Dissecting Momentum: We Need to Go Deeper

Dmitry Borisenko*

First draft: July 2019
This version: August 2019

Abstract

Cross-sectional predictability of returns by past prices, or momentum, is a lasting market anomaly. Previous research reports numerous ways to measure momentum and establishes a multitude of factors predicting its performance. The emerging machine learning asset pricing literature further identifies price-based firm characteristics as major predictors of returns. I investigate predictive power of a broad set of price-based variables over various time horizons in a deep learning framework and document rich non-linear structure in impact of these variables on expected returns in the US equity market. The magnitude and sign of the impact exhibit substantial time variation and are modulated by interaction effects among the variables. The degree of non-linearity in expected returns varies over time and is highest in distressed markets. Incorporating insights from the literature on time-varying, market state-dependent momentum risks and momentum crashes helps to improve out-of-sample performance of neural network portfolios, especially with respect to the downside risk – investment strategies built on predictions of the deep learning model actively exploit the non-linearities and interaction effects, generating high and statistically significant returns with a robust risk profile and their performance virtually uncorrelated with the established risk factors including momentum. Lastly, I make a case for adoption of automated hyperparameter optimization techniques as an important component of disciplined research in financial machine learning.

Keywords: Momentum, Return Predictability, Time-Varying Risk, Machine Learning, Deep Learning, Neural Networks, Bayesian Optimization

JEL Classification: G11, G12, G17, C45

E-mail: dmitrijborisenko@gmail.com

A Note to the Reader

I have a couple of issues on my mind:

- Wording. I use classification instead of regression and the phrases like 'predicted probability of return being above the cross-sectional median increases in X' does not roll off the tongue, to say the least, though it means that 'the expected return in X'. Perhaps I should define something like *'expected return*'* to mean the expected return in terms of the predicted probabilities and use it throughout the text to make the wording less cumbersome.
- The analysis of the aggregate magnitude of non-linearities in Section IV.D seems rather trivial to me, though it emphasizes that these non-linearities do matter, especially in distressed markets.

I would appreciate any thoughts, comments and feedback.

I. Introduction

Cross-sectional predictability of returns by past price data, or momentum, has been a lasting asset pricing anomaly for over twenty five years. Numerous ways to measure the momentum, e.g. different lookback horizons, and various factors explaining or predicting its risk and return have been proposed in the literature. The emerging field of machine learning asset pricing further identifies the price-based firm characteristics as the major drivers of returns which dominate fundamental variables such as accounting ratios.

In this paper I investigate predictive power of a broad set of price-based variables, or *features*, over various time horizons in a deep learning framework. I document rich non-linear structure in impact of these variables on expected returns in the US equity market. The magnitude and sign of the impact exhibit substantial time variation and are modulated by interaction effects among the features. The degree of non-linearity in expected returns also varies substantially over time and is at its highest in distressed markets. I show that investment strategies built on the out-of-sample predictions of the deep learning model actively exploit the non-linearities and interaction effects, generating high and statistically significant returns with a robust risk profile and their performance virtually uncorrelated with the established risk factors, including momentum, and machine learning portfolios from the current literature. I further make two methodological contributions to the emerging field of financial machine learning:

First, I leverage differentiability of neural networks' outputs with respect to input variables to study directional effects of features on models' predictions, their evolution over time and interactions with other variables. This analysis allows to explicitly relate the predictions to stylized facts about momentum, thus increasing transparency of the results and showcasing interpretability of the infamously black box algorithm. This approach also allows to take a look at existing strategies through the lens of a deep learning model – my model can identify the well known risks of the standard 12-1 momentum strategy such as negative exposure to volatility and optionality in returns during protracted bear markets.

Second, deep learning models are very sensitive to choice of hyperparameters that determine model architecture and guide estimation process. The machine learning asset pricing literature glosses over the important topic of hyperparameter optimization, relying either on manual or random search, while the computer science literature emphasizes that performance of machine learning models depends more on hyperparameters than, for instance on how sophisticated a particular model is. I fill this gap

by applying the tree-structured Parzen estimator (TPE) of Bergstra, Bardenet, Bengio, and Kégl (2011), a Bayesian optimization technique, to systematically search for best architectures in an unsupervised manner. I thus advocate using such techniques to increase transparency and reproducibility of financial machine learning research.

This paper is complementary to two strands of the literature: momentum and financial machine learning. The former investigates cross-sectional predictability of returns from past performance – a long standing asset pricing anomaly first documented by Jegadeesh and Titman (1993) and Asness (1995).[1] Over the past two and a half decades the literature has proposed numerous ways to measure the momentum and identified a host of factors explaining return on momentum strategies and predicting their performance. I investigate a large subset[2] of these factors jointly for different horizons and show that a deep learning model can capture many stylized facts about momentum in a flexible way and provide some fresh insights into the anomaly. For instance, I find that although 12-month raw and risk-adjusted returns are among the strongest unconditional cross-sectional predictors of the future outperformance of a stock relative to cross-section, sensitivity of expected returns to these variables fades away and becomes negative at horizons of eight to nine months, which is especially prominent for the former – even sooner than previously documented by Novy-Marx (2012). More importantly, the sensitivity exhibits substantial time-variation: when market volatility is high and market returns are low, the sensitivity of expected returns to long (short) horizon past returns decreases (increases) and the net impact of long versus short horizons becomes negative. Thus my model can account for the impact of market states on momentum payoffs, which relates this paper to studies examining predictability of momentum returns by market return and volatility (Cooper, Gutierrez Jr, and Hameed (2004)) and momentum crashes (Barroso and Santa-Clara (2015), Daniel and Moskowitz (2016)). The model also captures optionality in momentum payoffs during protracted bear markets documented by Daniel and Moskowitz (2016), who demonstrate that in depressed markets the standard 12-1 momentum strategy accumulates substantial negative and asymmetric market beta which leads to a crash when the market rapidly rebounds. However, in contrast to Daniel and Moskowitz (2016) who build a crash-resilient momentum strategy by forecasting conditional mean and variance of the long-short momentum portfolio using long-term bear market indicator and market volatility in the mean equation, my model puts an additional emphasis on the most recent market performance and thus exploits the optionality in a more direct

[1] Persistence of the anomaly has been documented across different asset classes and countries (Asness, Moskowitz, and Pedersen (2013), Menkhoff, Sarno, Schmeling, and Schrimpf (2012)) and time periods (Israel and Moskowitz (2013), Geczy and Samonov (2016), Goetzmann and Huang (2018)). This list is far from being exhaustive – I refer the reader to the recent studies of Daniel and Moskowitz (2016) and Barroso and Santa-Clara (2015) for an overview of the momentum literature.

[2] I discuss these variables in detail and review the corresponding literature in Section II.

fashion.

The financial machine learning literature further unanimously identifies the price-based characteristics to be among the strongest predictors of expected returns (Gu, Kelly, and Xiu (2018), Messmer (2017), Chen, Pelger, and Zhu (2019), Kelly, Pruitt, and Su (2017), Feng, Giglio, and Xiu (2017)).[3] Gu et al. (2018) compare a wide array of machine learning methods and report neural networks to be among the top performing models. I make three contributions to this field:

First, returns on neural network portfolios reported in the literature exhibit pathological behavior eerily similar to that of the standard momentum, especially with value-weighting. Messmer (2017) reports negative performance during high volatility regimes, Gu et al. (2018) report maximum drawdown going from 14.8 to 54.7 for their best performing neural network, and procyclicality in Sharpe ratios.[4] It is hardly surprising that non-linear models that use the same inputs as the standard momentum would step on the same rake. I show that incorporating insights from the literature on time-varying, market state-dependent momentum risks and momentum crashes helps to dramatically improve out-of-sample performance of neural network portfolios, especially with respect to the downside risk.

Second, value-weighting also reduces the performance of the machine learning portfolios by approximately one half in terms of Sharpe ratios, but does not eliminate exposure to the size factor.[5] In a large scale replication study Hou et al. (2017) demonstrate that the vast majority of the documented asset pricing anomalies disappear once small and microcaps are excluded from analysis, although momentum fares relatively well. I take this evidence seriously and demonstrate that my model is capable of producing portfolios with robust performance even within the largest US equities: a long-short portfolio effectively holding one half of the S&P 500 stocks and shorting another half has a Fama and French (2015) plus momentum alpha of 7.5 percent p.a. (t-statistic 6.7), no significant loading on any of the factors, and a Sharpe ratio of 1.1.

[3] Takeuchi and Lee (2013) show that price features alone are sufficient to generate strategies with remarkable out-of-sample performance, although they do not subject their results to a formal asset pricing analysis.

[4] The maximum drawdown of the long-short value-weighted portfolio of Chen et al. (2019) increases from 16 to 36.6 percent, with the second worst drawdown of over 30 percent peaking in early 2009. Although their strategy avoids the momentum crash of 2009, it underperforms in the highly volatile environment of 2007-2008, which is not surprising given its positive and significant loadings on size and value factors. For these calculations I use the data from Markus Pelger's web page (`https://mpelger.people.stanford.edu/research`).

[5] In fact, the test set spread in returns between the ninth and tenth value-weighted decile portfolios of Chen et al. (2019) is over 9 percent p.a. In a Fama and French (2015) plus momentum regression these portfolios have size loadings of 0.39 and 0.83 respectively with t-statistics of over 5 each. The long-short portfolio has a size beta of 0.72 (t-statistic 3.97) and the rest of the portfolios, in general, have small and statistically insignificant exposures to the size factor.

Third, the previous studies rely either on manual tuning or random search to choose model hyperparameters. Bergstra, Yamins, and Cox (2013) argue that hyperparameter tuning should be formal, quantified and reproducible part of model evaluation. Bergstra et al. (2011) introduce the TPE algorithm – a sequential Bayesian optimization technique allowing to formalize the hyperparameter tuning task as an outer optimization problem. They demonstrate that TPE outperforms both the manual and random search. I show that the algorithm consistently proposes hyperparameter configurations resulting in lower loss in the dataset standard for empirical asset pricing. Gu et al. (2018) report that relatively shallow networks outperform deeper models. I find that both shallow and deep models (up to five hidden layers) achieve similar performance, but optimal hyperparameters for shallower and deeper architectures are rather distinct.[6] The key point is that techniques like TPE allow to search for optimal architectures in a systematic way with minimal interference from the researcher. I thus advocate the use of automated hyperparameter optimization as an important component of disciplined research in financial machine learning.

The layout of the paper is as follows: Section II describes the data, construction of input variables and reviews the corresponding literature. Section III describes the model and estimation with a special emphasis on hyperparameter optimization. Section III also describes my approach to building an optimal ensemble of models and outlines several metrics I construct to measure degree of non-linearity of the model's outputs with respect to the input variables. Section IV presents results of the paper: it analyzes out-of-sample performance of the model and examines returns on the portfolios based on the model's predictions; it further investigates which variables are the most important predictors of expected returns and how non-linearity in the predictions varies over time; then it demonstrates how the model can be used to analyze existing strategies by revisiting the momentum crash of 2009 and closes with robustness checks. Section V concludes.

[6]Given that neural networks are low-bias, high-variance algorithms, diversity in architectures is desirable in ensemble methods. Intuitively, distinct architectures achieving similar validation performance are at most as likely to overfit the same noise in the training data, as closely related ones. Of course, the richer time-series structure of my data could also require deeper architectures to disentangle complex interactions.

II. Data

The stock data come from CRSP and span the period from January 1965 to December 2018 and cover ordinary shares (CRSP codes 10 and 11). To ensure robustness of the results to the potential impact of the size effect discussed in the previous section I deliberately focus on a subset of the largest US equities,[7] constructing the investment universe as follows: on the penultimate trading day of each month I select top five hundred stocks by market capitalization; I then exclude securities which were listed less than a year ago to ensure data availability for computing momentum signals and whose share price is below five dollars – the standard filter to alleviate potential microstructure effects, finally I require that all variables I construct are available for a stock to be included in the sample. The resulting subset includes on average 490 stocks each months and covers 73% of the total US equity market capitalization over the course of the sample.[8] On the final day t of each month for each stock in the sample I construct a number of characteristics (or features) starting with price momentum signals and continuing with variables that previous studies have found to be predictors of momentum returns:

A. Momentum

Similarly to Takeuchi and Lee (2013) I compute cumulative returns over horizons $[t-k, t-1]$ where k is the lookback horizon in days and $k = [1, 2, \ldots, 21, 42, \ldots, 252]$, that is k progresses in one-day increments for the first month and in one-month increments for the rest of the year, thus resulting in a total of 32 momentum features for each stock. The definitions of momentum vary across studies: Jegadeesh and Titman (1993) use 6-month momentum for the bulk of their analysis, Asness, Moskowitz, and Pedersen (2013) use 12-month lookback skipping the most recent month, Novy-Marx (2012) skips the most recent 6 months. Jegadeesh and Titman (1995) document negative short-term momentum returns on the horizons up to a month.

[7] I repeat the analysis for the whole CRSP universe as a robustness check to ensure that my results are comparable with results of other studies. I report this analysis in the Internet Appendix.

[8] The return on a value weighted index comprised of stocks in the sample is also almost indistinguishable from those of the S&P 500 and CRSP value weighted return indexes.

B. *Idiosyncratic Volatility, Market Betas and Pricing Errors*

I estimate the market model using daily data:

$$r_{i,\tau} = \hat{\alpha}_i + \hat{\beta}_i r_{m,\tau} + \hat{e}_{i,\tau}, \tag{1}$$

where $\tau \in [t-k, t-1]$ and $k = [10, 21, 42, 63, \ldots, 252]$ days; $r_{i,\tau}$ is day τ return of stock i; $r_{m,\tau}$ is the market return which I proxy with the return on the CRSP value weighted index, $\hat{\beta}_i$, $\hat{\alpha}_i$, $\hat{e}_{i,\tau}$, are the stock i's loading on the market risk, pricing error, and idiosyncratic return respectively. I use the slope and intercept estimates as gauges of the market exposure of the security and its market-risk adjusted mean return, I further employ the standard deviation of the residual to measure idiosyncratic volatility.[9] I discuss the rationale behind these variables below.

Idiosyncratic volatility. Ang, Hodrick, Xing, and Zhang (2006) document negative price of idiosyncratic volatility in the cross-section of the US stock returns and provide further empirical evidence from international markets (Ang et al. (2009)). However Bali and Cakici (2008) demonstrate that this negative relationship is not robust to stock size and weighting scheme and that the bulk of the negative premium is concentrated in small and illiquid stocks. In their recent large scale replication study Hou et al. (2017) confirm insignificance of idiosyncratic volatility premium once the microcaps are taken into account.

The theoretical relationship between idiosyncratic volatility and momentum returns is positive: in behavioral models (Barberis, Shleifer, and Vishny (1998); Hong and Stein (1999)) idiosyncratic volatility reflects firm-specific information which combined with underreaction to news boosts momentum profits. In rational models, on the other hand, it is an important limit of arbitrage (Shleifer and Vishny (1997)) preventing constrained arbitrageurs from fully exploiting the anomaly. The empirical evidence is however mixed: Arena, Haggard, and Yan (2008) document that momentum profits increase with idiosyncratic volatility and that this increase is primarily driven by the short leg of the momentum portfolio. In the same vein Brav, Heaton, and Li (2009) report that returns on the past losers portfolios decrease with idiosyncratic volatility, but do not find any significant effect in the past winners portfolios, except for small caps where the relationship is also negative. Contrary to these two studies McLean

[9] It is also common in the literature to measure idiosyncratic volatility with respect to the Fama and French (1993) three-factor model (FF-3), however Bali and Cakici (2008) report that the estimates for the idiosyncratic volatility risk premium do not differ significantly between the FF-3 and one-factor model (whether in the excess return or market model formulation). Hou et al. (2017) further corroborate this finding. I chose the single factor parsimonious specification similar to the studies investigating the relationship between idiosyncratic volatility and momentum returns (Arena, Haggard, and Yan (2008), McLean (2010)).

(2010) uses equal weighting arguing that value weighting downplays influence of the high idiosyncratic volatility stocks and finds no significant relationship. Da, Gurun, and Warachka (2014) further argue that the positive relationship between idiosyncratic volatility and momentum might be mechanical: stocks that appear in the extreme past return portfolio are likely to already have higher idiosyncratic volatility. They show that after controlling the idiosyncratic volatility in the formation period return the relationship is non-linear with high idiosyncratic volatility commanding lower return for both past winners and losers.

Market beta. Apart from the CAPM unconditional relationship between higher beta and higher expected return, the momentum has a time-varying exposure to the market risk (Grundy and Martin (2001)). Indeed portfolio sorts on past returns would mechanically load on market risk if the lookback horizon falls co-occurs with bull market states. Daniel and Moskowitz (2016) point out that after market declines the momentum factor invests in low-beta stocks which have suffered smaller losses during the downturn resulting in a sharp negative beta of the momentum portfolio during a subsequent rapid market rebound, leading to momentum crashes. They also argue that the market risk is difficult to hedge by using ex-ante beta estimates – the result further corroborated by Barroso and Santa-Clara (2015) who show that time-varying betas explain a modest proportion of the total momentum risk. I will discuss the results of these to studies in more detail in the section covering market state features later on.

Pricing error. Hühn and Scholz (2018) find the FF-3 alpha to price the cross-section of equity returns in the US and Europe and report the alpha momentum to dominate price momentum in the US. In comparison with the price momentum the alpha momentum strategy experiences slower reversions in post-formation periods providing evidence in favor of underreaction to the stock-specific information. Portfolios sorted on the past alpha deliver less volatile returns and experience less variability in exposures to other risk factors comparing to the price momentum.

The estimated intercept form equation 1 is also a component of the residual (or idiosyncratic) momentum measured by the abnormal returns over look back period with the 'normal' return model fitted over longer time span. As a 'normal' return model Gutierrez Jr and Prinsky (2007) use the one-factor market model, while Blitz, Huij, and Martens (2011) and Blitz, Hanauer, and Vidojevic (2017) employ the FF-3 model; similar to the alpha momentum the residual momentum has less variation in exposures to the risk factors and earns superior risk-adjusted returns. Although I do not model the residual momentum directly my feature set contains all ingredients for the model to learn it from the data: namely, raw returns from the previous sections, estimates of alpha and beta, and market returns discussed below.

C. *Market State Features*

Using the returns on the CRSP value weighted index I estimate the average return and standard deviation of returns over 10, 21, 42, 63, 126, 252, 378, 504 days. The relationship of momentum profitability to past market returns and volatility is nonlinear: Cooper et al. (2004) document higher momentum profitability in the months following the 'up' market states which they define as positive market return over horizons from one to three years and that this profitability is negatively related to the square of past market returns. They further report nonlinearity in the relationship between momentum returns and past market performance: the momentum performance gradually increases from lowest to intermediate quantiles of the past market returns distribution and decays thereafter. Wang and Xu (2015) document that the spread in momentum returns is highest between the low-past-volatility-high-past market-return months and high-past-volatility-low-past-market-return months, and that the bulk of momentum return predictability by the market return and volatility comes from the past losers portfolio. Barroso and Santa-Clara (2015) show the variance of the momentum portfolio to be predictable with its own past realized variance and the realized variance of the market, and that this predictability can be exploited to boost performance of the strategy. Daniel and Moskowitz (2016) find significant asymmetry between up- and down-market betas of the momentum portfolio during the bear markets, with the former being more negative this leads to momentum crashes when the market rapidly rebounds. They report this asymmetry to be driven by the short leg of the portfolio. In contrast to Barroso and Santa-Clara (2015) they also demonstrate that the conditional mean return of the momentum portfolio is predictable using interaction of past market return and volatility and provide evidence of superior performance of the enhanced strategy in international equity markets and other asset classes.

D. *Information Discreteness*

Da et al. (2014) hypothesize that investors are less attentive to information arriving continuously in small amounts comparing to the information arriving infrequently in large amounts, invoking the frog-in-the-pan anecdote. They show that momentum profits are concentrated in stocks where information arrives continuously and that conditioning momentum on information discreteness produces higher risk-adjusted returns. Following their study I define the information discreteness as follows:

$$FIP_{i,\tau} = sign(R_{i,\tau}) \frac{\%neg_{i,\tau} - \%pos_{i,\tau}}{\%neg_{i,\tau} + \%pos_{i,\tau}}, \tag{2}$$

where, as before, $\tau \in [t-1, t-k]$ and $k = [10, 21, 42, 63, \ldots, 252]$ days; $R_{i,\tau}$ is the cumulative return of stock i over the formation period τ; and $\%neg_{i,\tau}, \%pos_{i,\tau}$ are respectively the proportions of days with negative and positive returns over the formation period.

E. Feature Normalization

To facilitate training I normalize the variables with respect to the cross-section by computing z-scores every time period. For the time-series of the market state features I compute z-scores relative to their own history[10] up to the estimation date to avoid look-ahead bias.

