asset Pricing via Machine Learning," *The Review of Financial Studies*, May 2020, *33* (5), 2223–2273.

Guo, Hui and Robert F Whitelaw, "Uncovering the risk–return relation in the stock market," *The Journal of Finance*, 2006, *61* (3), 1433–1463.

Hansen, Lars Kai and Peter Salamon, "Neural network ensembles," *IEEE transactions on pattern analysis and machine intelligence*, 1990, *12* (10), 993–1001.

Heaton, James B, Nick G Polson, and Jan Hendrik Witte, "Deep learning for finance: deep portfolios," *Applied Stochastic Models in Business and Industry*, 2017, *33* (1), 3–12.

Hirshleifer, David, Sonya Seongyeon Lim, and Siew Hong Teoh, "Driven to distraction: Extraneous events and underreaction to earnings news," *The Journal of Finance*, 2009, *64* (5), 2289–2325.

Hodrick, Robert J, "Dividend yields and expected stock returns: Alternative procedures for inference and measurement," *The Review of Financial Studies*, 1992, *5* (3), 357–386.

Hoerl, Arthur E and Robert W Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, 1970, *12* (1), 55–67.

Hutchinson, James M, Andrew W Lo, and Tomaso Poggio, "A nonparametric approach to pricing and hedging derivative securities via learning networks," *The Journal of Finance*, 1994, *49* (3), 851–889.

Ioffe, Sergey and Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *ArXiv*, 2015, *abs/1502.03167*.

Jegadeesh, Narasimhan and Sheridan Titman, "Returns to buying winners and selling losers: Implications for stock market efficiency," *The Journal of Finance*, 1993, *48* (1), 65–91.

Jolliffe, I T, "Principal Component Analysis," *Technometrics*, 2003, *45* (3), 276–276.

Kalatzis, Aquiles E.G., Camila F. Bassetto, and Carlos R. Azzoni, "Multicollinearity and financial constraint in investment decisions: a Bayesian generalized ridge regression," *Journal of applied statistics*, 2011, *38* (2), 287–299.

Karabiyik, Hande and Joakim Westerlund, "Forecasting using cross-section average–augmented time series regressions," *The Econometrics Journal*, 2021, *24* (2), 315–333.

Kelly, Bryan T, Seth Pruitt, and Yinan Su, "Characteristics are covariances: A unified model of risk and return," *Journal of Financial Economics*, 2019, *134* (3), 501–524.

Khandani, Amir E, Adlar J Kim, and Andrew W Lo, "Consumer credit-risk models via machine-learning algorithms," *Journal of Banking & Finance*, 2010, *34* (11), 2767–2787.

Kingma, Diederik P. and Jimmy Ba, "Adam: A Method for Stochastic Optimization," *ArXiv*, 2017, *abs/1412.6980*.

Kothari, Smitu P and Jay Shanken, "Book-to-market, dividend yield, and expected market returns: A time-series analysis," *Journal of Financial Economics*, 1997, *44* (2), 169–203.

Kozak, Serhiy, Stefan Nagel, and Shrihari Santosh, "Shrinking the cross-section," *Journal of Financial Economics*, 2020, *135* (2), 271–292.

Lettau, Martin and Sydney Ludvigson, "Consumption, Aggregate Wealth, and Expected Stock Returns," *The Journal of Finance*, 2001, *56* (3), 815–849.

Lewellen, Jonathan and Jay Shanken, "Estimation Risk, Market Efficiency, and the Predictability of Returns," Working Paper 7699, National Bureau of Economic Research May 2000.

Linnainmaa, Juhani T and Michael R Roberts, "The history of the cross-section of stock returns," *The Review of Financial Studies*, 2018, *31* (7), 2606–2649.

Masters, Timothy, *Practical neural network recipes in C++*, Morgan Kaufmann, 1993.

McLean, R David and Jeffrey Pontiff, "Does academic research destroy stock return predictability?," *The Journal of Finance*, 2016, *71* (1), 5–32.

Moritz, Benjamin and Tom Zimmermann, "Tree-based conditional portfolio sorts: The relation between past and future stock returns," *Available at SSRN 2740751*, 2016.