[10]The CRSP value weighted return series start in January 1926.

III. Model Selection

This section describes the model, starting with an outline of the network architecture and estimation procedure, then proceeding to an in-depth analysis of hyperparameter optimization. At the end of the section I briefly summarize my approach to building an optimal ensemble of several models and construct a number of metrics to gauge the degree of non-linearity of the model's predictions. I deliberately keep the discussion on the mechanics of neural networks, their training and regularization short, focusing instead on a comprehensive exposition of hyperparameter optimization. I refer the reader to Messmer (2017), Gu et al. (2018) and Chen et al. (2019) for an overview of the neural network training and regularization techniques.

A. Model and Training

Similar to Messmer (2017) and Gu et al. (2018) I use multilayer perceptrons consisting of an input layer, that accepts the vector of features with dimension of 100; several hidden layers in which nonlinear transformations are sequentially applied to the inputs and an output layer that aggregates the results of these transformations into the model's predictions. In contrast to these studies I specify the return prediction as a classification problem instead of regression, predicting the probability of the next month return of a stock being above and below the median return of the entire cross-section. Thus the prediction targets, or labels, for stock i are defined as follows:

$$y_{i,t+1} = \begin{cases} 1, \text{ if } r_{i,t+1} > median(r_{t+1}) \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

I opt for the classification approach because it allows to model the second order effects of inputs on the model's predictions in straightforward fashion. Denote $\hat{p}^a_{i,t+1}$ as the predicted probability of stock i's return being above the median of the cross-sectional return distribution at $t+1$. A binary classifier models this probability as a function of the vector of time t's features $X_{i,t}$ and estimated parameters $\hat{w}$:

$$\hat{p}^a_{i,t+1} = f(X_{i,t}; \hat{w}) = \frac{1}{1 + e^{-h(X_{i,t};\hat{w})}}, \tag{4}$$

where $h(X_{i,t}; \hat{w})$ is the value of the output layer before the logistic transformation, which is always non-linear in parameters for networks with at least one hidden layer

and non-linear activation functions of hidden units. However it is still linear in $X_{i,t}$ when the rectified linear unit function ReLU $:= \max\{0, x\}$ is used as an activation function for the hidden units and makes it impossible to evaluate second and higher order effects of inputs on model's predictions using differentiation. On the other hand, in comparison with other activation functions such as logistic sigmoid and hyperbolic tangent, the ReLU is much more computationally efficient and improves the model training (Nair and Hinton (2010), Glorot et al. (2011)). I therefore use the ReLU activation for hidden units and classification setup to capture higher order effects of features on predictions, albeit at a rather high level.[11] Apart from its ability to capture higher order effects in a trivial way, classification has other advantages: (i) by construction the labels have the same distribution over time and same magnitude, and thus simplify training by alleviating the problem of time-varying cross-sectional dispersion in returns; (ii) under assumption that $p^a_{i,t+1} = \hat{p}^a_{i,t+1} + \varepsilon_{i,t+1}$, $\varepsilon_{i,t+1} \overset{iid}{\sim} F(0)$ the estimated probability is directly proportional to the expected return;[12] (iii) the empirical asset pricing focuses on excess returns of long-short self-financing portfolios and not on point estimates of returns on individual assets, hence the binary classification targets the median portfolios sorted on model's predictions; (iii) minimizing the binary cross-entropy loss function in equation 5 is equivalent to maximizing the log-likelihood of the model, yielding consistent and asymptotically normal MLE estimates of w (White (1989)):

$$\mathcal{L}(w) = -\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left[y_{i,t}\log(\hat{p}^a_{i,t}) + (1 - y_{i,t})\log\left(1 - \hat{p}^a_{i,t}\right)\right] \tag{5}$$

Following Takeuchi and Lee (2013) I split my sample as follows: the training set covers the period from January 1965 to December 1982 and includes 105177 stock-month examples; the validation set is from January 1983 to December 1989 (41408 examples); and the test set is from January 1990 to December 2018 (170385). Similar to Gu et al. (2018) I train the model minimizing the loss 5 using the minibatch stochastic gradient descent with adaptive learning rate algorithm of Kingma and Ba (2014). I further enforce equal representation of classes within each minibatch finding that it greatly improves stability of training especially for small batch size. Additionally I train the model preserving the temporal structure of the dataset reshuffling training examples within each month's cross-section after each epoch.[13]

[11] The higher order effects can also be directly incorporated into the model architecture. For example, to assess multiplicative feature interactions, i.e. the off-diagonal elements of the Hessian of model's outputs with respect to inputs, Cui et al. (2019) introduce a special layer computing the product of inputs.

[12] If returns follow a non-degenerate distribution.

[13] An epoch refers to the entire training set being passed through the neural network during opti-

Due to its high capacity the model can easily overfit the training data, I therefore employ dropout and early stopping as regularization techniques. The dropout (Srivastava et al. (2014)) refers to randomly shutting down a subset of units during each gradient update step, thus effectively training an ensemble of nested models and averaging their predictions. Early stopping monitors the loss achieved by the model on the validation set and terminates the estimation once this loss starts to increase.[14] Dropout probability,[15] batch size, learning rate, number of hidden layers, number of units in each hidden layer, maximum number of epochs to train the model and the patience for early stopping constitute the hyperparameters of the model optimization of which I discuss in the following section.

Finally, I estimate the model using price returns both for construction of the stock-specific features[16] and definition of the class labels. Note that the function estimating the probability in 4 depends neither on time nor on individual stock, while the dividend yield exhibits substantial time-series variation over the sample from over 5 percent p.a. in the early 1980s to on average 2 percent from 1990 onwards. Since I do not include any dividend-related information into the features, the model by construction can not capture the dividend yield component of the total return and would likely overfit the training set to the to the prevailing dividend yield environment.[17,18] It is also important to point out that at monthly forecasting frequency the dividend yield component of the total return is almost always known to investors before the position is opened as the declaration date precedes the ex-date by more than a month for the overwhelming majority of stocks, which makes the joint prediction of the dividend yield and price return components of total return excessive especially when taking in account the persistence and predictability of the former. As a robustness check I repeat all the analysis with the features constructed from total returns with targets being both price and total returns. I find that whether to construct features from price or total returns has almost no impact on the test set performance of the model, while using total returns as targets slightly impairs the ability of the model to generalize on the test set, although without any consequences for economic or statistical significance of the results.[19] I report these analyses in the Internet Appendix. Naturally, I use total

mization.

[14]Dropout leads to sparse unit activations and can be interpreted as l_1-penalty on the corresponding layer's weights, whereas early stopping is similar to the l_2 regularization (Bishop (1995)).

[15]I apply dropout only to units within hidden layers.

[16]For consistency I also use price return on CRSP value weighted index in regression 1.

[17]I find that training the model to predict the total return distribution results in higher objective function loss on the validation set comparing to predicting the price distribution; and that the choice whether to use price or total returns for feature construction is almost inconsequential.

[18]This critique also applies to models of Messmer (2017) and Gu et al. (2018), who although include dividend yield into features also normalize the features with respect to cross-section and do not take in account the time-series variation in the dividend yield component of total returns.

[19]Indeed, over the whole sample, the features constructed using total or price returns are almost

returns for the analysis of performance of neural network-based portfolios and asset pricing tests in any of the cases.

B. *Hyperparameter Optimization*

The validation performance y of a model can be represented as a function $f : \Theta \to \mathbb{R}$ of its hyperparameters $\theta \in \Theta$, therefore the optimal set of hyperparameters is given by $\theta^* \in \arg\min_{\theta \in \Theta} f(\theta)$. Direct evaluation of f is, however, infeasible. Bayesian optimization replaces the true objective with a probabilistic surrogate model $p(y|D_i)$ where $D_i = \{(\theta_k, y_k)\}_{k=1}^{i}$ is a sequence of accumulated evaluation observations. Each iteration the TPE algorithm of Bergstra et al. (2011) proposes a new set of hyperparameters θ_{i+1} that maximizes expected improvement:

$$\mathbb{E}[I(\theta_{i+1})] = \int_{-\infty}^{\infty} \max\{\bar{y}_i - y_{i+1}, 0\} dp(y_{i+1}|\theta_{i+1}, D_i) \tag{6}$$

The expected improvement is the expectation under the model $p(y_{i+1}|\theta_{i+1}, D_i)$ that the performance y_{i+1} (in terms of loss) will be below some threshold value $\bar{y}_i$ which is defined as an empirical quantile γ of the accumulated values of the objective function $\{y_k\}_{k=1}^{i}$. The TPE splits the history D_i into two sets where values of the objective function are below and above $\bar{y}_i$ and estimates the respective probability distributions of hyperparameters $l(\theta)$, $g(\theta)$ using adaptive Parzen windows. Bergstra et al. (2011) demonstrate that maximizing the ratio $l(\theta)/g(\theta)$ is equivalent to maximizing the expected improvement in equation 6. The algorithm then samples n_{EI} values from $l(\theta)$, evaluates them according to $l(\theta)/g(\theta)$ and uses the value maximizing the expected improvement to evaluate the objective for the next iteration. Appendix A provides a detailed description of the TPE algorithm.

I define the hyperparameter optimization objective function y as follows: for a set of hyperparameter values I first estimate the model five times and pick the five best values of the validation loss achieved by each model, the value of the objective is the average validation loss over these 25 values. Since the training is largely stochastic, I am explicitly looking for architectures that can consistently achieve lower loss both within each estimation run and across different runs. Indeed Messmer (2017) reports that over half of the hyperparameter space configurations he estimates using random search fail to improve the validation loss. I further set the proportion of observations γ used to estimate $l(\theta)$ to 0.2, and the number of draws from $l(\theta)$ to evaluate the expected improvement n_{EI} to 100. I initialize the TPE algorithm by accumulating the

perfectly correlated, while the price and total return labels are only 95% correlated.

initial observations $D_{init} = (y_k, \theta_k)_{k=1}^{25}$ by drawing 25 samples from prior distributions of the hyperparameters and evaluating the objective. I fix the maximum number of epochs for each evaluation at 250 and set the patience of early stopping to 15 epochs, meaning that optimization is terminated if the validation loss has not improved over the last 15 epochs. Table I summarizes the prior distributions of the rest of hyperparameters. I use the Hyperopt package (Bergstra et al. (2013)) for the numerical implementation of the TPE.

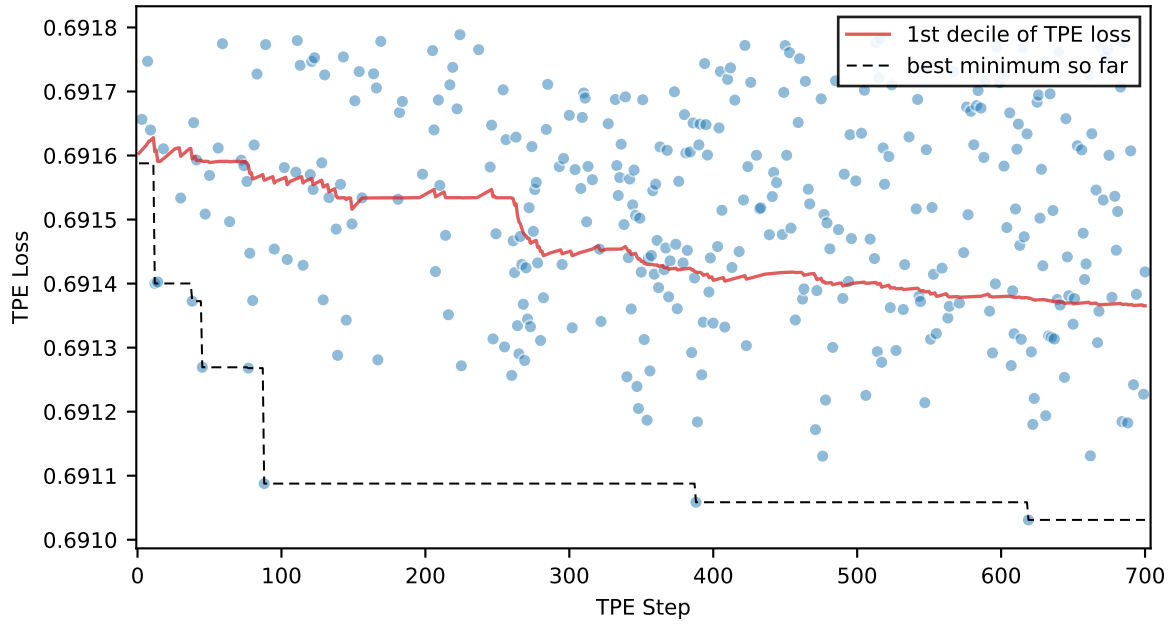
Table I: Prior distributions of hyperparameters

Hyperparameter	Prior Distribution
Learning rate	$10^u,\ u \sim U(-5.75, -2.95)$
Batch size	$\{32, 64, \ldots, 256\}$
Dropout rate	$U(0, 0.5)$
Number of hidden layers	$\{2, 3, \ldots, 6\}$
Number of units in each layer	$\{20, 25, \ldots, 170\}$

I run 700 TPE evaluations. Figure 1 shows the progress of the TPE hyperparameter optimization. The iterations and the TPE objective function values are along the horizonal and vertical axis respectively. The solid red line is the expanding first decile of the TPE loss and the dashed black line tracks the best TPE loss at each iteration. The blue dots depict the best 50% of the TPE evaluations. Over time the algorithm consistently proposes hyperparameter configurations resulting in lower loss. Figure 2 shows how TPE adjusts the distributions of hyperparameters over iterations. In each plot the solid blue and red lines draw the kernel density estimates of the distributions of hyperparameters tried in the first and second halves of the TPE iterations respectively, and the dashed black line shows the prior distribution as presented in Table I. The plots depict densities for (from top to bottom and left to right) base-10 logarithm of the learning rate, batch size, dropout rate, number of layers, and number of units in the first and last hidden layer. For clarity of exposition I depict the discrete uniform priors for batch size, number of layers and number of units with continuous uniform densities over the same support. As TPE iterations progress the algorithm picks shallower architectures with smaller learning rate and batch size and larger dropout rate. The increase in the number of units in the first hidden layer is likely to be an artifact of the increasing dropout rate. Srivastava et al. (2014) point out that dropout induces sparsity in the activations of the hidden units and reduces the model capacity during training, therefore a greater number of units may be required to carry the inputs deeper through the network.

The key disadvantage of the TPE is that it models hyperparameters using unidimensional kernel density estimates and thus by assuming hyperparameter independence

Figure 1: TPE hyperparameter optimization progress



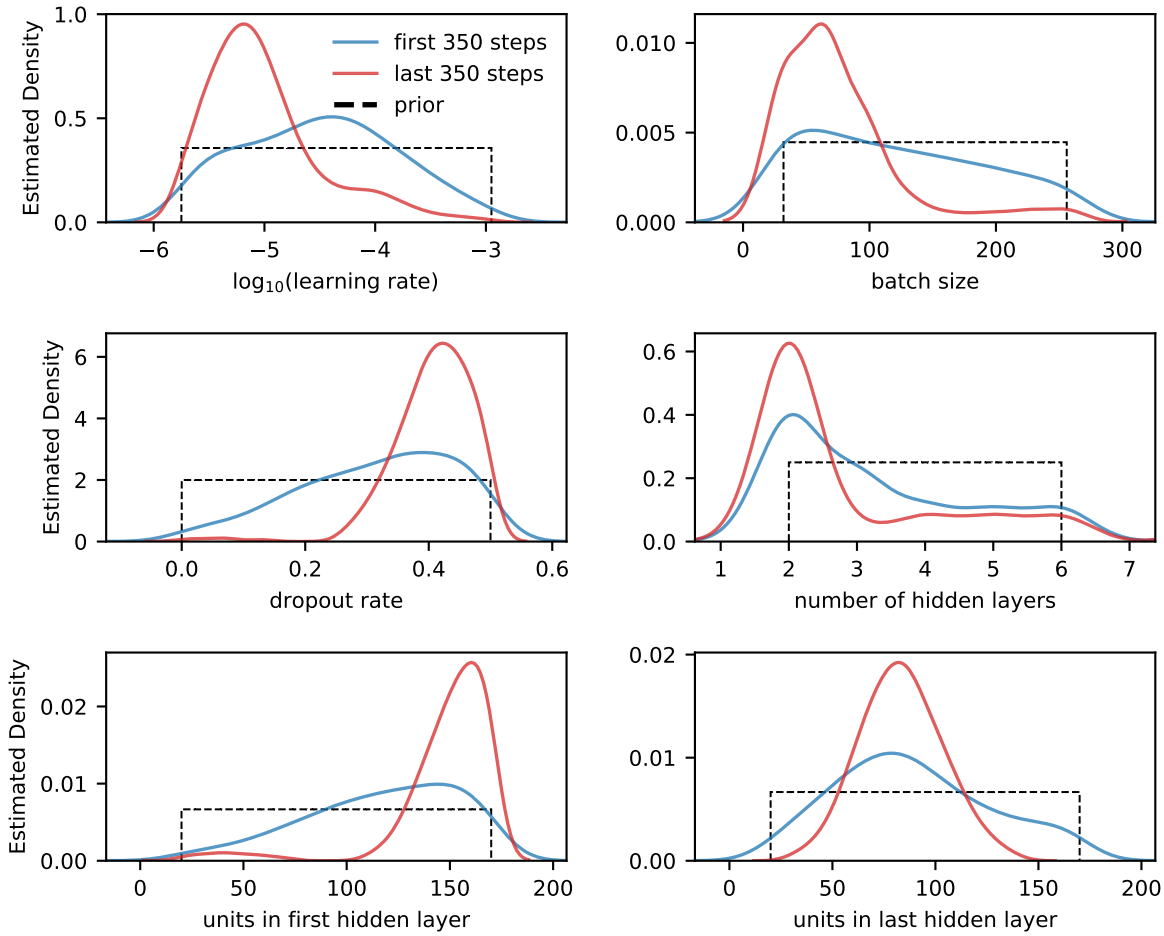
This figure depicts progress of the TPE hyperparameter optimization. The iterations and the TPE objective function values are along the horizontal and vertical axis respectively. The blue dots show the best 50% of the TPE evaluations, the solid red line is the expanding first decile of the TPE loss and the dashed black line tracks the best TPE loss at each iteration.

it ignores interaction effects. Figure 3 highlights this issue by plotting the TPE loss as functions of the learning rate (left plot) and dropout rate (right plot). The blue dots depict the TPE losses of the shallower architectures with only two hidden layers, and the red dots show the losses of the deeper architectures with more than two hidden layers. Although both the shallow and deep models exhibit a U-shaped relationship between the TPE loss and hyperparameter values, the loss of the deep models starts to deteriorate much earlier with increase in the dropout and reduction in the learning rate. Since the hyperparameter interactions are not taken into account, the TPE consistently proposes, for example, lower values of the learning rate in every hyperparameter configuration resulting in higher loss for the deeper models.[20] As observations accumulate the algorithm enters a positive feedback loop overexploiting the shallow architectures and underexploring the deeper ones.[21] I address this issue by running the second TPE pass for a hyperparameter space that excludes shallow models, thus forcing the algorithm to explore deeper architectures. In addition to the lower dropout and higher learning rates, the best hyperparameter configurations also exhibit larger

[20]Recall that I allocate the computational budget in terms of epochs equally for every hyperparameter configuration, and deeper architectures might require more training steps to achieve optimal loss with low learning rate or high dropout.

[21]The TPE quite aggressively converges to the shallow architectures: out of top 30 models by TPE loss there are only two with more than two hidden layers. Over the first 350 TPE iterations the shallow architectures constitute 44% of all evaluated models, this share rises to over 70% for the last 350 TPE steps.

Figure 2: Distributions of hyperparameters during TPE optimization

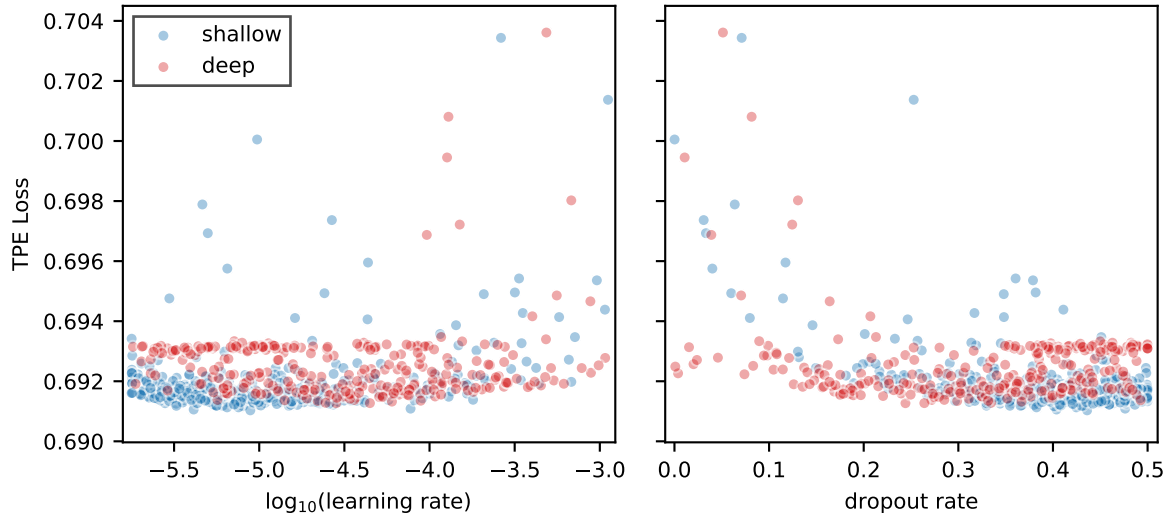


This figure displays distributions of hyperparameters over the course of TPE optimization. In each plot the solid blue and red lines draw the kernel density estimates of distributions of hyperparameters tried in the first and second halves of the TPE iterations respectively, and the dashed black line shows the prior distribution as presented in Table I. The plots depict densities for (from top to bottom and left to right) base-10 logarithm of the learning rate, batch size, dropout rate, number of layers and number of units in the first and last hidden layers. The discrete uniform priors for batch size, number of layers and number of units are drawn as continuous uniform densities over the same support.

batch size which further distinguishes them from their shallower counterparts. In terms of the TPE loss they slightly outperform the shallow models comprising 50, 70, and 62 percent of models yielding the best 10, 20, and 100 TPE losses in the pooled sample of 1400 evaluations over the two TPE runs. Appendix B discusses the results of the second run in more detail. In addition to that the best performing deep architectures are rather diverse: out of the top ten models with more than two hidden layers there are three networks with 4 hidden layers and one with five, with the number of units in each hidden layer experiencing substantial variation as well. This diversity is a desirable characteristic for constructing an ensemble of models – indeed predictions of structurally different models are likely to be less correlated than predictions of, basically, several copies of the same model from the first pass of TPE optimization, thus 'diversifying away' the overfitting of idiosyncratic noise inherent to each particular

architecture.

Figure 3: Hyperparameters of shallow vs. deep models



This figure plots the TPE loss as a function of learning rate (left plot) and dropout rate (right plot). The blue dots depict the TPE losses of the shallower architectures with only two hidden layers, and the red dots show the losses of the deeper architectures with more than two hidden layers.

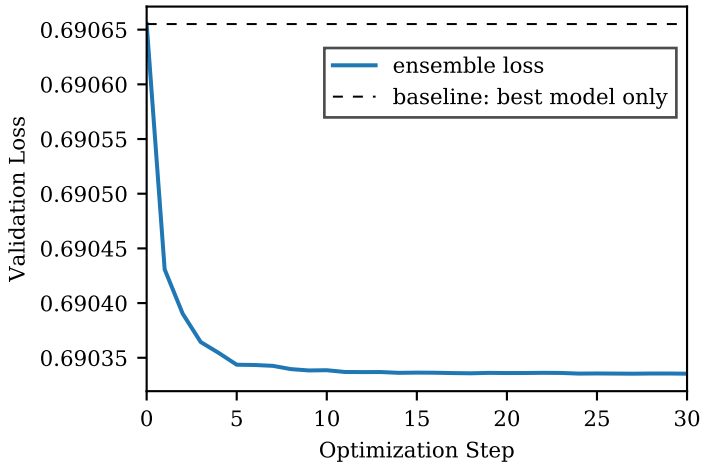
The problem of hyperparameter independence in the TPE can also be addressed directly. For example, in a recent study Falkner et al. (2018) estimate a single multidimensional KDE in the Bayesian optimization part of their algorithm.

C. *Ensemble Selection*

I pick top ten model architectures from each of the two TPE runs as initial candidates for the ensemble and then follow the Caruana et al. (2004) algorithm: starting from an ensemble of size one with the best model, each iteration I add a new model from the model pool (with replacement) such that the average prediction of the ensemble yields the lowest validation loss. Figure 4 plots the ensemble's validation loss over 30 algorithm iterations: the black dashed line corresponds to the loss of the best model and the blue solid line plots the ensemble loss as optimization progresses.

After approximately twelve iterations the algorithm stops to consider new models and continues to adjust the weights of its constituents instead exhibiting no significant improvement in the validation loss after 15 updates. After 30 updates the ensemble achieving the best loss consists of eight architectures with two shallower networks from the first TPE pass having a combined weight of 40 percent. Since the ensemble optimization is computationally cheap I re-optimize the ensemble before every prediction on the test set using the newly available information.

Figure 4: Validation loss during ensemble optimization



This figure depicts the validation loss of the ensemble over the course of the Caruana et al. (2004) optimization procedure. The black line shows the validation loss of the best model and the blue solid line plots the ensemble loss as optimization progresses.

D. Variable Importance

Together with being praised as powerful function approximators neural networks are often criticized for their black box properties and lack of interpretability of their predictions. In response to this criticism machine learning literature offers an array of methods to measure impact of input variables and their interactions on predictions of models.[22] In this paper I employ gradients and Hessians of estimated probabilities with respect to the input variables to identify key drivers behind the ensemble's predictions. The gradients are direct analogues to slope coefficients in linear regression and allow to gauge the effect of features on predictions without looking at estimated parameters. Similar to Simonyan et al. (2013) and Hechtlinger (2016) I compute gradients and Hessians at the point of prediction for each example in the test set.[23]

I further construct several metrics to capture the aggregate cross-sectional and time-series variation in the degree of non-linearity of model's predictions in input variables and the magnitude of interaction effects.

[22]See Tsang et al. (2017) and references therein for an overview of different methods of analyzing variable importance and detecting feature interactions.

[23]There are also several extensions of the gradient-based approach addressing potential problems arising from discontinuities in the gradients of ReLU units (Sundararajan et al. (2017), Shrikumar et al. (2017), Bach et al. (2015)).

D.1. Time-series variation in non-linearity

Given the predicted probability $\hat{p}^a_{i,t}$ defined in 4, denote $\mathbf{g}_{i,t} = \nabla \hat{p}^a_{i,t}$ as the gradient with respect to the input variables and $\mathbf{H}_{i,t+1}$ as the corresponding Hessian matrix. Consider the following ratio:

$$Q_{i,t} = \frac{\mathbf{1}^\top |\mathbf{H}_{i,t}| \mathbf{1}}{2 \left\| \mathbf{g}_{i,t} \right\|_1} = \frac{1}{2} \left\| \mathbf{g}_{i,t} \right\|_1 \left| \frac{1 - 2\hat{p}^a_{i,t}}{\hat{p}^a_{i,t}(1 - \hat{p}^a_{i,t})} \right| = \frac{1}{2} \left\| \mathbf{g}_{i,t} \right\|_1 z_{i,t} \tag{7}$$

Intuitively this ratio measures the relative magnitudes of the first and second order Taylor approximation terms around the point which is 1 standard deviation away from $X_{i,t-1}$ for all variables, and thus captures the local curvature of the decision boundary. The second equality follows from Proposition 1 and decomposes the ratio into two parts: (i) $\left\| \mathbf{g}_{i,t} \right\|_1$ is higher for stocks with higher impact of input variables on predicted probabilities; (ii) $z_{i,t}$ increases with absolute deviation of the predicted probability from 0.5.

Proposition 1. *In a single-output multilayer perceptron with logistic sigmoid activation of the output unit and ReLU activations of the hidden units the second order partial derivatives of the output $p = f(X)$ with respect to the i-th and j-th elements of the input vector $X = [x_1, x_2, \ldots, x_N]^\top$ are given by:*

$$\frac{\partial^2 p}{\partial x_i \partial x_j} = \frac{\partial p}{\partial x_i} \frac{\partial p}{\partial x_j} \frac{1 - 2p}{p(1 - p)} \tag{8}$$

Proof. See Appendix C.

The cross-sectional average of 7 measures the average curvature of the decision boundary at a given point in time:

$$\bar{Q}_t = \frac{1}{2N} \sum_{i=1}^{N} \left\| \mathbf{g}_{i,t} \right\|_1 z_{i,t} \tag{9}$$

Once again the degree of non-linearity increases with larger cross-sectional dispersion of the predicted probabilities and the absolute magnitude of the gradients. An important feature of 9 is that time-variation in the ratio captures to an extent the aggregate impact of the market state variables and their interactions with other inputs. [24] It is

[24] Consider the following illustrative example: assume there is only two variables in the model, say, 12-month momentum and market return. Further assume that the momentum input is normalized in a way that guarantees that the cross-sectional distribution of the variable is the same at every time step

straightforward to compute the non-linearity ratio of a portfolio:

$$Q_t^P = \frac{\sum_{p=1}^{P} w_p \left\| \mathbf{g}_{i,t} \right\|_1^2 z_{i,t}}{2 \sum_{p=1}^{P} w_p \left\| \mathbf{g}_{i,t} \right\|_1}, \quad \sum_{p=1}^{P} w_p = 1 \tag{10}$$

D.2. Cross-sectional non-linearity

To gauge the average shape of the decision boundary over the cross-section I follow the following procedure: each month in the test sample I compute the first order Taylor approximation of the predicted probabilities around the median prediction $p^a_{m,t+1}$; I then sort the rest of the cross-section into deciles by $d_t = ||X_{i,t} - X_{m,t}||_2$ – the L^2 distance between the features vector of a stock and that of the median prediction, such that the first and tenth deciles correspond to observations closest and farthest from the median prediction in the Euclidean space; finally I aggregate these bins over months and compare the average losses 5 of the model and its first order approximation within each decile. The key idea of this approach is that the market state variables stay constant within the cross-section and hence the distance d_t depends only on dispersion in the cross-sectional features. The difference between the two losses is then higher the larger the aggregate impact of this dispersion on the predictions.

(e.g. by using quantiles). Thus if the market return variable, common for all stocks, does not vary over time, so do not the distributions of the predictions and the gradients with respect to the inputs as the model, and therefore any time variation in 9 would be due to changes in the market return. Of course, in reality interactions between cross-sectional features, for example, market beta and past return can implicitly convey information about the market state, e.g. relative outperformance of low beta stocks during a market crash. Also standardization of features does not guarantee time-invariance of the distribution of the cross-sectional inputs. I nevertheless find empirically that the cross-sectional part of $\|\mathbf{g}_{i,t}\|_1$ experiences very little variation over time if the market state variables are kept at their mean values.

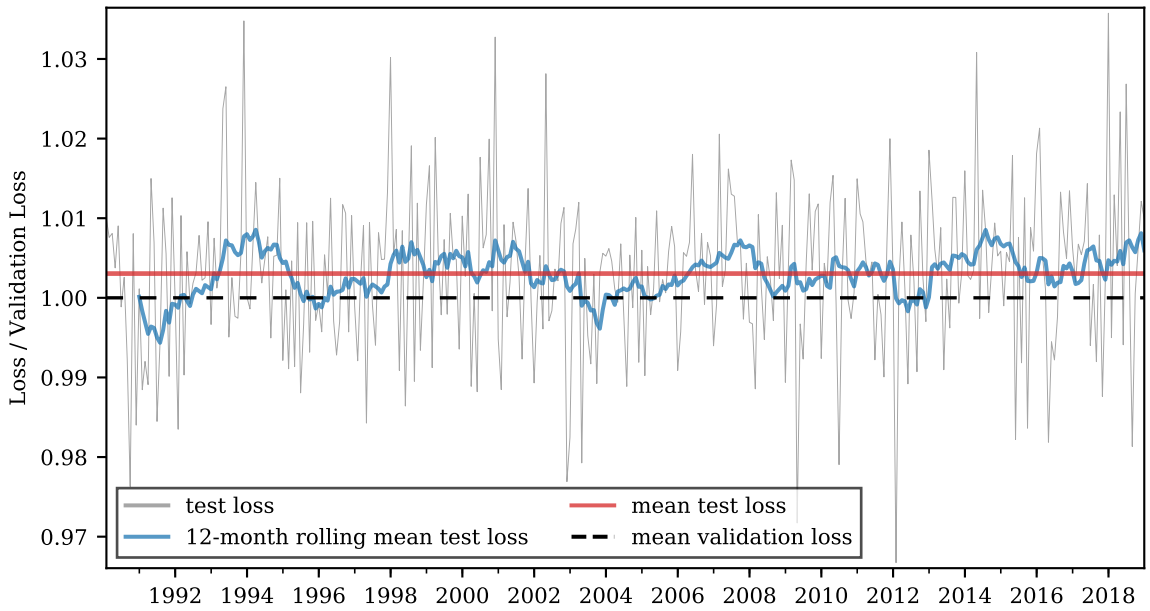
IV. Results

This section presents the results of the paper. I start with analyzing out-of-sample performance of the model and evaluating performance of portfolios based on the predictions of the ensemble. I then proceed to the variable importance analysis and examine the variation in non-linearity of the predictions. The section concludes with looking at the standard 12-1 momentum strategy through the lens of the deep learning model and series of robustness checks.

A. Out-of-Sample Performance

Figure 5 shows the out-of-sample performance of the neural network ensemble relative to its performance on the validation set (dashed black line normalized to 1) in terms of loss 5. The red line depicts the average loss on the test set, the gray line shows cross-sectional average of the loss at each month; and the blue line draws the 12-month rolling mean of this average. The model generalizes well: comparing to the performance on the validation sample the loss increases on average by a third of a percent and fluctuates around its mean over the course of the test sample.

Figure 5: Out-of-sample performance

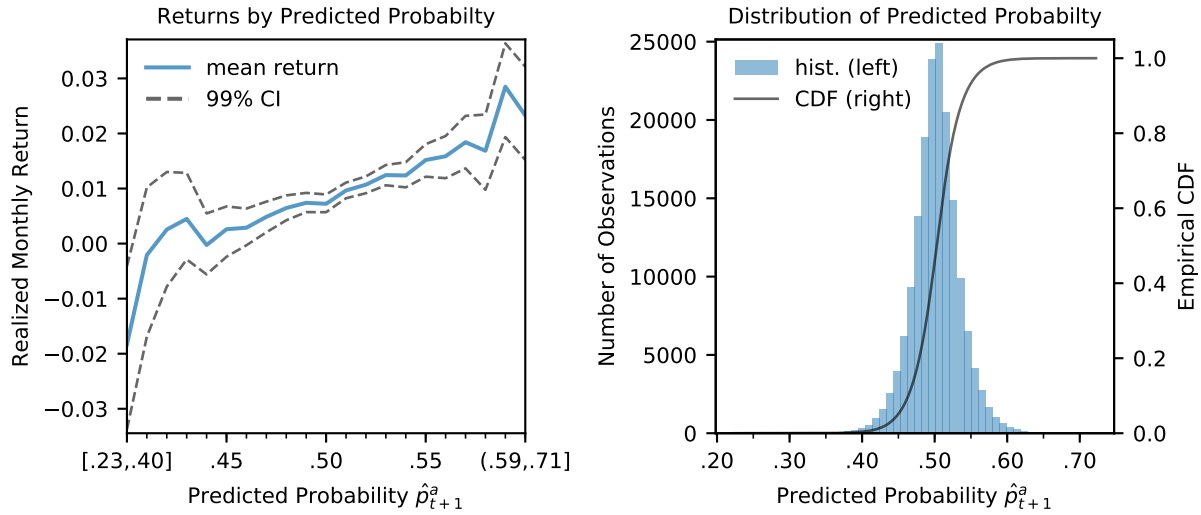


This figure displays out-of-sample performance of the neural network ensemble relative to its performance on the validation set (dashed black line normalized to 1). The red line depicts the average loss on the test set, the gray line shows cross-sectional average the loss at each month, and the blue line draws the 12-month rolling mean of this average.

Figure 6 summarizes the aggregate relationship between the predicted probabilities

and realized returns. The left plot displays realized monthly return as a function of predicted probability of stock return being above the cross-sectional median return in the next month. Each (x, y) point shows the average return over all stocks and months for the estimated probability bin $\hat{p}^a_{t+1} \in (x - 0.01, x]$ except for the left- and right-most points which report the averages over extremes of the empirical distribution of predicted probabilities which together contain around 1 percent of all observations. The dashed black lines draw the corresponding 99% bootstrapped confidence bounds. The right plot shows the histogram of the predicted probabilities (in blue, against the left y-axis) and their empirical CDF (solid black line, against the right y-axis). Because of the ensemble averaging a large amount of the probability density is concentrated around the mean: more than 95% of the predicted probabilities are between 43 and 57 percent. For the vast majority of the observations the relationship between the predicted probability and realized return is positive and monotone: on average the monthly return increases from around zero in the first quartile of the distribution of predicted probabilities to over 1 percent in the top quartile. The ensemble also captures the cross-sectional distribution of returns – over 348 months the average Kendall's tau for the cross-sectional correlation between the predicted probability of return being above the cross-sectional median and realized returns is 3.2% with a bootstrapped 99% confidence interval of [2.1%, 4.3%]. The corresponding figures for Spearman's rho are 4.7% and [3.0%, 6.1%].

Figure 6: Out-of-sample predicted probabilities and realized returns



The left plot displays realized monthly return as a function of predicted probability of stock return being above the cross-sectional median return in the next month. Each (x, y) point shows the average return over all stocks and months for the estimated probability bin $\hat{p}^a_{t+1} \in (x - 0.01, x]$ except for the left- and right-most points which report the averages over extremes of the empirical distribution of predicted probabilities. The dashed black lines draw the corresponding 99% confidence bounds. The right plot shows the histogram of the predicted probabilities (in blue, against the left y-axis) and their empirical CDF (solid black line, against the right y-axis). The sample is from January 1990 to December 2018.

B. *Neural Network Ensemble Portfolios and Time-Series Asset Pricing Tests*

Next, I construct portfolios using the ensemble's predictions as signals. First, I compute excess returns by subtracting the one month Treasury bill rate. Then at the end of each month I sort the stocks into equally weighted portfolios based on the predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. I also construct long-short 'high-minus-low' portfolios investing in stocks with highest predicted probabilities and funding this position by shorting stocks with lowest predicted probabilities. Table II reports descriptive statistics of excess returns on these portfolios. Each triplet of columns shows descriptive statistics of a low, high, and high-minus-low portfolio for median, quintile and decile sorts. Means, medians, standard deviations are in percent p.a., Sharpe ratios are annualized, maximum drawdowns and maximum one-month loss are in percent, average turnover is in percent per month. The numbers in brackets are the Newey and West (1987) t-statistics for the null hypothesis of mean return being zero with number of lags according to Newey and West (1994).

Table II: Descriptive statistics: long-short portfolios

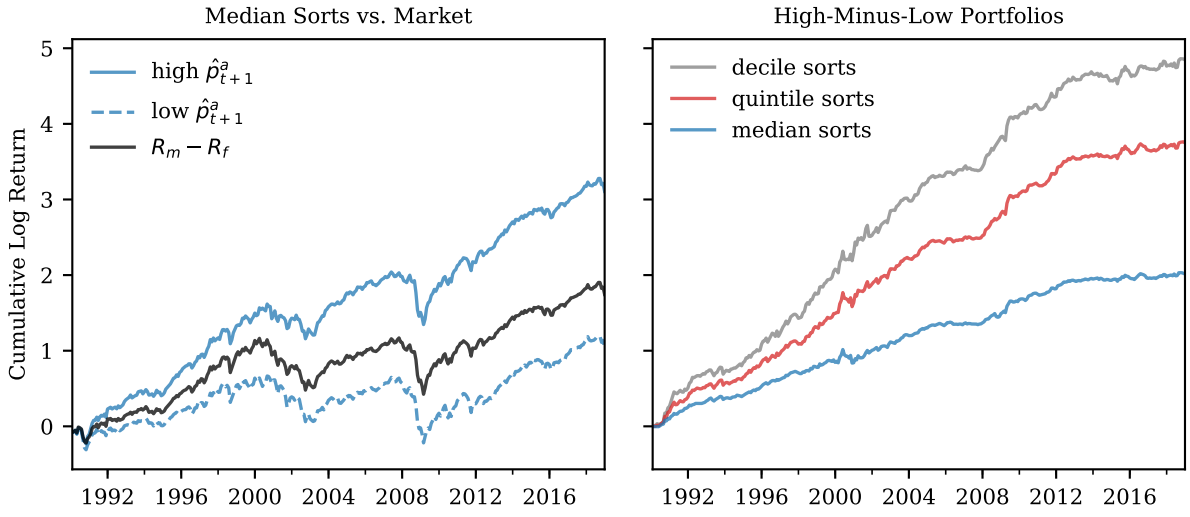
	Median			Quintile			Decile		
Portfolio	1	2	2-1	1	5	5-1	1	10	10-1
Mean	4.81	11.99	7.18	1.22	14.72	13.50	−1.29	16.38	17.67
[t-stat]	[1.58]	[3.88]	[6.84]	[0.37]	[4.76]	[7.84]	[−0.37]	[4.98]	[7.84]
Median	11.51	16.83	5.88	7.51	17.18	11.30	2.66	19.02	13.50
Std	15.50	15.56	6.44	16.37	16.10	10.08	18.15	17.16	13.16
Skew	−0.73	−0.64	0.01	−0.78	−0.32	0.56	−0.65	−0.04	0.64
Kurtosis	1.63	2.26	4.59	2.08	2.52	3.47	2.24	3.18	3.51
Sharpe	0.31	0.77	1.11	0.07	0.91	1.34	−0.07	0.95	1.34
MaxDD	58.72	50.11	16.41	74.94	49.39	17.00	84.23	50.85	14.74
Max 1 M loss	19.68	20.45	9.69	22.09	20.22	11.13	24.68	21.51	13.44
Turnover	78.96	76.13	155.08	122.39	122.35	244.74	140.15	143.19	283.34

The table reports descriptive statistics of excess returns on portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. Each triplet of columns shows descriptive statistics of a low, high, and high-minus-low portfolio for median, quintile and decile sorts. Means, medians, standard deviations are in percent p.a., Sharpe ratios are annualized, maximum drawdowns and maximum 1-month losses are in percent, average turnover is in percent per month. The numbers in brackets are the Newey and West (1987) t-statistics for the null hypothesis of mean return being zero with number of lags according to Newey and West (1994). The sample is from January 1990 to December 2018.