Moskowitz, Tobias J, Yao Hua Ooi, and Lasse Heje Pedersen, "Time series momentum," *Journal of Financial Economics*, 2012, *104* (2), 228–250.

Polk, Christopher, Samuel Thompson, and Tuomo Vuolteenaho, "Cross-sectional forecasts of the equity premium," *Journal of Financial Economics*, 2006, *81* (1), 101–141.

Rapach, David E, Jack K Strauss, and Guofu Zhou, "International stock return predictability: What is the role of the United States?," *The Journal of Finance*, 2013, *68* (4), 1633–1662.

Sadhwani, Apaar, Kay Giesecke, and Justin Sirignano, "Deep learning for mortgage risk," *Journal of Financial Econometrics*, 2021, *19* (2), 313–368.

Scott, M, Lee Su-In et al., "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, 2017, *30*, 4765–4774.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, 2014, *15* (1), 1929–1958.

Theoharidis, Alexandre Fernandes, Diogo Abry Guillén, and Hedibert Lopes, "Deep learning models for inflation forecasting," *Applied stochastic models in business and industry*, 2023, *39* (3), 447–470.

Welch, Ivo and Amit Goyal, "A Comprehensive Look at The Empirical Performance of Equity Premium Prediction," *The Review of Financial Studies*, July 2008, *21* (4), 1455–1508.

Yao, Jingtao, Yili Li, and Chew Lim Tan, "Option price forecasting using neural networks," *Omega*, 2000, *28* (4), 455–466.

Appendix

A Tables and Figures

	OLS	PCR	PLS	Ridge	GLM	NN1	NN2	NN3	NN4
Panel A: equal-weighted average cross-sectional predictors									
1 Month	−14.57	−2.20	−5.09	−0.35	−0.12	0.47	0.47	0.43	0.06
3 Month	−9.34	−1.86	−2.27	−0.28	−0.09	0.45	0.46	0.47	0.22
12 Month	−6.45	−1.96	−1.23	−0.51	0.05	0.47	0.41	0.46	−0.03
Panel B: value-weighted average cross-sectional predictors									
1 Month	−4.94	−0.15	−0.87	0.08	0.14	0.47	0.46	0.48	0.29
3 Month	−3.54	0.06	0.28	0.03	0.14	0.46	0.46	0.46	0.37
12 Month	−1.67	−0.04	0.49	0.36	0.17	0.46	0.51	0.50	0.58

Table A1. Out-of-sample prediction performance with 14 validated predictors.
This table shows the percentage out-of-sample R^2 of various models, including OLS, PCR, PLS, Ridge, GLM with group lasso, and neural networks (NN1-NN4), over forecast horizons of 1, 3, and 12 months, using 14 validated aggregated cross-sectional predictors.

	PCR	PLS	Ridge	GLM	NN1	NN2	NN3	NN4
Panel A: equal-weighted average cross-sectional predictors								
1 Month	−9.15	−15.69	−0.01	0.15	0.45	0.53	0.46	0.46
3 Month	−6.39	−14.62	0.25	0.21	0.45	0.46	0.46	0.48
12 Month	−4.67	−14.73	0.21	0.23	0.50	0.47	0.47	0.57
Panel B: value-weighted average cross-sectional predictors								
1 Month	−7.20	−4.31	0.12	0.30	0.48	0.64	0.46	0.47
3 Month	−7.62	−8.59	0.43	0.30	0.48	0.45	0.46	0.54
12 Month	−5.35	−6.85	0.36	0.14	0.51	0.47	0.49	0.60

Table A2. Out-of-sample prediction performance with all predictors.
This table shows the percentage out-of-sample R^2 of various models, including PCR, PLS, Ridge, GLM with group lasso, and neural networks (NN1-NN4), over forecast horizons of 1, 3, and 12 months, using all (105) average cross-sectional predictors. We do not report OLS results because of its large negative values.

	PCR	PLS	Ridge	GLM	NN1	NN2	NN3	NN4
Panel A: equal-weighted average cross-sectional predictors								
1 Month	−10.08	−24.16	−0.16	0.22	0.45	0.47	0.44	0.49
3 Month	−8.19	−16.40	0.35	0.27	0.46	0.44	0.46	0.47
12 Month	−3.95	−12.52	0.30	0.27	0.49	0.49	0.46	0.42
Panel B: value-weighted average cross-sectional predictors								
1 Month	−7.80	−6.20	0.12	0.36	0.47	0.47	0.46	0.45
3 Month	−8.12	−5.60	0.52	0.45	0.36	0.50	0.46	0.42
12 Month	−4.18	−6.52	0.41	0.25	0.45	0.44	0.47	0.44

Table A3. Out-of-sample prediction performance with all cross-sectional and time-series predictors.
This table shows the percentage out-of-sample R^2 of various models, including PCR, PLS, Ridge, GLM with group lasso, and neural networks (NN1-NN4), over forecast horizons of 1, 3, and 12 months, using the combined cross-sectional and time-series predictors. We do not report OLS results because of its large negative values.

	PCR	PLS	Ridge	GLM	NN1	NN2	NN3	NN4
OLS	**4.05**	**4.12**	**4.29**	**4.28**	**4.28**	**4.27**	**4.28**	**4.28**
PCR		−0.65	**2.46**	**2.43**	**2.50**	**2.49**	**2.51**	**2.51**
PLS			1.63	1.64	1.65	1.65	1.65	1.65
Ridge				0.68	0.78	0.77	0.80	0.80
GLM					0.47	0.53	0.48	0.49
NN1						0.42	0.69	0.61
NN2							−0.38	−0.37
NN3								0.13

Table A4. Pairwise comparison of monthly out-of-sample predictions.
This table presents the Diebold-Mariano test statistics comparing out-of-sample prediction performance of different models using equal-weighted predictors. Positive values indicate that the model in the column outperforms the model in the row, and bold values indicate the difference is statistically significant at 5% level or better.