The portfolio effectively shorting half of the S&P 500 constituents and investing in the other half earns on average 7.2% p.a., which is statistically significant at any conventional level. The spread between high and low portfolios increases as sorts become more aggressive to 17.7% p.a. for the difference in the extreme decile portfolios. This

increase comes simultaneously from higher (lower) returns on the 'high' and 'low' portfolios, providing additional evidence of the model capturing the cross-sectional distribution of expected returns. The annualized Sharpe ratio rises from 1.11 to 1.34 for the median and decile sorts respectively. Returns on the portfolios also become more positively skewed for the more concentrated sorts. Maximum drawdown of any of the long-short portfolios is around 16% similar to the equally weighted neural network portfolios in Gu et al. (2018), however it is four times lower than the drawdown of their value-weighted portoflios. The monthly turnover is on average two times higher than that of standard 12-1 momentum portfolios in this sample. Figure 7 plots cumulative log returns on the ensemble portfolios. The left panel shows the returns of the high and low portfolios sorted by median $\hat{p}^a_{t+1}$ (solid and dashed blue lines) along with the return on the CRSP value weighted index in black. The right panel displays the returns on the long-short portfolios: median, quintile, and decile sorts are drawn in blue, red, and gray respectively. With respect to the overall out-of-sample performance my results for the decile sorts are very close to those reported by Gu et al. (2018) for value weighted decile portfolios for the entire CRSP universe.

Figure 7: Return on predicted probability portfolios



The figure plots out-of-sample performance of portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The left plot shows cumulative log excess returns on equally weighted portfolios investing in stocks with 50% highest and lowest $\hat{p}^a_{t+1}$ (solid and dashed blue lines) and on the CRSP value weighted index (solid black line). The right plot displays cumulative log returns on long-short predicted probability portfolios: median, quintile, and decile sorts are drawn in blue, red, and gray respectively. All portfolios are rebalanced monthly. The sample is from January 1990 to December 2018.

To sum up, the model is capable to capture the cross-section of expected returns well. Table III further illustrates this point by reporting the descriptive statistics for each of the decile portfolios. The relationship between the predictions of the model and expected returns is monotone but rather non-linear: the bulk of the spread in the perfor-

Table III: Descriptive statistics: decile portoflios

Portfolio	1	2	3	4	5	6	7	8	9	10
Mean	−1.29	3.70	7.31	7.06	7.21	9.59	9.33	11.57	13.02	16.38
[t-stat]	[−0.37]	[1.18]	[2.47]	[2.23]	[2.30]	[2.86]	[2.87]	[3.58]	[4.23]	[4.98]
Median	2.66	9.82	13.86	11.29	12.3	13.35	13.35	13.06	16.21	19.02
Std	18.15	15.57	15.53	16.26	15.82	16.45	15.80	15.93	15.72	17.16
Skew	−0.65	−0.72	−0.69	−0.62	−0.44	−0.58	−0.66	−0.77	−0.55	−0.04
Kurtosis	2.24	1.75	1.47	1.89	0.63	2.05	1.93	2.45	2.25	3.18
Sharpe	−0.07	0.24	0.47	0.43	0.46	0.58	0.59	0.73	0.83	0.95
MaxDD	84.23	62.00	53.77	55.69	52.58	54.62	52.54	49.18	48.04	50.85
Max 1 M loss	24.68	19.56	18.42	21.12	16.27	21.68	20.51	19.65	18.93	21.51
Turnover	140.15	168.23	173.25	175.12	176.44	175.58	175.45	173.03	167.67	143.19

The table reports descriptive statistics of excess returns on decile portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. Means, medians, standard deviations are in percent p.a., Sharpe ratios are annualized, maximum drawdowns and maximum 1-month loss are in percent, average turnover is in percent per month. The numbers in brackets are the Newey and West (1987) t-statistics for the null hypothesis of mean return being zero with number of lags according to Newey and West (1994). The sample is from January 1990 to December 2018.

mance of the long-short portfolios comes from the extreme quintiles of the predicted probabilities.

Can the established risk factors explain these returns? No, they can not. Table IV reports results of time-series regressions for excess returns on portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The test portfolios are across rows and explanatory variables are across columns. The first column reports pricing errors in percent p.a.; the regressors include the five factors of Fama and French (2015) plus 12-1 momentum.[25] The last column shows the adjusted R^2. Panels A and B report results for median and decile sorts respectively. The numbers in brackets are the Newey and West (1987) t-statistics with number of lags according to Newey and West (1994). The loadings on the factors other than the market risk premium are tiny and rarely statistically significant, furthermore the risk factors explain less than 2 percent of the variation in the returns on the long-short neural network portfolios, leaving the long-short alphas positive and highly significant. Given that my variables have time dimension I repeat the asset pricing tests on a shorter sample with trend following factors of Fung and Hsieh (2001) that aim to capture returns of trend following hedge funds and CTAs.[26] The results remain virtually unchanged: R-squared increases from 2 to 4 percent for both median

[25]The data are from Kenneth French's web page (`https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html`).

[26]The data come from David A. Hsieh's data library (`https://faculty.fuqua.duke.edu/~dah7/HFRFData.htm`) and start in 1994.

and decile sorts. I report these results in Table D.I in Appendix.

C. *Which Features Drive Predictions?*

Figure 8 shows partial derivatives of the predicted probabilities of stock return being above the cross-sectional median return in the next month with respect to model's inputs. The top and bottom 10 input variables ranked by their average gradient are across the vertical axis. For a given feature the colored bars and whiskers represent respectively the interquartile and 5-95% range of all gradient evaluations on the test set. The solid black lines and dots inside each bar show median and mean value of the gradients. Since the variables are normalized to have mean of zero and standard deviation of one the interpretation is as follows: keeping other things equal an increase in one-year alpha of a stock by a small Δ relative to the cross-section increases the predicted probability of stock return being above the cross-sectional median in the next month by approximately $100\times\Delta\%$.

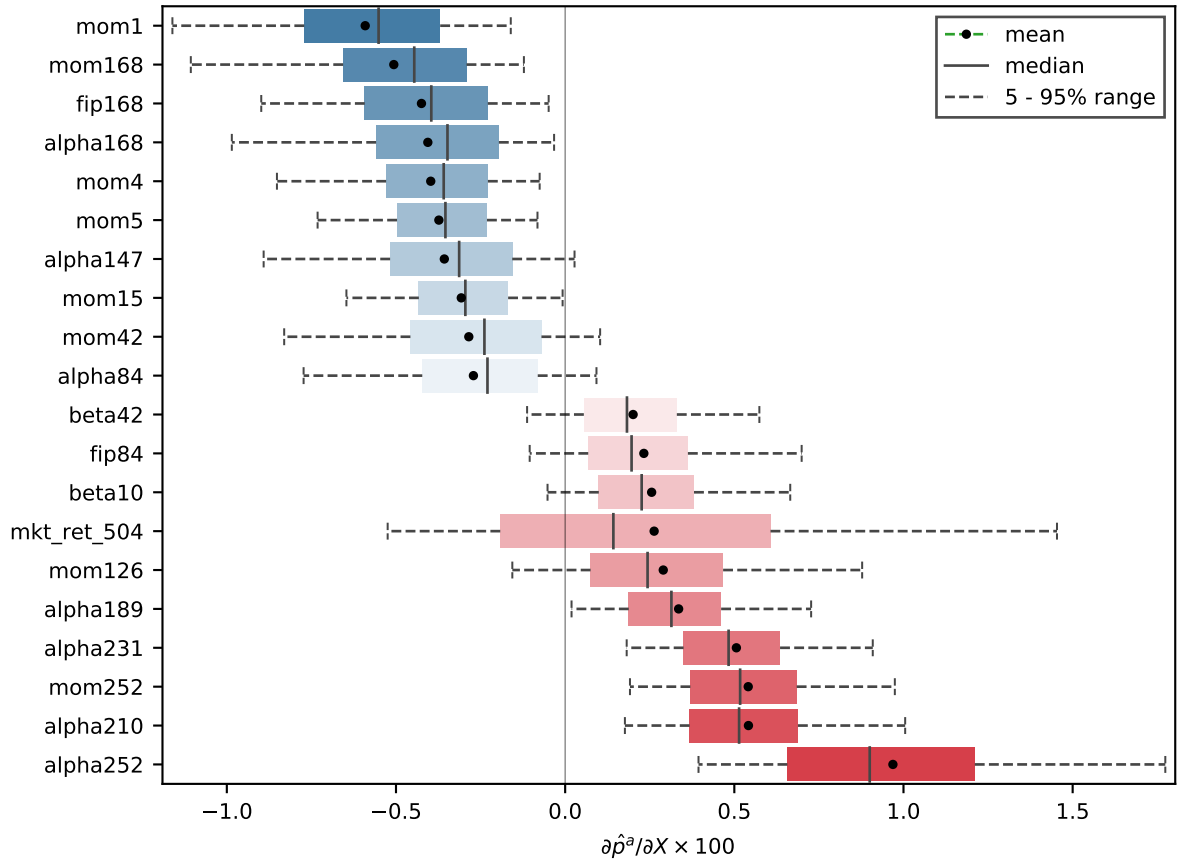
Essentially Figure 8 reports *unconditional* predictors of expected returns. The most salient cross-sectional features predicting positive return are the market model alpha over horizons from nine months to one year, corroborating the findings of Hühn and Scholz (2018), along with the six months and one year price momentum. The latter echoes to an extent the results of Novy-Marx (2012) who reports that 7-12 months momentum subsumes other momentum signals. In fact the one-year alpha is extremely robust: out of more than 170,000 observations only two have negative gradients with respect to this variable. The only market state variable among the unconditional predictors is the 2-year market return which falls right into the middle of the range of market return horizons identified by Cooper et al. (2004) to predict momentum returns; and which Daniel and Moskowitz (2016) further identify as a bear market indicator and predictor of momentum crashes. Apart from the short horizon price features consistent with the short-term reversal documented by Jegadeesh and Titman (1995) and information discreteness of Da et al. (2014), the price momentum and market model alpha at the horizons of seven to eight months are the major predictors of low expected returns. In fact the largest positive contributions of the price momentum to the predicted probability are at six and twelve months lookback horizons which is illustrated by Figure 9 that plots average partial derivatives of the predicted probabilities with respect to the price momentum (left plot) and market model alpha (right plot) against their lookback horizons. The horizons shorter than one month are aggregated into the one-month bins. The contribution of the alpha on the other hand reverts to negative values at horizons shorter than nine months. A peculiar feature is that at the six month

Table IV: Time-series asset pricing tests

Portfolio	α	$R_m - R_f$	SMB	HML	RMW	CMA	MOM	$\bar{R}^2$
Panel A: Median Sorts								
P1	−3.37	1.03	0.08	0.06	0.02	0.06	0.00	0.94
	[−4.85]	[52.44]	[2.93]	[1.90]	[0.60]	[0.92]	[0.01]	
P2	4.15	1.02	0.05	0.04	0.09	0.04	−0.06	0.92
	[4.98]	[43.30]	[1.45]	[0.81]	[1.79]	[0.59]	[−2.22]	
P2-P1	7.53	−0.01	−0.03	−0.02	0.06	−0.03	−0.06	0.02
	[6.71]	[−0.37]	[−0.47]	[−0.29]	[0.75]	[−0.28]	[−1.37]	
Panel B: Decile Sorts								
P1	−9.55	1.07	0.14	−0.06	0.15	0.14	−0.12	0.75
	[−5.25]	[22.78]	[2.10]	[−1.01]	[1.49]	[0.87]	[−1.73]	
P2	−4.99	1.02	0.08	0.09	0.13	0.08	0.01	0.86
	[−4.34]	[41.36]	[2.43]	[2.08]	[2.80]	[0.95]	[0.27]	
P3	−0.45	1.00	0.08	0.06	0.03	0.02	−0.02	0.89
	[−0.53]	[48.23]	[1.76]	[1.55]	[0.51]	[0.32]	[−0.58]	
P4	−1.44	1.07	0.05	0.11	−0.08	0.04	0.08	0.91
	[−1.53]	[48.08]	[1.77]	[2.25]	[−1.80]	[0.59]	[3.04]	
P5	−0.49	1.01	0.05	0.08	−0.11	0.04	0.05	0.90
	[−0.51]	[48.28]	[1.59]	[2.11]	[−2.28]	[0.60]	[1.55]	
P6	1.44	1.06	0.01	0.07	−0.06	0.05	0.04	0.89
	[1.28]	[32.53]	[0.10]	[1.30]	[−1.03]	[0.68]	[1.38]	
P7	1.63	1.01	0.06	0.04	−0.01	0.02	−0.00	0.89
	[1.54]	[33.74]	[1.38]	[0.68]	[−0.12]	[0.20]	[−0.12]	
P8	3.94	1.02	0.03	−0.00	0.08	0.07	−0.07	0.87
	[3.59]	[32.40]	[0.65]	[−0.03]	[1.11]	[0.85]	[−2.04]	
P9	5.05	1.00	0.08	0.03	0.15	0.03	−0.06	0.85
	[4.56]	[40.59]	[1.68]	[0.47]	[2.27]	[0.36]	[−1.78]	
P10	8.67	1.02	0.09	0.06	0.27	0.01	−0.20	0.80
	[5.56]	[29.96]	[1.60]	[0.96]	[3.97]	[0.07]	[−4.04]	
P10-P1	18.23	−0.04	−0.04	0.12	0.11	−0.13	−0.09	0.02
	[6.67]	[−0.83]	[−0.40]	[1.33]	[0.90]	[−0.69]	[−0.82]	

The table reports results of time-series regressions for excess returns on portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The test portfolios are across rows and explanatory variables are across columns. The first column reports pricing errors in percent p.a.; the regressors include the five factors of Fama and French (2015) plus 12-1 momentum. The last column shows the adjusted R^2. Panels A and B report results for median and decile sorts respectively. The numbers in brackets are the Newey and West (1987) t-statistics with number of lags according to Newey and West (1994). The sample is from January 1990 to December 2018.

Figure 8: Feature importance: gradients of predicted probabilities



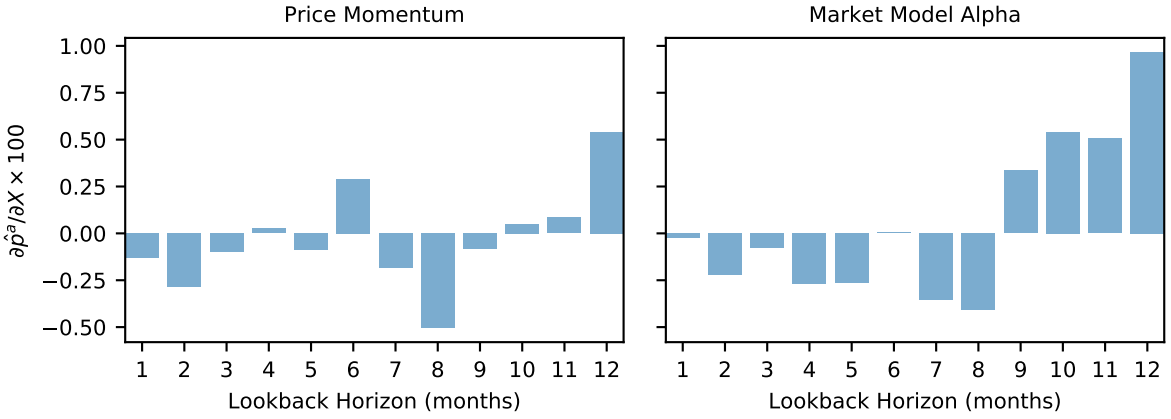
The figure shows partial derivatives of predicted probabilities of stock return being above the cross-sectional median return in the next month with respect to model's inputs. The top and bottom 10 input variables ranked by their average gradient are across the vertical axis. For a given feature the colored bars and whiskers represent respectively the interquartile and 5-95% range of all gradient evaluations. The solid black lines and dots inside each bar show median and mean value of the gradients. The sample is from January 1990 to December 2018.

horizon the gradients of both variables are much higher than those in the surrounding lookback periods.[27]

Figure 10 shows a square partition of the Hessian of ensemble's predictions with respect to input variables. The partition includes rows and columns containing 10 larges and smallest values in the Hessian. Although higher long horizon alphas, price momentum and market return predict higher expected returns the magnitudes of these variables' contributions to the predicted probability are modulated by market volatility. Keeping other things equal, the gradient of the predicted probability with respect

[27]The 12-month effect is consistent with the momentum seasonality literature (Jegadeesh (1990), Heston and Sadka (2008), Heston and Sadka (2010)). Although this discussion is beyond the scope of this paper I speculatively point out that the six-month spike might reflect the index rebalancing effect in an environment where large institutional investors follow a rule-based benchmark in spirit of Barberis and Shleifer (2003) and Barberis, Shleifer, and Wurgler (2005). For example, the MSCI Momentum Indexes, arguably the most popular benchmarks, measure price momentum as an equally weighted average of past twelve and six month returns divided by three-year standard deviation.

Figure 9: Gradients of price momentum and alpha

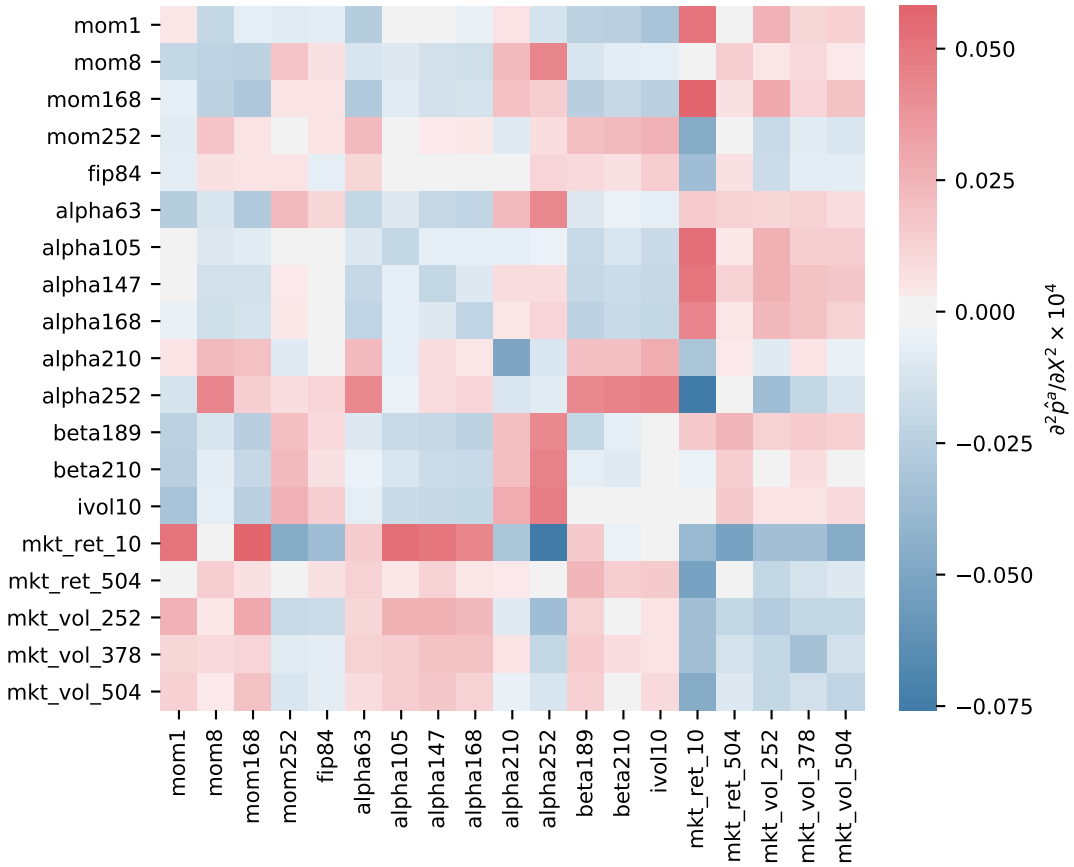


This figure accompanies Figure 8 and plots average partial derivatives of the predicted probabilities with respect to price momentum (left plot) and market model alpha (right plot) against their lookback horizons. The horizons shorter than one month are aggregated into the one-month bins.

to these variables fall when volatility rises. On the other hand the gradients of the short-horizon price variables tend to increase with volatility. The model thus captures the negative relationship between market volatility and predictive power of the standard momentum characteristics documented by Cooper et al. (2004), Barroso and Santa-Clara (2015), and Wang and Xu (2015). The model also accounts for the conditional relationship between the market beta of a stock and market state in spirit of Grundy and Martin (2001) and Daniel and Moskowitz (2016): higher betas contribute negatively to the expected return if the two-year market return is below its long-run average, that is if there is a 'panic state'.

Figure 11 repeats the analysis in Figure 8 for the long-short decile portfolio sorted on predicted probabilities. Interpretation of the gradients becomes a bit more cumbersome, for example: keeping everything else equal, a small Δ change in the 10-days market return on average increases the predicted probability of a stock in the long leg of the portfolio (or decreases the probability of a stock in the short leg) by $0.2 \times 100 \times \Delta\%$. Although many variables are rather strong unconditional predictors of returns, the long-short neural network portfolio does not simply buy one year alpha and sell short-term momentum. In fact the portfolio does not exhibit any systematic exposures to the cross-sectional features with magnitudes similar to those in the long-only case in Figure 8. Of course, on average the portfolio is tilted toward betting against the beta, information discreteness and short term idiosyncratic volatility, but there are around 25% of stock-months where these bets are reversed. The portfolio is also on average long market volatility and market return, but once again variation in these gradients is significant: in the next section I will look deeper at how time variation in these gradients allowed the neural network portfolio to avoid the 2008-2009 crash. The dominance of the market state variables together with dispersed bets in

Figure 10: Feature importance: Hessian of predicted probabilities

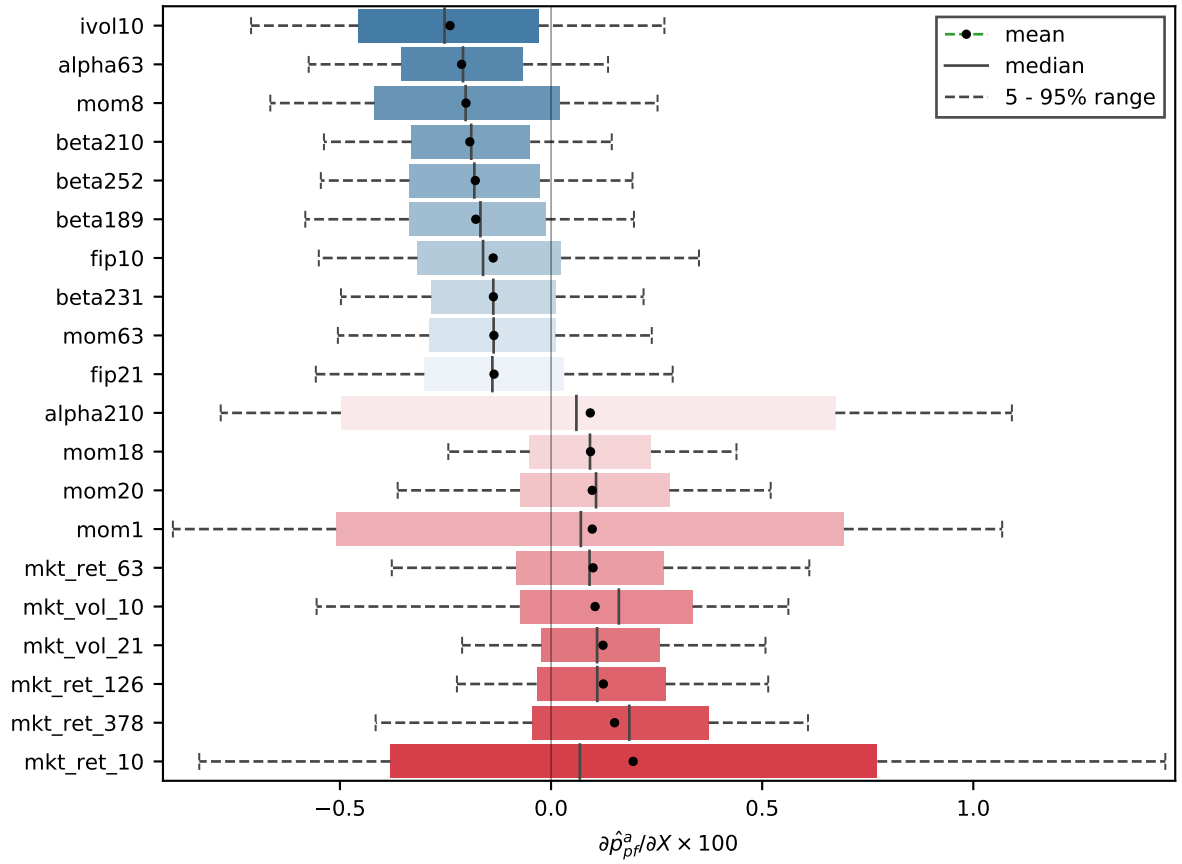


This figure accompanies Figure 8 and shows a square partition of the Hessian of ensemble's predictions with respect to input variables. The partition includes rows and columns containing 10 larges and smallest values in the Hessian.

terms of the gradients makes it hardly surprising that the classical static asset pricing factors in the time-series tests in Table IV possess virtually no explanatory power in capturing the variation in the returns on the neural network portfolios.

To demonstrate how the market state features modulate the importance of the cross-sectional characteristics I plot partitions of the Hessian of the long-short portfolio in Figure 12 for alpha (top panel), beta (bottom panel), and market state features (across horizontal axes). As market volatility rises the importance of alpha measured over longer horizons goes down and the gradients of shorter term alphas go up. The reverse applies to market returns at horizons of up to three months. For the betas, the impact of increasing market volatility is generally positive except for the shortest horizons. The beta gets higher gradients for longer horizons if the market return on similar horizons is above its long-term average. The short-term market return modulates the beta quite aggressively allowing, for example, to exploit optionality in momentum payoffs during bear markets documented extensively by Daniel and Moskowitz (2016), who show that during market downturns the simple momentum strategy mechanically ac-

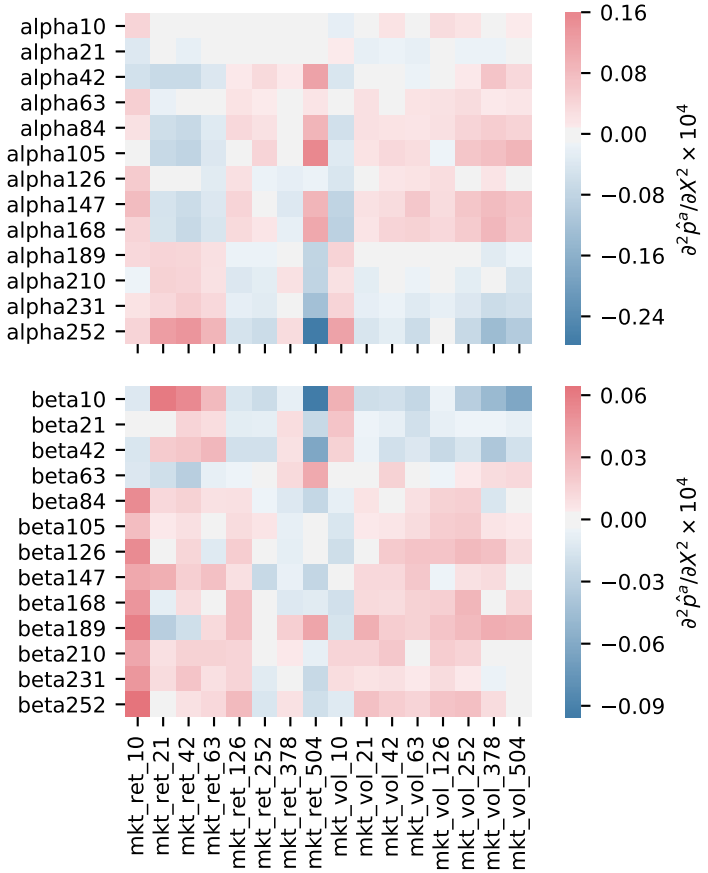
Figure 11: Feature importance: gradients of high-minus-low portfolio



The figure shows difference in the gradients of ensemble's predictions for 'high' and 'low' portfolios with respect to input variables. The portfolios are sorted on predicted probabilities of stock return being above the cross-sectional median return in the next month. Each month the 'high' and 'low' portfolios invest in stocks with predicted probabilities being in the top and bottom 10% of the cross-sectional distribution. The top and bottom 10 input variables ranked by their average gradient are across the vertical axis. For each variable the colored bars and whiskers represent respectively the interquartile and 5-95% range of all gradient evaluations for a given variable. The solid black lines and dots inside each bar show the median and mean values of the gradients. The sample is from January 1990 to December 2018.

cumulates negative market beta by buying low beta stocks and shorting high beta stocks that suffered the largest losses during the downturn. They further demonstrate that the beta becomes even more negative when the market trend reverts, making the momentum strategy to be effectively short a call option on the market and thus leading to a momentum crash. Indeed in Figure 12 as the two-year market return falls the gradients of the predicted porbabilities with respect to betas increase, especially for the very short horizons and when accompanied by decresing volatility. Furthermore, if the market rebounds, i.e. the short-horizon market returns increase, the gradients of the predictions with respect to betas increase as well, which is especially prominent for the 10-day market return.

Figure 12: Hessian of long-short portfolio: impact of market state features



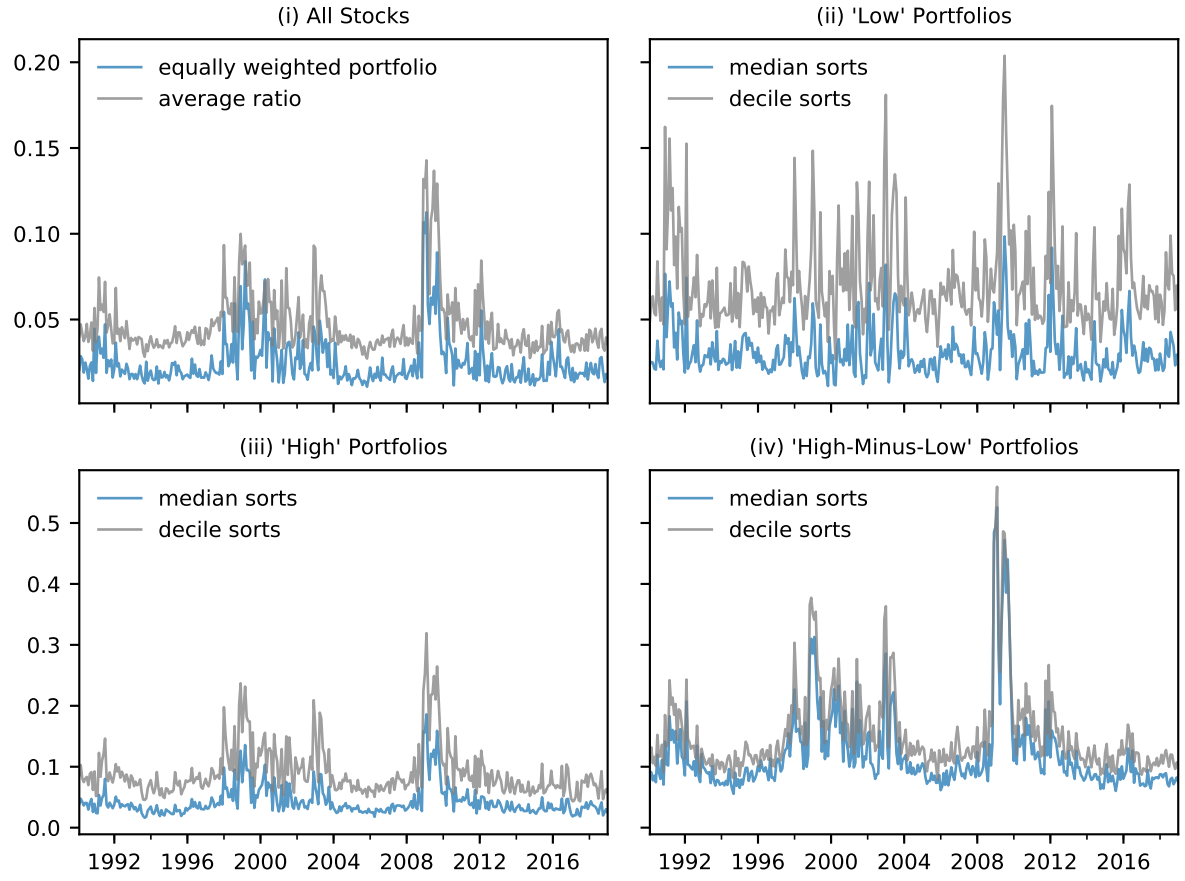
This figure accompanies Figure 11 and shows the difference in the Hessians of ensemble's predictions for 'high' and 'low' decile portfolios with respect to the input variables. The depicted partitions of the Hessian show second order derivatives with respect to market state variables for alpha and beta in the top and bottom panels respectively.

D. Aggregate Magnitude of Non-Linearities

Although the average magnitudes in the Hessians in Figures 10 and 12 are rather small they exhibit substantial variation over time. Figure 13 plots evolution of the non-linearity ratios defined in equations 9, 10 over time. From left to right and from top to bottom each plot shows (i) the average ratio over entire cross-section (in gray) and the ratio of an equally weighted portfolio (in blue); (ii, iii) the ratios of 'low' and 'high' portfolios sorted by the predicted probability of stock return being above the cross-sectional median return in the next month – for the median (in blue) and decile (in gray) sorts; (iv) the ratios of the corresponding high-minus-low portfolios (also in blue and gray). The average curvature of the decision boundary increases dramatically in distressed market states, in fact, the ratios of the Hessians to gradients for model predictions are well correlated with VIX: the correlation with the average ratio is 53.6% and ranges from 16.1% for the short median portfolio to 58.5% for the long-short decile portfolio. This supports the discussion in Section III.D.1 that time-

series variation in 9 should be primarily driven by market state variables: as I showed in the previous section the impact of these variables on predictions of the model is of considerable magnitude, especially for the long-short portfolios. Thus, when the market state variables are several standard deviations away from their long-term averages, both the absolute magnitude of the gradient and cross-sectional dispersion of the predicted probabilities increase leading to higher values of the non-linearity ratio. Interestingly, the decision boundary of the low predicted probability portfolios is on average much flatter than that of the predicted winners, which, given that the two portfolios do not differ systematically in their sensitivities to cross-sectional characteristics as shown in Figure 11, can be attributed to higher impact of market state variables and feature interaction effects in the high predicted probability portfolio.

Figure 13: Aggregate nonlinearity in ensemble predictions over time



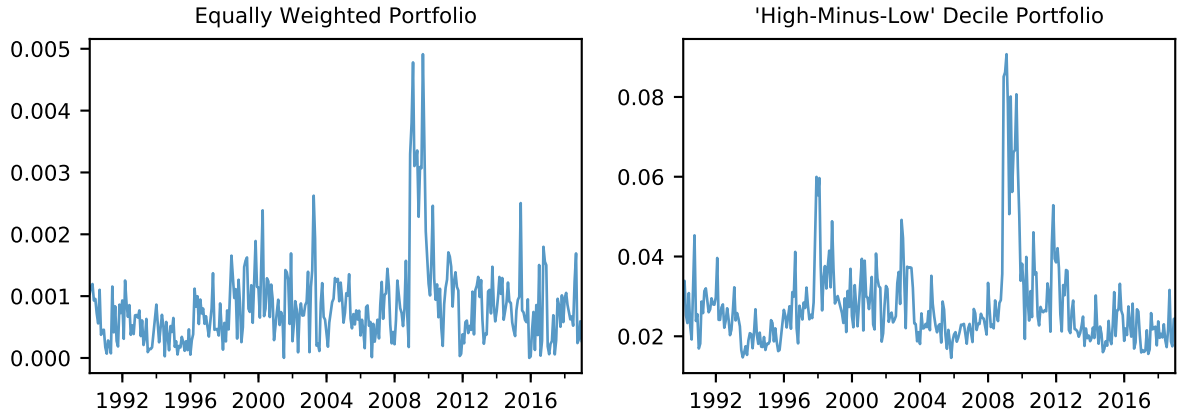
The figure plots evolution of non-linearity ratios defined in equations 9, 10 over time. From left to right and from top to bottom each plot shows (i) the average ratio over entire cross-section (in gray) and the ratio of an equally weighted portfolio (in blue); (ii, iii) the ratios of 'low' and 'high' portfolios sorted by $\hat{p}^a_{t+1}$ – the predicted probability of stock return being above the cross-sectional median return in the next month – for the median (in blue) and decile (in gray) sorts; (iv) the ratios of the corresponding high-minus-low portfolios (also in blue and gray).

In order to asses the net impact of the second order effects I modify equation 10 by taking absolute values after computing the average gradient and Hessian of the portfolio:

$$Q_t^P = \frac{\mathbf{1}^\top | \sum_{p=1}^{P} w_p \mathbf{H}_{i,t} | \mathbf{1}}{\mathbf{2}^\top | \sum_{p=1}^{P} w_p \mathbf{g}_{i,t} |}, \quad \sum_{p=1}^{P} w_p = 1 \tag{11}$$

The left panel of Figure 14 plots the ratio of an equally weighted portfolio and the right panel shows the ratio of the high-minus-low portfolio sorted by deciles of the predicted probability. On the cross-sectional level, i.e. in the equally weighted portfolio, the higher order effects by large cancel out but still surge in distressed markets. The long-short portfolio on the other hand actively exploits these effects.

Figure 14: Net impact of second order effects

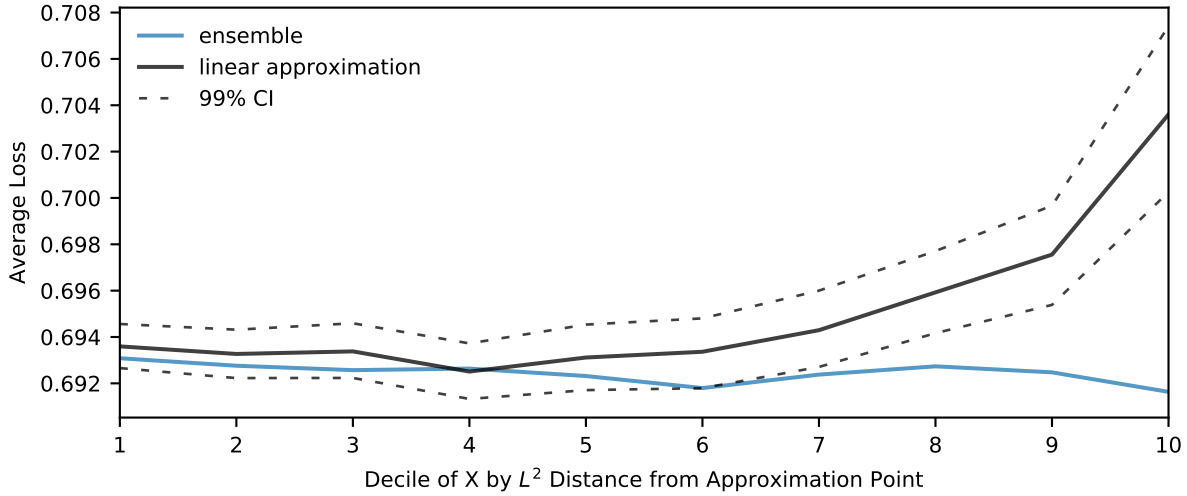


The figure shows evolution of ratios of second to first order Taylor approximation terms defined in equation 11. The left panel plots the ratio of an equally weighted portfolio and the right panel draws the ratio of the high-minus-low portfolio sorted by deciles of $\hat{p}^a_{t+1}$ – the predicted probability of stock return being above the cross-sectional median return in the next month.