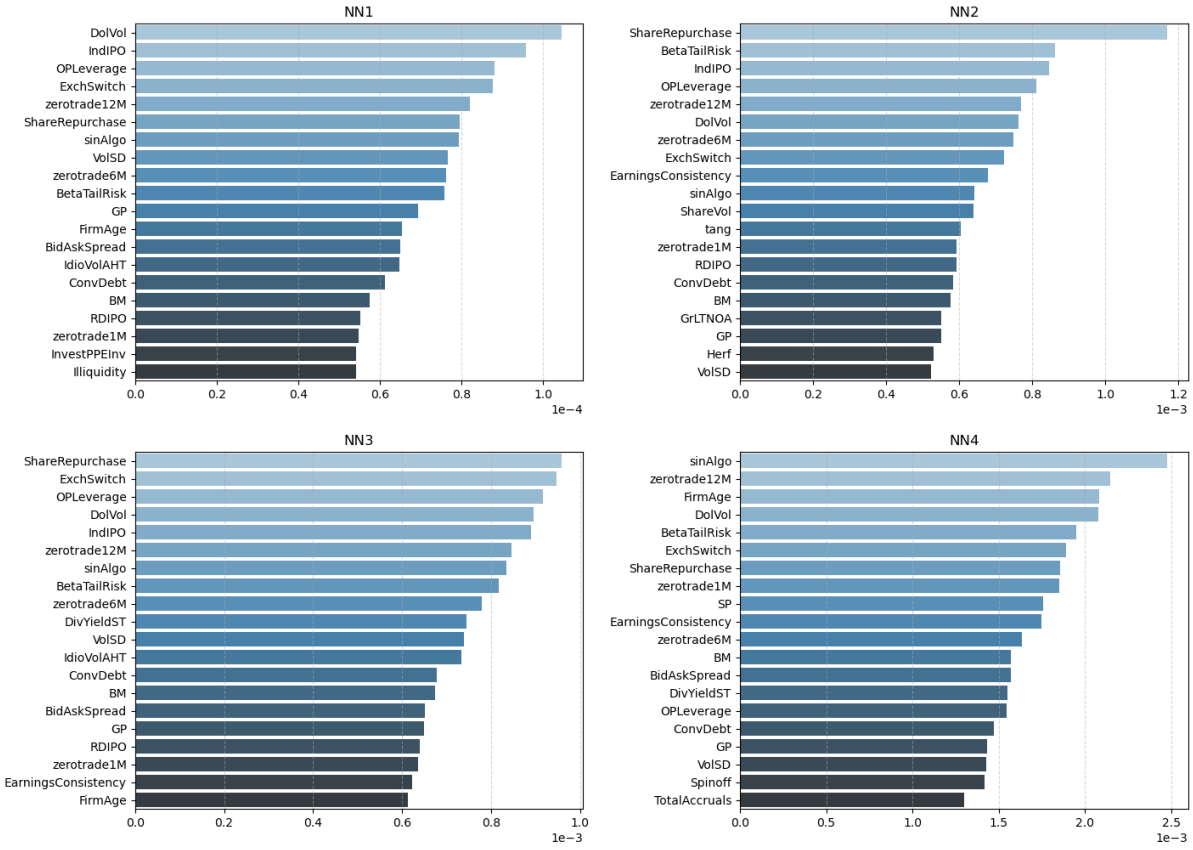

Figure A1. Top 20 most important variables (EW) for each model.
This figure shows variable importance for the top-20 most important variables in each neural network, base on the SHAP method.



Figure A2. Overall ranking of variables (EW).
Rankings of equal-weighted firm-level predictors based on their overall contributions to the model, with darker blue indicating higher importance. Variables with the highest importance are positioned at the top, and those with the lowest at the bottom.

B Details of Firm-level Predictors

No.	Acronym	Long Description	Author	Year, Journal
1	Accruals	Working capital accruals	Sloan	1996, AR
2	AM	Total assets to market	Fama and French	1992, JF
3	AssetGrowth	Asset growth	Cooper et al.	2008, JF
4	Beta	CAPM beta	Fama and MacBeth	1973, JPE
5	BetaFP	Frazzini-Pedersen beta	Frazzini and Pedersen	2014, JFE
6	BetaTailRisk	Tail risk beta	Kelly and Jiang	2014, RFS
7	BidAskSpread	Bid-ask spread	Amihud and Mendelson	1986, JFE
8	BM	Book to market, original	Stattman	1980, Other
9	BMdec	Book to market using December value of market equity	Fama and French	1992, JPM
10	BookLeverage	Book leverage (annual)	Fama and French	1992, JF
11	CashProd	Cash productivity	Chandrashekar and Rao	2009, WP
12	CF	Cash flow to market	Lakonishok et al.	1994, JF
13	ChAssetTurnover	Change in asset turnover	Soliman	2008, AR
14	ChInv	Inventory growth	Thomas and Zhang	2002, RAS
15	ChInvIA	Change in capital inv (ind adj)	Abarbanell and Bushee	1998, AR
16	ChNNCOA	Change in net noncurrent operating assets	Soliman	2008, AR
17	ChNWC	Change in net working capital	Soliman	2008, AR
18	CompEquIss	Composite equity issuance	Daniel and Titman	2006, JF
19	ConvDebt	Convertible debt indicator	Valta	2016, JFQA
20	Coskewness	Coskewness	Harvey and Siddique	2000, JF
21	DelCOA	Change in current operating assets	Richardson et al.	2005, JAE
22	DelCOL	Change in current operating liabilities	Richardson et al.	2005, JAE
23	DelFINL	Change in financial liabilities	Richardson et al.	2005, JAE
24	DelLTI	Change in long-term investment	Richardson et al.	2005, JAE
25	DelNetFin	Change in net financial assets	Richardson et al.	2005, JAE
26	DivInit	Dividend Initiation	Michaely et al.	1995, JF
27	DivOmit	Dividend Omission	Michaely et al.	1995, JF
28	DivSeason	Dividend seasonality	Hartzmark and Salomon	2013, JFE
29	DivYieldST	Predicted dividend yield next month	Litzenberger and Ramaswamy	1979, JFE
30	DolVol	Past trading volume	Brennan, Chordia, Subra	1998, JFE

Table B1. List of firm-level predictors used in our study (Part I).

No.	Acronym	Long Description	Author	Year, Journal
31	EarningsConsistency	Average earnings growth over previous 48 months	Alwathainani	2009, BAR
32	EntMult	Enterprise multiple	Loughran and Wellman	2011, JFQA
33	EP	Earnings-to-price ratio	Basu	1977, JF
34	ExchSwitch	Exchange switch	Dharan and Ikenberry	1995, JF
35	FirmAge	Firm age based on CRSP	Zhang	2006, JF
36	GP	Gross profits / total assets	Novy-Marx	2013, JFE
37	GrLTNOA	Growth in long term operating assets	Fairfield et al.	2003, AR
38	GrSaleToGrInv	Sales growth over inventory growth	Abarbanell and Bushee	1998, AR
39	GrSaleToGrOverhead	Sales growth over overhead growth	Abarbanell and Bushee	1998, AR
40	Grcapx3y	Investment growth (three years)	Anderson and Garcia-Feijoo	2006, JF
41	Grcapx	Change in capex (two years)	Anderson and Garcia-Feijoo	2006, JF
42	Herf	Industry concentration (sales)	Hou and Robinson	2006, JF
43	HerfAsset	Industry concentration (assets)	Hou and Robinson	2006, JF
44	HerfBE	Industry concentration (equity)	Hou and Robinson	2006, JF
45	High52	52 week high	George and Hwang	2004, JF
46	Idiovol3f	Idiosyncratic risk (FF3 model)	Ang et al.	2006, JF
47	IdiovolAHT	Idiosyncratic risk (AHT)	Ali et al.	2003, JFE
48	Illiquidity	Amihud's illiquidity	Amihud	2002, JFM
49	IndIPO	Initial public offerings	Ritter	1991, JF
50	IndMom	Industry momentum	Grinblatt and Moskowitz	1999, JF
51	IndRetBig	Industry return of big firms	Hou	2007, RFS
52	IntMom	Intermediate momentum	Novy-Marx	2012, JFE
53	InvGrowth	Inventory growth	Belo and Lin	2012, RFS
54	InvestPPEInv	Change in PPE and inv/assets	Lyandres et al.	2008, RFS
55	Investment	Investment to revenue	Titman et al.	2004, JFQA
56	LRreversal	Long-run reversal	De Bondt and Thaler	1985, JF
57	Leverage	Market leverage	Bhandari	1988, JF
58	MRreversal	Medium-run reversal	De Bondt and Thaler	1985, JF
59	MaxRet	Maximum return over month	Bali et al.	2011, JFE
60	Mom12m	Momentum (12 month)	Jegadeesh and Titman	1993, JF
61	Mom12mOffSeason	Momentum (12 month) without the seasonal part	Heston and Sadka	2008, JFE
62	Mom6m	Momentum (6 month)	Jegadeesh and Titman	1993, JF

Table B2. List of firm-level predictors used in our study (Part II).