Of course, although they demonstrate the impact of the market state variables in a simple way, the Hessian-gradient ratios in the analysis above is but a crude approximation of the true shape of the decision boundary. I follow the procedure outlined in Section III.D.2 to emphasize the effect of dispersion in the cross-sectional characteristics on the predictions of the model. Figure 15 plots losses of the ensemble and first order Taylor approximation of its predictions by L^2 distance from approximation point in the feature space. Each month the approximation point is set at the median predicted probability $p^a_{m,t+1}$, the rest of the cross-section of predictions is then sorted into deciles by $||X_{i,t} - X_{m,t}||_2$ – the L^2 distance between the features vector of a stock and that of the median prediction, such that the first and tenth deciles correspond to observations closest and farthest from the median prediction in the Euclidean space. For each decile across the horizontal axis, the solid blue line shows the average ensemble loss in the corresponding bin over all months. Similarly, the solid black line plots the loss of the first order approximation of ensemble's predictions around $p^a_{m,t+1}$. The

dashed lines draw 99% bootstrapped confidence intervals around the approximation loss. Each bin contains around 17000 stock-month observations. Note that the goal of this exercise is not to show that the non-linear model is better than its linear approximation – the latter can at most be as good as the former – but to ask at which point does this non-linearity start to matter while controlling for direct impact of the market state features. The decision boundary is rather flat for the half of observations whose feature vectors are the closest to the feature vector of the median prediction. Past this point, the more dissimilar cross-sectional characteristics become the higher becomes the difference in losses between the linear approximation and the model. This means that high and low predicted probability regions of the decision boundary, where the linear approximation, in general, over- and under-predicts the probability, correspond to the stocks whose cross-sectional characteristics differ in aggregate from those of the stock with the median predicted probability. Furthermore, the difference in losses rises as this difference increases.

Figure 15: Cross-sectional non-linearity in ensemble predictions



The figure plots losses of the ensemble and first order Taylor approximation of its predictions by L^2 distance from approximation point in the feature space. Each month the approximation point is set at the median predicted probability $p^a_{m,t+1}$, the rest of the cross-section of predictions is then sorted into deciles by $||X_{i,t} - X_{m,t}||_2$ – the L^2 distance between the features vector of a stock and that of the median prediction, such that the first and tenth deciles correspond to observations closest and farthest from the median prediction in the Euclidean space. For each decile across the x-axis, the solid blue line shows the average ensemble loss in the corresponding bin over all months. Similarly, the solid black line plots the loss of the first order approximation of ensemble's predictions around $p^a_{m,t+1}$. The dashed lines draw 99% bootstrapped confidence intervals around the approximation loss. Each bin contains around 17000 stock-month observations. The sample is from January 1990 to December 2018.

In fact, the cross-sectional features and interactions thereof are the main drivers of the returns on the ensemble portfolios. I show it by setting the market state features to their unconditional means – zeros – on the test sample. The performance of the neural network portfolios in terms of return deteriorates insignificantly in terms of

returns and standard deviations. For example, long-short decile portfolio earns, on average 15% p.a. with Sharpe ratio of 1.22 versus 17.7% and 1.34 for the full model. An important feature, however that omission of market state variables leads to negative skewness in the returns on the portfolio: the portfolio experiences drawdowns coinciding with momentum crashes, albeit much less severe than those of the classical momentum – the maximum drawdown of the long-short ensemble portfolio with muted market state features is 18.5%. Once the market state features are switched off, the returns on the neural network portfolios also become more correlated with returns on the 12-1 momentum portfolio in asset pricing tests: the long-short portfolios for both median and decile sorts become positively and significantly loaded on the momentum factor, the R-squared in time-series asset pricing tests with FF-5 model plus momentum as factors rises to 17 and 13% for the median and decile sorts respectively. The alphas of the ensemble portfolios nevertheless remain positive and statistically significant at any conventional level. I report the descriptive statistics and results of asset pricing tests for the restricted version of the model in Tables D.II and D.III in Appendix.

Finally, the restricted version of the model spans the returns on the 12-1 momentum strategy in a univariate regression:[28]

$$\widehat{MOM}_{1,t} = \underset{[-0.27]}{-1.04\%\ p.a.} + \underset{[4.42]}{0.49} HML_{NN,t}, \quad \bar{R}^2 = 0.13 \tag{12}$$

$$\widehat{MOM}_{2,t} = \underset{[-0.05]}{-0.25\%\ p.a.} + \underset{[4.70]}{0.71} HML_{NN,t}, \quad \bar{R}^2 = 0.10, \tag{13}$$

The dependent variables in each equation are the momentum factor from the Kenneth French's web page and a 12-1 strategy with decile sorts computed on my dataset, and the independent variable is the return on the decile ensemble portfolio with market state features set to their unconditional means. The alphas are in percent p.a. and the numbers in brackets are t-statistics. Together with the market risk premium the ensemble portfolio can also fully price the individual portfolios sorted on past 12-1 returns.

[28] There is actually not much to span: from 1990 to 2018 the momentum factor from Kenneth French's data library and 12-1 decile momentum strategy earned on average 6.3 and 10.3% p.a. with annualized Sharpe ratios of around 0.38 and t-statistics below 2.

E. Revisiting the 2009 Momentum Crash

In 2009 momentum experienced one of the most severe crashes in history loosing more than half of its value from March to August.[29] Over the same period the long-short ensemble portfolio earned 30 percent return. In the restricted model specification the ensemble portfolios can also span the standard momentum in time-series asset pricing tests what makes it a more general model of expected returns. The fact that the neural network model encompasses the standard momentum makes it a natural laboratory to better grasp the risk-return profile of the latter.

I begin with analyzing what are the fundamental differences in the gradients of predicted probabilities between the long-short decile ensemble and 12-1 momentum portfolios over the 2008-2009 period. To make the wording less cumbersome I define the gradient of an equally weighted portfolio with respect to an input as the average gradient of its constituents' predicted probabilities, with respect to this input, to be above the median cross-sectional return in the next month.

Table V reports four metrics for the ensemble (Panel A) and momentum portfolios (Panel B) over the 2008-2009 period: maximum decrease and increase in the portfolio gradient (columns 1-2 and 3-4); lowest and highest average gradients (columns 5-6 and 7-8). For each metric Table V reports the five most extreme values. In terms of the model's predictions, keeping everything else equal, on average the ensemble bets against the long horizon beta and is long volatility, while the momentum portfolio is short volatility and short-term market return and is long intermediate horizon momentum and alpha along with the long horizon market return. To paraphrase it, the expected return (under the model) of the ensemble portfolio loads positively on the market volatility and short-term market return, while the expected return of the momentum portfolio loads negatively on the volatility and positively on the standard cross-sectional momentum characteristics and long-term market return. With regard to the extreme drops and increases in their gradients both portfolios experience the most dramatic changes with respect to both short- and long-term market trends. The question is when?

The top two panels of Figure 16 plot gradients of the long-short ensemble (in red) and momentum (in blue) portfolios with respect to the two-year (left) and 10-day (right) market return over the 2008-2009 period. The bottom panels show realizations of these variables. The optionality in the payoff of the 12-1 momentum strategy documented by Daniel and Moskowitz (2016) becomes immediately obvious: up until the middle of 2009 the momentum strategy holds stocks whose expected return (in terms of the

[29] See Daniel and Moskowitz (2016) for an in-depth review of momentum crashes.

Table V: Extreme gradients of ensemble and momentum portoflios in 2008-2009

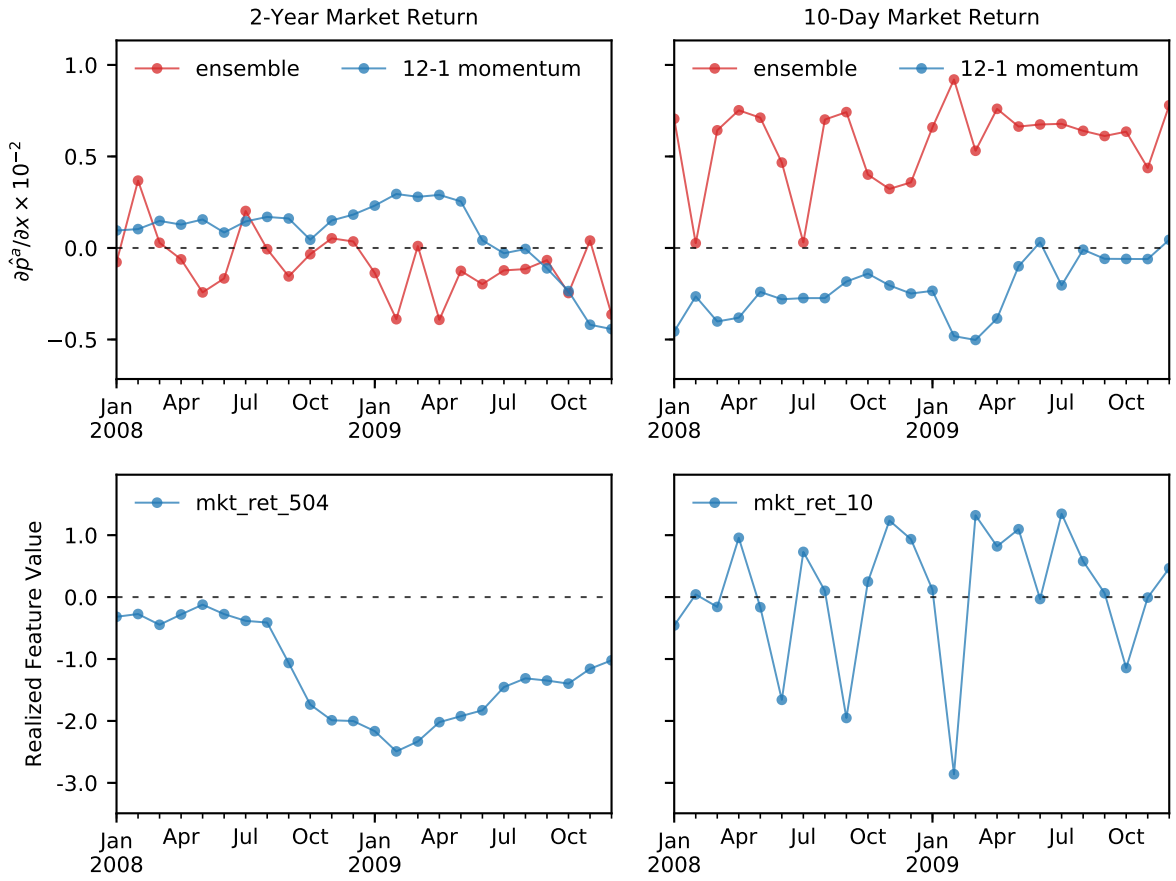
Max Drop		Max Increase		Lowest Mean		Highest Mean	
X	$\nabla \times 10^2$	X	$\nabla \times 10^2$	X	$\nabla \times 10^2$	X	$\nabla \times 10^2$
Panel A: Ensemble Portfolio							
mkt_ret_504	-0.760	mkt_ret_10	0.894	ivol10	-0.280	mkt_ret_10	0.577
mkt_ret_10	-0.722	mkt_ret_21	0.543	alpha63	-0.220	mkt_vol_252	0.186
mkt_ret_252	-0.445	mkt_ret_42	0.502	beta189	-0.203	mkt_vol_42	0.178
alpha252	-0.443	mkt_vol_63	0.492	beta210	-0.189	mkt_vol_126	0.155
mkt_vol_63	-0.425	mkt_ret_504	0.445	beta252	-0.178	mkt_vol_21	0.131
Panel B: 12-1 Momentum Portfolio							
mkt_ret_504	-0.738	mkt_ret_21	0.592	mkt_ret_10	-0.224	mkt_vol_10	0.187
alpha252	-0.513	mkt_ret_10	0.547	mkt_vol_63	-0.211	mom1	0.173
mkt_ret_10	-0.362	mkt_ret_42	0.507	mkt_vol_252	-0.147	mkt_ret_378	0.172
ivol10	-0.328	mkt_vol_63	0.371	alpha252	-0.144	mom168	0.170
mkt_vol_504	-0.317	alpha105	0.351	mkt_vol_378	-0.119	alpha168	0.153

The table reports gradients of long-short equally weighted ensemble and and 12-1 momentum decile portfolios with respect to input variables for the 2008-2009 period. The gradient of a portfolio with respect to an input is the average gradient of its constituents' predicted probabilities, with respect to this input, to be above the median cross-sectional return in the next month. Thus a gradient of ∇ with respect to feature X means that a small Δ increase in X, on average, changes the difference in the predicted probabilities of stocks in the long and short legs of the portfolio to be above the cross-sectional median in the next period by $\Delta \times \nabla \times 100\%$. Columns 1-2 and 3-4 in Panel A report five input variables with respect to which the ensemble portfolio experienced the sharpest drops and increases in its gradient respectively. Columns 5-6 and 7-8 report the input variables with respect to which the portfolio maintained lowest and highest gradients on average. Panel B repeats the analysis for the 12-1 momentum portfolio.

predicted probability) loads positively and negatively on the long and short market return respectively, and more so on the latter in absolute terms. The ensemble portfolio on the other hand actively exploits both trends not only by reducing its exposure to the two-year market return in 2008 and early 2009 but also by almost halving the gradient with respect to the short-term market return during the collapse of the market in the fall of 2008.

Figure 17 further shows exposures of the momentum and ensemble portfolios to the cross-sectional characteristics over the 2008-2009 period. The top two panels of the figure plot gradients of the long-short ensemble (in red) and momentum (in blue) portfolios with respect to the one-year alpha (left) and one-year beta (right). The bottom panels show realizations of these variables on the portfolio level also in red and blue. The realized value of a cross-sectional input variable at time t in a portfolio is the average value of this variable in its holdings at t. Since the input features are standardized with respect to the cross-section, positive (negative) values in the bottom panel mean that the portfolio is on average long (short) the corresponding characteristic. By investing in the past 12-month winners the momentum portfolio also maintains stable

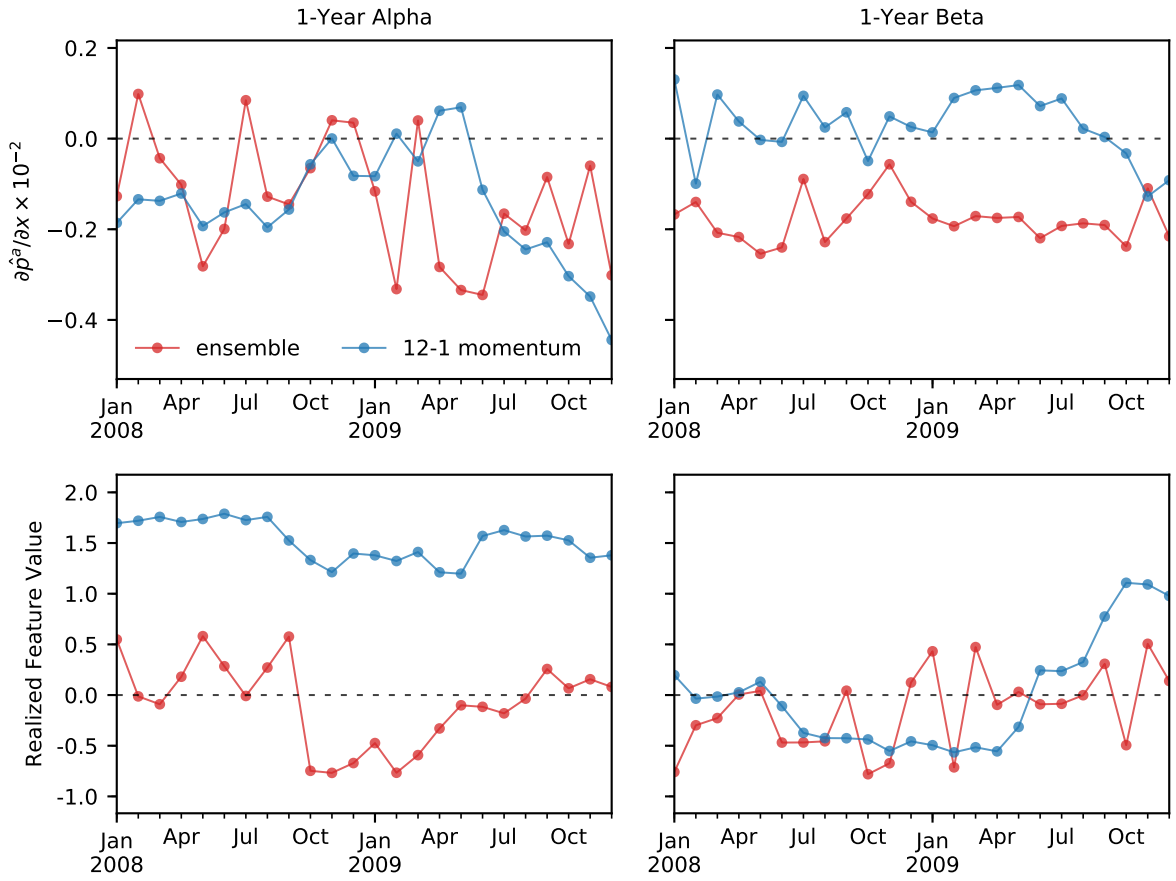
Figure 16: Gradients of ensemble and momentum portfolios with respect to market return in 2008-2009



The top two panels of the figure plot gradients of the long-short ensemble (in red) and momentum (in blue) portfolios with respect to 2-year (left) and 10-day (right) market returns over the 2008-2009 period. The bottom panels show realizations of these variables. The gradient of a portfolio with respect to an input is the average gradient of its constituents' predicted probabilities, with respect to this input, to be above the median cross-sectional return in the next month.

exposure to the one-year alpha, as it is strongly correlated with the one-year return, even though under the model, keeping everything else equal, reducing the exposure to this variable would lead to higher expected return. Simultaneously the ensemble portfolio aggressively reverts its exposure to this feature in the fall of 2008. With regard to the beta the momentum mechanistically accumulates negative exposure to the market risk well into 2009, while the expected return starts to increase in this variable from February 2009 onwards. The ensemble portfolio on the other hand by large reverts its bet against the beta in March 2009. Note that for the ensemble portfolio the realized allocation in both variables shifts toward zero by the middle of 2009 despite their negative gradients, as other variables with larger gradients like the market state features dominate the predictions.

Figure 17: Gradients of ensemble and momentum portfolios with respect to cross-sectional features in 2008-2009



The top two panels of the figure plot gradients of the long-short ensemble (in red) and momentum (in blue) portfolios with respect to one-year alpha (left) and one-year beta (right) over the 2008-2009 period. The bottom panels show realizations of these variables on portfolio level also in red and blue for the ensemble and momentum portfolios respectively. The realized value of a cross-sectional input variable at time t in a portfolio is the average value of this variable in its holdings at t. Since the input features are standardized with respect to the cross-section positive (negative) values in the bottom panel mean that a portfolio is on average long (short) the corresponding characteristic. The gradient of a portfolio with respect to an input is the average gradient of its constituents' predicted probabilities, with respect to this input, to be above the median cross-sectional return in the next month.

F. Robustness Checks: all CRSP Stocks and other ML Portfolios

In order to ensure that the results are comparable to other studies and are not limited to the subsample of the largest stocks, I construct a set of features for all CRSP stocks using the procedure outlined in Section II and use the ensemble of neural networks to predict returns on the test sample without any additional training. The number of stock-month observations in the test set increases more than sevenfold from over 170,000 to over 1.2 million. I then repeat the asset pricing analysis for both equally- and value-weighted portfolios of stocks from the entire CRSP universe. The main highlights are as follows: (i) the return on the long-short decile ensemble portfolios increases to over 22 percent p.a. for both equally- and value-weighted portfolios, with

t-statistic and Sharpe ratio for the former rising to over 10.5 and 2 respectively, (ii) the results for the value-weighted portfolios are overall very similar to those reported so far for the equally-weighted portfolios on the large stocks sample; (iii) while similar in performance (in terms of Sharpe ratios) to the neural network portfolios of Gu et al. (2018) and Chen et al. (2019) my portfolios have about half of their maximum drawdown and worst single month loss. These results are available in Internet Appendix.