No.	Acronym	Long Description	Author	Year, Journal
63	MomOffSeason	Off season reversal yrs 2-5	Heston and Sadka	2008, JFE
64	MomOffSeason06YrPlus	Off season reversal yrs 6-10	Heston and Sadka	2008, JFE
65	MomOffSeason11YrPlus	Off season reversal yrs 11-15	Heston and Sadka	2008, JFE
66	MomOffSeason16YrPlus	Off season reversal yrs 16-20	Heston and Sadka	2008, JFE
67	MomRev	Momentum and LT reversal	Chan and Ko	2006, JOIM
68	MomSeason	Return seasonality yrs 2-5	Heston and Sadka	2008, JFE
69	MomSeason06YrPlus	Return seasonality yrs 6-10	Heston and Sadka	2008, JFE
70	MomSeason11YrPlus	Return seasonality yrs 11-15	Heston and Sadka	2008, JFE
71	MomSeason16YrPlus	Return seasonality yrs 16-20	Heston and Sadka	2008, JFE
72	MomSeasonShort	Return seasonality last year	Heston and Sadka	2008, JFE
73	MomVol	Mom in high volume stocks	Lee and Swaminathan	2000, JF
74	NetPayoutYield	Net payout yield	Boudoukh et al.	2007, JF
75	OPLeverage	Operating leverage	Novy-Marx	2011, ROF
76	OrgCap	Organizational capital	Eisfeldt and Papanikolaou	2013, JF
77	PayoutYield	Payout yield	Boudoukh et al.	2007, JF
78	RD	R&D over market cap	Chan et al.	2001, JF
79	RDIPO	IPO and no R&D spending	Gou, Lev and Shi	2006, JBFA
80	RIO_Turnover	Inst own and turnover	Nagel	2005, JFE
81	RIO_Volatility	Inst own and idio vol	Nagel	2005, JFE
82	RealizedVol	Realized (total) volatility	Ang et al.	2006, JF
83	ResidualMomentum	Momentum based on FF3 residuals	Blitz et al.	2011, JEmpFin
84	ReturnSkew	Return skewness	Bali et al.	2015, Book
85	ReturnSkew3F	Idiosyncratic skewness (FF3 model)	Bali et al.	2015, Book
86	SP	Sales-to-price	Barbee et al.	1996, FSJ
87	ShareIss1Y	Share issuance (1 year)	Pontiff and Woodgate	2008, JF
88	ShareIss5Y	Share issuance (5 year)	Daniel and Titman	2006, JF
89	ShareRepurchase	Share repurchases	Ikenberry et al.	1995, JFE
90	ShareVol	Share volume	Datar et al.	1998, JFM
91	SinAlgo	Sin stock (selection criteria)	Hong and Kacperczyk	2009, JFE
92	Spinoff	Spinoffs	Cusatis et al.	1993, JFE
93	Std_turn	Share turnover volatility	Chordia et al.	2001, JFE
94	SurpriseRD	Unexpected R&D increase	Eberhart et al.	2004, JF
95	Tang	Tangibility	Hahn and Lee	2009, JF

Table B3. List of firm-level predictors used in our study (Part III).

No.	Acronym	Long Description	Author	Year, Journal
96	Tax	Taxable income to income	Lev and Nissim	2004, JF
97	TotalAccruals	Total accruals	Richardson et al.	2005, JAE
98	TrendFactor	Trend factor	Han, Zhou, Zhu	2016, JFE
99	VarCF	Cash-flow to price variance	Haugen and Baker	1996, JFE
100	VolMkt	Volume to market equity	Haugen and Baker	1996, JFE
101	VolSD	Volume variance	Chordia et al.	2001, JFE
102	VolumeTrend	Volume trend	Haugen and Baker	1996, JFE
103	Zerotrade12M	Days with zero trades (12 months)	Liu	2006, JFE
104	Zerotrade1M	Days with zero trade (1 month)	Liu	2006, JFE
105	Zerotrade6M	Days with zero trades (6 months)	Liu	2006, JFE

Table B4. List of firm-level predictors used in our study (Part IV).

C Algorithms and hyperparameters

In this section, we detail the hyperparameters and algorithms used to train different models. For neural networks, we applied the adaptive moment estimation algorithm (Adam), a widely used variant of stochastic gradient descent (SGD) introduced by Kingma and Ba (2017). Adam calculates adaptive learning rates for individual parameters by combining estimates of the first and second moments of gradients, making it highly efficient for training deep learning models. Algorithm 1 outlines this procedure.

Algorithm 1: Adam for stochastic optimization

Input: Learning rate α, decay rates $\beta_1, \beta_2 \in [0, 1]$, small constant ϵ (e.g., 10^{-8}).

Initialize: Parameter vector θ_0, moment vectors $m_0 = 0$, $v_0 = 0$, and $t = 0$.

while θ_t not converged **do**

$t \leftarrow t + 1$;

$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Compute gradient w.r.t stochastic objective)

$m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$ (Update biased first moment estimate)

$v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ (Update biased second moment estimate)

$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)

$\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second moment estimate)

$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)

Result: The final parameter estimate θ_t.