I further investigate whether returns on other machine learning portfolios from the literature can explain the returns on the ensemble portfolios. Table VI reports results of time-series regressions for excess returns on portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The test portfolios are across rows and explanatory variables include five Fama and French (2015) factors, momentum and the stochastic discount factor portfolio of Chen et al. (2019). Each triplet of columns reports pricing error in percent p.a., regression coefficient on the SDF portfolio and adjusted R^2 of the regression. The first triplet reports estimates for the large cap sample with test assets being equally-weighted portfolios constructed in the universe of five hundred largest stocks. The second and third triplets repeat the exercise for the entire CRSP sample with equal- and value-weighted test portfolios respectively. Panels A and B report results for median and decile sorts respectively. The numbers in brackets are the Newey and West (1987) t-statistics with number of lags according to Newey and West (1994). The sample is from January 1992 to December 2016. For the equally-weighted portfolios (middle triplet), the SDF factor portfolio's loadings increase and gain statistical significance from low to high test portfolios and explain more time-series variation in the returns on the test portfolios that the rest of the factors combined (see Table IV). The proportion of the explained variance is, however, still tiny and the SDF factor cannot capture the ensemble portfolios' alphas. Furthermore the seven-factor model performs even worse on the value-weighted and large cap portfolios (third and first triplets of columns), explaining next to nothing in the time-series variation in the returns on the long-short ensemble portfolios. I repeat this exercise with similar results using long-short equally- and value-weighted decile portfolios of Chen et al. (2019) as the seventh factor instead of their SDF, these results are available in Internet Appendix.

Overall, the uncorrelatedness of the returns on the ensemble portfolios with other risk factors allows to substantially expand the efficient frontier of an investor while simultaneously reducing the downside risk: simple equally-weighted combination of the long-short value-weighted decile portfolio of Chen et al. (2019) (mean return 26.3% p.a., SR 1.19 p.a., MDD 36.6%) with the similar ensemble portfolio (mean return 22.6% p.a., SR 1.39 p.a., MDD 26.1%) yields an out-of-sample mean return of 24.3% p.a., Sharpe ratio of 1.63 p.a. and maximum drawdown of 23.8%.

Table VI: Time-series asset pricing tests: FF-5 + MOM + Chen et al. (2019) SDF

	Top 500 MV			All CRSP EW			All CRSP VW		
Portfolio	α	SDF	$\bar{R}^2$	α	SDF	$\bar{R}^2$	α	SDF	$\bar{R}^2$
Panel A: Median Sorts									
P1	−2.70	−0.09	0.94	−3.98	0.08	0.97	−4.00	0.03	0.94
	[−2.37]	[−0.96]		[−3.60]	[0.88]		[−4.38]	[0.46]	
P2	3.97	0.01	0.92	2.60	0.30	0.95	2.88	0.09	0.94
	[3.13]	[0.16]		[2.33]	[3.92]		[2.90]	[0.83]	
P2-P1	6.67	0.10	0.03	6.58	0.22	0.06	6.88	0.06	0.03
	[3.61]	[0.72]		[3.92]	[1.61]		[3.82]	[0.35]	
Panel B: Decile Sorts									
P1	−10.53	0.09	0.74	−10.41	−0.00	0.86	−11.69	−0.08	0.67
	[−3.44]	[0.36]		[−4.38]	[−0.02]		[−3.60]	[−0.30]	
P2	−5.25	−0.03	0.85	−5.45	0.07	0.93	−7.06	0.18	0.78
	[−3.18]	[−0.21]		[−3.57]	[0.54]		[−3.60]	[1.11]	
P3	−0.50	0.04	0.88	−3.05	0.08	0.96	−4.32	0.18	0.87
	[−0.34]	[0.29]		[−3.06]	[1.16]		[−3.13]	[1.71]	
P4	0.35	−0.22	0.91	−0.90	0.15	0.96	−2.28	0.07	0.86
	[0.31]	[−1.96]		[−0.96]	[1.78]		[−1.44]	[0.49]	
P5	2.36	−0.30	0.90	−0.09	0.10	0.97	−0.37	−0.01	0.90
	[1.92]	[−3.67]		[−0.12]	[1.72]		[−0.25]	[−0.10]	
P6	1.47	0.01	0.88	1.36	0.15	0.96	3.07	−0.07	0.88
	[0.93]	[0.03]		[1.31]	[1.71]		[2.38]	[−0.47]	
P7	2.23	−0.16	0.89	2.66	0.15	0.95	2.55	0.05	0.89
	[1.40]	[−1.60]		[2.13]	[1.68]		[1.89]	[0.38]	
P8	4.63	−0.03	0.85	1.50	0.27	0.94	0.33	0.20	0.86
	[2.72]	[−0.29]		[1.19]	[2.97]		[0.19]	[1.05]	
P9	4.28	0.06	0.85	2.16	0.35	0.92	0.60	0.28	0.82
	[2.72]	[0.42]		[1.40]	[3.24]		[0.37]	[1.89]	
P10	7.22	0.19	0.80	5.30	0.59	0.89	9.84	0.03	0.74
	[3.22]	[0.90]		[3.09]	[4.88]		[3.29]	[0.11]	
P10-P1	17.75	0.10	0.03	15.72	0.60	0.05	21.52	0.11	−0.01
	[4.43]	[0.31]		[4.43]	[2.13]		[4.33]	[0.25]	

The table reports results of time-series regressions for excess returns on portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The test portfolios are across rows and explanatory variables include five Fama and French (2015) factors, momentum and SDF portfolio of Chen et al. (2019). Each triplet of columns reports pricing error in percent p.a., regression coefficient on the SDF portoflio and adjusted R^2 of the regression. The first triplet reports estimates for the large cap sample with test assets being equally-weighted portfolios constructed in the universe of five hundred largest stocks. The second and third triplets repeat the exercise for the entire CRSP sample with equal- and value-weighted test portfolios respectively. Panels A and B report results for median and decile sorts respectively. The numbers in brackets are the Newey and West (1987) t-statistics with number of lags according to Newey and West (1994). The sample is from January 1992 to December 2016.

V. Conclusion

I investigate predictive power of a broad set of price-based variables over various time horizons in a deep learning framework. I document rich non-linear structure in impact of these variables on expected returns in the US equity market. The magnitude and sign of the impact exhibit substantial time variation and are modulated by interaction effects among features. The degree of non-linearity in expected returns also varies substantially over time and is at its highest in distressed markets. I show that investment strategies built on the out-of-sample predictions of the deep learning model actively exploit the non-linearities and interaction effects, generating high and statistically significant returns with a robust risk profile and their performance virtually uncorrelated with the established risk factors including momentum and other machine learning portoflios. I further demonstrate how to exploit differentiability of neural networks' outputs with respect to input variables to study directional effects of the variables on models' predictions and show that this analysis allows to relate the predictions to existing intuitions about momentum and its risks, thus increasing transparency of the results. I find that that incorporating insights from the literature on time-varying, market state-dependent momentum risks and momentum crashes helps to improve out-of-sample performance of neural network portfolios, especially with respect to the downside risk. Lastly, I make a case for adoption of automated hyperparameter optimization techniques as an important component of disciplined research in financial machine learning.

This research can be continued in several directions. First, although capable of grasping economically meaningful interaction effects, the architecture I employ is rather simplistic: the model remains quasi-linear in inputs until the last layer of the network. Research aiming to investigate the interaction effects in greater depth might opt for architectures specifically designed for this task (Tsang et al. (2017), Cui et al. (2019)). Second, statistical inference for neural network inputs and predictions is an active area of research. Bootstrap methods (Baxt and White (1995), White and Racine (2001)) are computationally costly, because they require fitting the model several times, and parametric approaches (Horel and Giesecke (2019)) are applicable for certain types of architectures only. Approximate Bayesian inference (Gal and Ghahramani (2016), Cui et al. (2019)), on the other hand, is a non-parametric approach that can be applied to all networks that use dropout as regularization.

References

Ang, Andrew, Robert J Hodrick, Yuhang Xing, and Xiaoyan Zhang, 2006, The cross-section of volatility and expected returns, *The Journal of Finance* 61, 259–299.

Ang, Andrew, Robert J Hodrick, Yuhang Xing, and Xiaoyan Zhang, 2009, High idiosyncratic volatility and low returns: International and further us evidence, *Journal of Financial Economics* 91, 1–23.

Arena, Matteo P, K Stephen Haggard, and Xuemin Yan, 2008, Price momentum and idiosyncratic volatility, *Financial Review* 43, 159–190.

Asness, Clifford S, 1995, The power of past stock returns to explain future stock returns, *Available at SSRN 2865769* .

Asness, Clifford S, Tobias J Moskowitz, and Lasse Heje Pedersen, 2013, Value and momentum everywhere, *The Journal of Finance* 68, 929–985.

Bach, Sebastian, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek, 2015, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, *PloS one* 10, e0130140.

Bali, Turan G, and Nusret Cakici, 2008, Idiosyncratic volatility and the cross section of expected returns, *Journal of Financial and Quantitative Analysis* 43, 29–58.

Barberis, Nicholas, and Andrei Shleifer, 2003, Style investing, *Journal of financial Economics* 68, 161–199.

Barberis, Nicholas, Andrei Shleifer, and Robert Vishny, 1998, A model of investor sentiment, *Journal of financial economics* 49, 307–343.

Barberis, Nicholas, Andrei Shleifer, and Jeffrey Wurgler, 2005, Comovement, *Journal of financial economics* 75, 283–317.

Barroso, Pedro, and Pedro Santa-Clara, 2015, Momentum has its moments, *Journal of Financial Economics* 116, 111–120.

Baxt, William G, and Halbert White, 1995, Bootstrapping confidence intervals for clinical input variable effects in a network trained to identify the presence of acute myocardial infarction, *Neural Computation* 7, 624–638.

Bergstra, James, Daniel Yamins, and David Daniel Cox, 2013, Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures, *Journal of Machine Learning Research* 28.

Bergstra, James S, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl, 2011, Algorithms for hyper-parameter optimization, in *Advances in neural information processing systems*, 2546–2554.

Bishop, Christopher M, 1995, Regularization and complexity control in feed-forward networks, in *Proceedings International Conference on Artificial Neural Networks ICANN*, volume 95, 141–148.

Blitz, David, Matthias X Hanauer, and Milan Vidojevic, 2017, The idiosyncratic momentum anomaly .

Blitz, David, Joop Huij, and Martin Martens, 2011, Residual momentum, *Journal of Empirical Finance* 18, 506–521.

Brav, Alon, James B Heaton, and Si Li, 2009, The limits of the limits of arbitrage, *Review of Finance* 14, 157–187.

Caruana, Rich, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes, 2004, Ensemble selection from libraries of models, in *Proceedings of the twenty-first international conference on Machine learning*, 18, ACM.

Chen, Luyang, Markus Pelger, and Jason Zhu, 2019, Deep learning in asset pricing, *Available at SSRN 3350138* .

Cooper, Michael J, Roberto C Gutierrez Jr, and Allaudeen Hameed, 2004, Market states and momentum, *The Journal of Finance* 59, 1345–1365.

Cui, Tianyu, Pekka Marttinen, and Samuel Kaski, 2019, Recovering pairwise interactions using neural networks, *arXiv preprint arXiv:1901.08361* .

Da, Zhi, Umit G Gurun, and Mitch Warachka, 2014, Frog in the pan: Continuous information and momentum, *The review of financial studies* 27, 2171–2218.

Daniel, Kent, and Tobias J Moskowitz, 2016, Momentum crashes, *Journal of Financial Economics* 122, 221–247.

Falkner, Stefan, Aaron Klein, and Frank Hutter, 2018, Bohb: Robust and efficient hyperparameter optimization at scale, *arXiv preprint arXiv:1807.01774* .

Fama, Eugene F, and Kenneth R French, 1993, Common risk factors in the returns on stocks and bonds, *Journal of financial economics* 33, 3–56.

Fama, Eugene F, and Kenneth R French, 2015, A five-factor asset pricing model, *Journal of financial economics* 116, 1–22.

Feng, Guanhao, Stefano Giglio, and Dacheng Xiu, 2017, Taming the factor zoo, *Fama-Miller Working Paper* 24070.

Fung, William, and David A Hsieh, 2001, The risk in hedge fund strategies: Theory and evidence from trend followers, *The review of financial studies* 14, 313–341.

Gal, Yarin, and Zoubin Ghahramani, 2016, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in *international conference on machine learning*, 1050–1059.

Geczy, Christopher C, and Mikhail Samonov, 2016, Two centuries of price-return momentum, *Financial Analysts Journal* 72, 32–56.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio, 2011, Deep sparse rectifier neural networks, in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 315–323.

Goetzmann, William N, and Simon Huang, 2018, Momentum in imperial russia, *Journal of Financial Economics* 130, 579–591.

Grundy, Bruce D, and J Spencer Martin Martin, 2001, Understanding the nature of the risks and the source of the rewards to momentum investing, *The Review of Financial Studies* 14, 29–78.

Gu, Shihao, Bryan Kelly, and Dacheng Xiu, 2018, Empirical asset pricing via machine learning, Technical report, National Bureau of Economic Research.

Gutierrez Jr, Roberto C, and Christo A Prinsky, 2007, Momentum, reversal, and the trading behaviors of institutions, *Journal of Financial Markets* 10, 48–75.

Hechtlinger, Yotam, 2016, Interpretation of prediction models using the input gradient, *arXiv preprint arXiv:1611.07634* .

Heston, Steven L, and Ronnie Sadka, 2008, Seasonality in the cross-section of stock returns, *Journal of Financial Economics* 87, 418–445.

Heston, Steven L, and Ronnie Sadka, 2010, Seasonality in the cross section of stock returns: the international evidence, *Journal of Financial and Quantitative Analysis* 45, 1133–1160.

Hong, Harrison, and Jeremy C Stein, 1999, A unified theory of underreaction, momentum trading, and overreaction in asset markets, *The Journal of finance* 54, 2143–2184.

Horel, Enguerrand, and Kay Giesecke, 2019, Towards explainable ai: Significance tests for neural networks, *arXiv preprint arXiv:1902.06021* .

Hou, Kewei, Chen Xue, and Lu Zhang, 2017, Replicating anomalies, Technical report, National Bureau of Economic Research.

Hühn, Hannah, and Hendrik Scholz, 2018, Alpha momentum and price momentum, *International Journal of Financial Studies* 6, 49.

Israel, Ronen, and Tobias J Moskowitz, 2013, The role of shorting, firm size, and time on market anomalies, *Journal of Financial Economics* 108, 275–301.

Jegadeesh, Narasimhan, 1990, Evidence of predictable behavior of security returns, *The Journal of finance* 45, 881–898.

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

Jegadeesh, Narasimhan, and Sheridan Titman, 1995, Short-horizon return reversals and the bid-ask spread, *Journal of Financial Intermediation* 4, 116–132.

Kelly, Bryan T, Seth Pruitt, and Yinan Su, 2017, Some characteristics are risk exposures, and the rest are irrelevant, *Unpublished Manuscript, University of Chicago* .

Kingma, Diederik P, and Jimmy Ba, 2014, Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* .

McLean, R David, 2010, Idiosyncratic risk, long-term reversal, and momentum, *Journal of Financial and Quantitative Analysis* 45, 883–906.

Menkhoff, Lukas, Lucio Sarno, Maik Schmeling, and Andreas Schrimpf, 2012, Currency momentum strategies, *Journal of Financial Economics* 106, 660–684.

Messmer, Marcial, 2017, Deep learning and the cross-section of expected returns, *Available at SSRN 3081555* .

Nair, Vinod, and Geoffrey E Hinton, 2010, Rectified linear units improve restricted boltzmann machines, in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

Newey, Whitney K, and Kenneth D West, 1987, A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix, *Econometrica* 55, 703–708.

Newey, Whitney K, and Kenneth D West, 1994, Automatic lag selection in covariance matrix estimation, *The Review of Economic Studies* 61, 631–653.

Novy-Marx, Robert, 2012, Is momentum really momentum?, *Journal of Financial Economics* 103, 429–453.

Shleifer, Andrei, and Robert W Vishny, 1997, The limits of arbitrage, *The Journal of finance* 52, 35–55.

Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje, 2017, Learning important features through propagating activation differences, in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 3145–3153, JMLR. org.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman, 2013, Deep inside convolutional networks: Visualising image classification models and saliency maps, *arXiv preprint arXiv:1312.6034* .

Smith, Samuel L, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le, 2017, Don't decay the learning rate, increase the batch size, *arXiv preprint arXiv:1711.00489* .

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

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan, 2017, Axiomatic attribution for deep networks, in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 3319–3328, JMLR. org.

Takeuchi, Lawrence, and Yu-Ying Albert Lee, 2013, Applying deep learning to enhance momentum trading strategies in stocks, in *Technical Report* (Stanford University).

Tsang, Michael, Dehua Cheng, and Yan Liu, 2017, Detecting statistical interactions from neural network weights, *arXiv preprint arXiv:1705.04977* .

Wang, Kevin Q, and Jianguo Xu, 2015, Market volatility and momentum, *Journal of Empirical Finance* 30, 79–91.

White, Halbert, 1989, Some asymptotic results for learning in single hidden-layer feedforward network models, *Journal of the American Statistical Association* 84, 1003–1013.

White, Halbert, and Jeffrey Racine, 2001, Statistical inference, the bootstrap, and neural-network modeling with application to foreign exchange rates, *IEEE Transactions on Neural Networks* 12, 657–673.

Appendix A Tree-structured Parzen Estimator

Denote the hyperparameter configuration space as Θ and the validation set performance of a model as y. The performance can be represented as a function of the model's hyperparameters $y = f(\theta)$, $f : \Theta \to \mathbb{R}$, $\theta \in \Theta$. The optimal set of hyperparameters is then given by $\theta^* \in \arg\min_{\theta\in\Theta} f(\theta)$. Direct evaluation of f is, however, costly and often infeasible. Bayesian optimization addresses this problem by replacing the true objective with a probabilistic surrogate model $p(y|D_i)$ where $D_i = \{(\theta_k, y_k)\}_{k=1}^{i}$ is a sequence of accumulated evaluation observations, and maximizing an acquisition function $I : \Theta \to \mathbb{R}$. The set of hyperparameters maximizing the acquisition function $\theta_{i+1} = \arg\max_{\theta\in\Theta} I(\theta)$ is then used to evaluate the objective function, $y_{i+1} = f(\theta_{i+1})$. The pair (θ_{i+1}, y_{i+1}) is then added to the observation history: $D_{i+1} = D_i \cup (\theta_{i+1}, y_{i+1})$. The Tree-structured Parzen Estimator (TPE) of Bergstra et al. (2011) uses expected improvement as the acquisition function:

$$\begin{aligned}\mathbb{E}[I(\theta_{i+1})] &= \int_{-\infty}^{\infty} \max\{\bar{y}_i - y_{i+1}, 0\} dp(y_{i+1}|\theta_{i+1}, D_i) = \\ &= \int_{-\infty}^{\bar{y}_i} (\bar{y}_i - y_{i+1}) \frac{p(\theta_{i+1}|y_{i+1}, D_i)p(y_{i+1}|D_i)}{p(\theta_{i+1}|D_i)} dy\end{aligned} \tag{A.1}$$

That is, the expected improvement is the expectation under the model $p(y_{i+1}|\theta_{i+1}, D_i)$ that the performance y_{i+1} (in terms of loss) will be below some threshold value $\bar{y}_i$. The TPE does not model the posterior distribution and models the likelihood and the prior distribution instead. Each iteration the TPE sets $\bar{y}_i$ to the empirical quantile γ of the accumulated objective function evaluations and thus the prior $p(y_{i+1} < \bar{y}_i) = \gamma$. Then the algorithm splits the history D_i into two sets corresponding to 'good' and 'bad' regions in the configuration space, i.e. the 'good' region contains observations where values of the objective function are below $\bar{y}_i$. Then TPE estimates probability distributions $l(\theta)$, $g(\theta)$ of hyperparameters in the 'good' and 'bad' sets respectively using adaptive Parzen windows. Therefore the likelihood can be defined as follows:

$$p(\theta_{i+1}|y_{i+1}, D_i) = \begin{cases} l(\theta), & \text{if } y_{i+1} < \bar{y}_i \\ g(\theta), & \text{if } y_{i+1} \geq \bar{y}_i \end{cases} \tag{A.2}$$

Combining the prior $p(y_{i+1} < \bar{y}_i) = \gamma$ with the likelihood function A.2 Bergstra et al. (2011) derive the following expression for the expected improvement A.1:

$$\mathbb{E}[I(\theta_{i+1})] \propto \left(\gamma + \frac{g(\theta)}{l(\theta)}(1-\gamma)\right)^{-1} \tag{A.3}$$

The expected improvement increases when the next iteration's set of hyperparameters has higher probability under $l(\theta)$ and lower probability under $g(\theta)$. Finally, the TPE samples n_{EI} values from $l(\theta)$, evaluates them according to $l(\theta)/g(\theta)$ and uses the value maximizing A.1 to evaluate the objective before proceeding to the next iteration. The algorithm is initialized by accumulating the initial history $D_{init} = (y_k, \theta_k)_{k=1}^{n_0}$ by drawing n_0 samples from the prior distributions of hyperparameters $p_0(\theta)$ and evaluating the objective. Algorithm 1 summarizes the TPE procedure.