Batch normalization reduces the variability of predictors by adjusting and scaling the activations, which increases the stability and training speed of the neural network. To prevent the covariates from shifting in the training process, we adopt batch normalization (Ioffe and Szegedy, 2015), as outlined in Algorithm 2, before the activation function at each layer.

Algorithm 2: Batch normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_1, x_2, \ldots, x_m\}$; parameters γ, β.

$\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i$ (Mini-batch mean)

$\sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_B)^2$ (Mini-batch variance)

$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$ (Normalize input)

$y_i \leftarrow \gamma \hat{x}_i + \beta$ (Scale and shift)

Result: $y_i = \mathrm{BN}_{\gamma,\beta}(x_i)$ for all i.

Algorithm 3 is the dropout algorithm. We use dropout (Srivastava et al., 2014) to avoid overfitting, as it randomly removes units from the neural network during training to prevent excessive unit co-adaptation.

Algorithm 3: Dropout

Input: Input x, weight matrix W, bias b, activation function f, dropout probability p.

$r_i \sim \text{Bernoulli}(1-p)$ (Generate a random mask r)

$h_i = f(W \cdot (x \odot r) + b)$ (Compute activations)

$W' = (1-p) \cdot W$ (Scale weights)

$h = f(W' \cdot x + b)$ (Compute activations)

Result: The activation after dropout h.

Early stopping allows us to monitor the performance of the model on the validation set (e.g., validation loss) during the training process. When the validation loss no longer improves after a number of epochs, training is automatically stopped early, and the model with the best validation performance is retrieved. Algorithm 3 gives the procedure of early stopping.

Algorithm 4: Early stopping

Input: Patience parameter p, initial parameters θ, training and validation datasets.

Initialize: $j = 0$, $\epsilon = \infty$. **while** $j < p$ **do**

    Update θ using Algorithm 2 ;

    Calculate the prediction error from the validation sample, denoted as ϵ';

    if $\epsilon' < \epsilon$ **then**

        $j \leftarrow 0$;

        $\epsilon \leftarrow \epsilon'$;

        $\theta' \leftarrow \theta$;

    else

        $j \leftarrow j + 1$;

Result: The final parameter estimate θ'.

Finally, it is important to emphasize that we use a form of ensembling, where we train multiple copies of the network with different seeds for randomly drawn initial network weights. Using fixed seeds typically produces replicable results. However, different seeds produce different predictions. Therefore, to reduce the prediction variance, we average the predictions of networks initialized with different seeds. Specifically, we initialize and train five models with different but fixed seeds and take the average of their predictions as the final result. Due to computational resource limitations, our results can only represent a lower bound for deep learning models.

Models	**Parameters**
PCR	$\rho = 0.8$
PLS	$P \in [1, 10]$, CV=5
Ridge	$\alpha \in [100, 10000]$, CV=5
GLM	$\lambda = 0.1$, $\delta = 0.1$, n_knots = 3
NN1–NN4	L2 penalty $\lambda = 0.01$, batch size = 6, epochs = 150, patience = 20, ensemble = 5
	Adam Para. = default

Table C1. Hyperparameter settings for models.
This table shows the hyperparameters that we tune in each model.

D AI disclosure

During the preparation of this thesis, the only AI tool we used was ChatGPT. The tool was used for the following two purposes:

1. **Code review and refinement:** We used ChatGPT to review and refine code implementations. The feedback provided by the tool allowed us to identify potential issues and improve the code's accuracy and functionality, ensuring that it satisfied the intended research objectives.

2. **Text revision and improvement:** We used ChatGPT to refine the thesis text by suggesting improvements to its clarity and conciseness. These suggestions were carefully considered and incorporated, resulting in a more polished and professional presentation of the research findings.

In all cases, we critically evaluated the suggestions and outputs from ChatGPT to ensure their appropriateness and relevance to the research. By maintaining full oversight and control over the integration of AI-generated content, we ensured that the thesis adheres to rigorous academic standards.