Algorithm 1: Tree-structured Parzen Estimator

Input: n_0, γ, n_{EI}, $p_0(\theta)$, N - the maximum number of iterations

Initialize: draw n_0 samples from $p_0(\theta)$ and accumulate D_{init}

for *i=0 to N* **do**

- Compute $\bar{y}_{n_0+i}$
- Compute $D^l_{n_0+i} = \{(y_k, \theta_k) | y_k < \bar{y}_{n_0+i}\}_{k=0}^{n_0+i}$, estimate $l(\theta)$
- Compute $D^g_{n_0+i} = \{(y_k, \theta_k) | y_k \geq \bar{y}_{n_0+i}\}_{k=0}^{n_0+i}$, estimate $g(\theta)$
- Draw n_{EI} samples from $l(\theta)$: $\theta_{EI} = \{\theta_q : q = 1, \ldots, n_{EI}\}$, $\theta_q \sim l(\theta)$
- $\theta_{i+1} = \arg\max_{\theta \in \theta EI} \mathbb{E}[I(\theta_{i+1})]$
- Evaluate $y_{i+1} = f(\theta_{i+1})$
- Update observations $D_{n_0+i} \leftarrow D_{n_0+i} \cup (y_{i+1}, \theta_{i+1})$

end

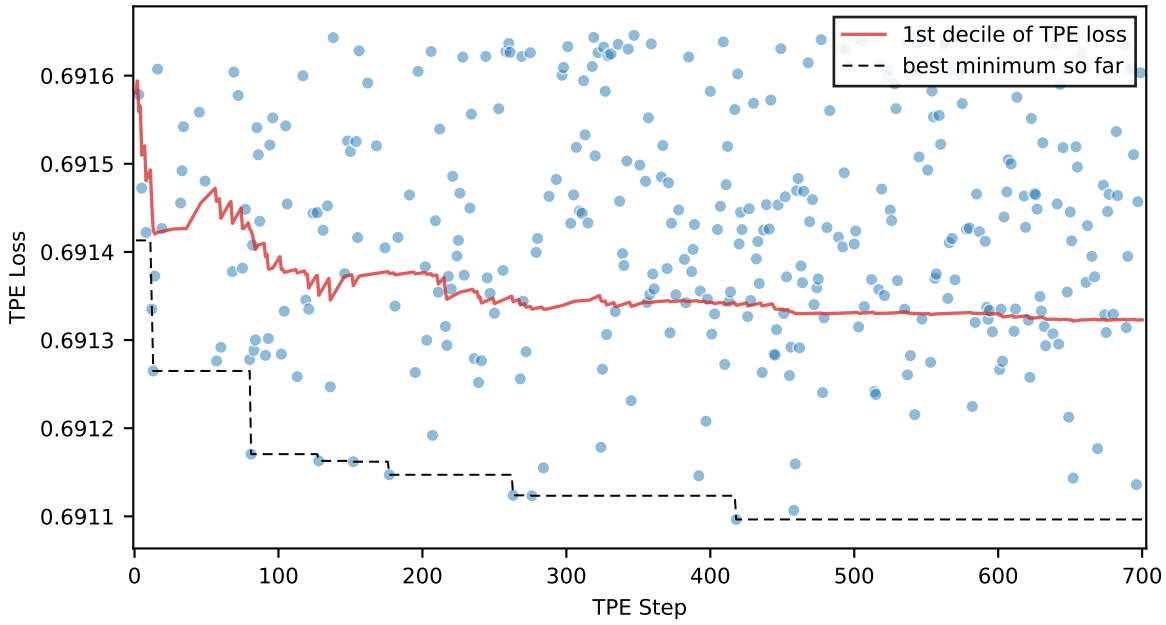
Appendix B TPE Optimization of Deep Architectures

I begin the second pass of the TPE optimization with updating the hyperparameter priors in line with the results displayed for the first pass in the main text. To speed up evaluations I constrain the values of hyperparameters removing ranges which are unlikely to yield any well-performing architectures: specifically, I constrain the learning rate to a narrower range, exclude the lower values for the dropout rate, restrict the number of units in the first and last hidden layers and make the discrete uniform grid for number of units in each hidden layer coarser. Table B.I summarizes the updated priors. The training regime is described in the main text and stays unchanged.

Table B.I: Prior distributions of hyperparameters

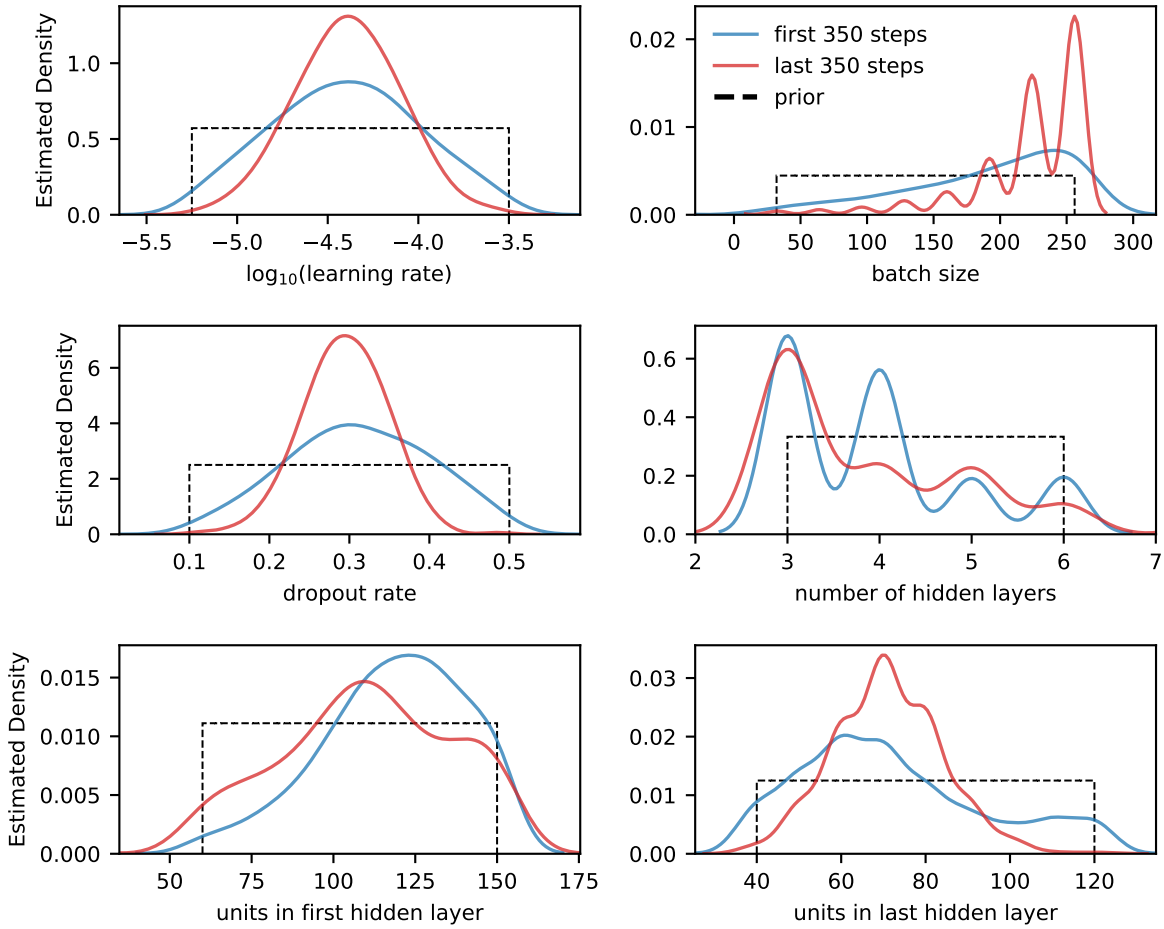
Hyperparameter	Prior Distribution
Learning rate	$10^u,\ u \sim U(-5.25, -3.5)$
Batch size	$\{32, 64, \ldots, 256\}$
Dropout rate	$U(0.1, 0.5)$
Number of hidden layers	$\{3, 4, \ldots, 6\}$
Number of units in the first hidden layer	$\{60, 70, \ldots, 150\}$
Number of units in the last hidden layer	$\{40, 50, \ldots, 120\}$
Number of units in other hidden layers	$\{20, 30, \ldots, 170\}$

Figure B.1: TPE hyperparameter optimization progress



This figure depicts progress of the TPE hyperparameter optimization. The iterations and the TPE objective function values are along the horizontal and vertical axis respectively. The blue dots show the best 50% of the TPE evaluations, the solid red line is the expanding first decile of the TPE loss and the dashed black line tracks the best TPE loss at each iteration.

Figure B.2: Distributions of hyperparameters during TPE optimization

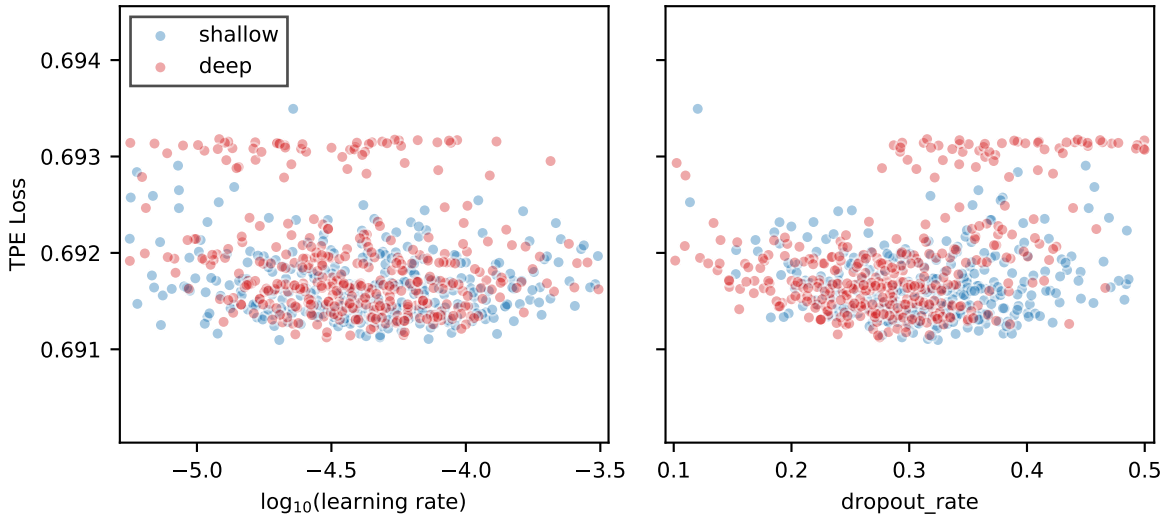


This figure displays distributions of hyperparameters over the course of TPE optimization. In each plot the solid blue and red lines draw the kernel density estimates of distributions of hyperparameters tried in the first and second halves of the TPE iterations respectively, and the dashed black line shows the prior distribution as presented in Table B.I. The plots depict densities for (from top to bottom and left to right) base-10 logarithm of the learning rate, batch size, dropout rate, number of layers and number of units in the first and last hidden layers. The discrete uniform priors for batch size, number of layers and number of units are drawn as continuous uniform densities over the same support.

Figure B.1 repeats the analysis of Figure 1 from the main text with virtually the same results. Similarly, Figure B.2 repeats the analysis of Figure 2, the results are however quite different. Consistent with the results of the first TPE run discussed in the main text, as TPE accumulates observations the proposed deep architectures gravitate towards having lower dropout rate and higher learning rate comparing to their shallower counterparts. The most striking feature is the increase in the batch size: whereas the best shallow models in the first TPE pass mainly have batch sizes of 32 and 64, the deeper architectures tend to concentrate in the upper range of values of this hyperparameter. These shifts in the optimal values of hyperparameters correspond to a reduction in (implicit) regularization of the deeper models to enable the learning of meaningful data representations within the given number of epochs which I keep fixed. See Smith et al. (2017) and references therein for additional information on

the regularizing properties of the batch size and interplay between the batch size and learning rate.

Figure B.3: Hyperparameters of 3 vs. 3+ layer deep models



This figure plots the TPE loss as a function of learning rate (left plot) and dropout rate (right plot). The blue dots depict the TPE losses of the shallower architectures with three hidden layers, and the red dots show the losses of the deeper architectures with more than three hidden layers.

Finally, Figure B.3 repeats the analysis of Figure 3 but now it compares the architectures with three and four and more layers. It further corroborates the conjecture that the deeper models capable of achieving good validation set performance display better convergence with reduced regularization.

Appendix C Proofs

Proof of Proposition 1

Proof. Write the output of the network as $p = (1 + e^{-h(X)})^{-1}$ where h is an affine function of the ReLU activations in the penultimate layer. The partial derivative of the output with respect to input i is then $\frac{\partial h(X)}{\partial x_i} p(1-p)$; differentiating it with respect to x_j yields:

$$\begin{aligned} \frac{\partial^2 p}{\partial x_i \partial x_j} &= \frac{\partial^2 h(X)}{\partial x_i \partial x_j} p(1-p) + \frac{\partial h(X)}{\partial x_i} \frac{\partial h(X)}{\partial x_j} p(1-p)(1-2p) = \\ &= \frac{\partial p}{\partial x_i} \frac{\partial p}{\partial x_j} \frac{1-2p}{p(1-p)} \end{aligned} \tag{C.1}$$

The first term in the first line of the equation is zero h because the gradients of ReLU units are constant and the last equality follows from the definition of the first order partial derivative. □

Appendix D Additional Asset Pricing Tests

Table D.I: Time-series asset pricing tests: hedge fund factors

Portfolio	α	$R_m - R_f$	SMB	HML	RMW	CMA	MOM	BD	FX	COM	IR	STK	$\bar{R}^2$
Panel A: Median Sorts													
P1	−3.41	1.02	0.08	0.06	0.00	0.09	0.01	0.00	−0.00	−0.00	−0.00	−0.00	0.94
	[−4.58]	[48.68]	[3.04]	[2.02]	[0.09]	[1.25]	[0.42]	[0.59]	[−1.10]	[−1.08]	[−1.33]	[−0.80]	
P2	3.51	1.00	0.04	0.04	0.08	0.02	−0.07	0.00	0.01	−0.00	−0.01	−0.01	0.92
	[4.11]	[51.41]	[0.99]	[0.78]	[1.45]	[0.35]	[−2.66]	[0.17]	[2.01]	[−0.56]	[−1.26]	[−2.21]	
P2-P1	6.91	−0.02	−0.04	−0.02	0.07	−0.06	−0.08	−0.00	0.01	0.00	−0.00	−0.01	0.04
	[5.97]	[−0.86]	[−0.67]	[−0.36]	[0.89]	[−0.64]	[−1.85]	[−0.27]	[2.07]	[0.30]	[−0.65]	[−1.20]	
Panel B: Decile Sorts													
P1	−9.65	1.05	0.17	−0.07	0.16	0.18	−0.11	0.02	−0.02	−0.01	−0.01	−0.00	0.75
	[−5.03]	[21.21]	[2.64]	[−1.09]	[1.59]	[1.11]	[−1.57]	[1.30]	[−1.62]	[−1.13]	[−1.79]	[−0.07]	
P2	−5.18	1.01	0.08	0.10	0.11	0.10	0.02	0.00	−0.00	−0.01	−0.00	0.00	0.85
	[−3.96]	[34.72]	[2.14]	[1.93]	[2.16]	[1.17]	[0.69]	[0.45]	[−0.86]	[−1.23]	[−0.68]	[0.00]	
P3	−1.18	0.99	0.08	0.08	0.01	0.04	−0.01	0.00	−0.00	0.00	−0.00	−0.01	0.89
	[−1.19]	[40.12]	[1.58]	[1.90]	[0.10]	[0.60]	[−0.23]	[0.47]	[−0.45]	[0.30]	[−0.69]	[−2.43]	
P4	−1.32	1.04	0.02	0.13	−0.12	0.06	0.10	−0.01	−0.00	−0.01	−0.00	−0.00	0.91
	[−1.29]	[44.56]	[0.90]	[2.38]	[−2.89]	[0.93]	[3.73]	[−0.92]	[−0.67]	[−1.65]	[−0.19]	[−0.53]	
P5	0.25	1.00	0.05	0.09	−0.15	0.05	0.05	−0.00	0.01	0.00	−0.00	0.00	0.90
	[0.26]	[37.18]	[1.54]	[2.11]	[−3.03]	[0.83]	[1.66]	[−0.55]	[1.40]	[0.88]	[−0.03]	[0.33]	
P6	1.51	1.03	−0.01	0.07	−0.09	0.05	0.04	−0.01	0.01	0.00	−0.01	−0.00	0.88
	[1.20]	[31.20]	[−0.21]	[1.16]	[−1.21]	[0.78]	[1.40]	[−1.28]	[0.90]	[0.08]	[−1.89]	[−0.31]	
P7	0.98	0.99	0.04	0.06	−0.03	0.01	−0.00	−0.00	0.01	−0.01	−0.01	−0.01	0.89
	[0.89]	[33.31]	[0.85]	[0.81]	[−0.55]	[0.12]	[−0.08]	[−0.24]	[1.78]	[−0.95]	[−1.42]	[−0.87]	
P8	3.29	0.98	0.03	0.00	0.08	0.04	−0.10	−0.00	0.01	0.00	−0.01	−0.01	0.86
	[2.60]	[32.58]	[0.58]	[0.01]	[1.06]	[0.39]	[−2.62]	[−0.25]	[1.20]	[0.39]	[−0.91]	[−1.29]	
P9	3.93	0.98	0.06	0.03	0.15	0.03	−0.08	0.00	0.01	−0.00	−0.00	−0.02	0.85
	[3.47]	[39.31]	[1.20]	[0.38]	[2.19]	[0.28]	[−2.13]	[0.20]	[2.18]	[−0.20]	[−0.70]	[−2.16]	
P10	7.78	0.99	0.09	0.06	0.27	−0.02	−0.23	0.02	0.02	−0.01	−0.00	−0.03	0.80
	[5.28]	[31.40]	[1.32]	[0.89]	[3.59]	[−0.15]	[−4.59]	[1.89]	[2.13]	[−1.33]	[−0.56]	[−2.63]	
P10-P1	17.43	−0.06	−0.08	0.14	0.10	−0.19	−0.11	−0.00	0.03	0.00	0.01	−0.03	0.04
	[6.48]	[−0.97]	[−0.74]	[1.33]	[0.75]	[−1.00]	[−1.02]	[−0.15]	[2.39]	[0.11]	[0.74]	[−1.68]	

The table reports results of time-series regressions for excess returns on portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The test portfolios are across rows and explanatory variables are across columns. The first column reports pricing errors in percent p.a.; the regressors include the five factors of Fama and French (2015), 12-1 momentum, and five trend-following factors of Fung and Hsieh (2001) for bonds, currencies, commodities, interest rates and stocks. The last column shows the adjusted R^2. Panels A and B report results for median and decile sorts respectively. The numbers in brackets are the Newey and West (1987) t-statistics with number of lags according to Newey and West (1994). The sample is from January 1994 to December 2018.

Table D.II: Descriptive statistics: long-short portfolios, restricted model

	Median			Quintile			Decile		
Portfolio	1	2	2-1	1	5	5-1	1	10	10-1
Mean	5.32	11.48	6.17	2.45	13.34	10.90	0.29	15.27	14.98
[t-stat]	[1.69]	[3.80]	[5.18]	[0.74]	[4.48]	[6.12]	[0.08]	[4.99]	[6.53]
Median	11.72	16.96	5.07	7.70	19.80	8.99	5.20	21.72	15.04
Std	15.57	15.47	6.31	16.28	15.50	9.99	17.87	16.25	12.25
Skew	−0.61	−0.81	−0.64	−0.57	−0.77	−0.48	−0.52	−0.63	−0.08
Kurtosis	1.32	2.61	10.50	1.13	2.33	7.76	1.34	2.26	2.03
Sharpe	0.34	0.74	0.98	0.15	0.86	1.09	0.02	0.94	1.22
MaxDD	58.24	49.80	17.06	76.72	47.93	21.63	86.29	48.99	18.48
Max 1 M loss	18.94	21.19	12.98	17.81	19.40	19.06	19.51	21.34	15.55
Turnover	79.37	76.37	155.75	121.00	125.42	246.42	140.32	146.71	287.03

The table reports descriptive statistics of excess returns on portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The predicted probability is estimated from a restricted version of the model which suppresses effects of market state variables by setting them to their unconditional means – zeros – over the course of the test sample. Each triplet of columns shows descriptive statistics of a low, high, and high-minus-low portfolio for median, quintile and decile sorts. Means, medians, standard deviations are in percent p.a., Sharpe ratios are annualized, maximum drawdowns and maximum 1-month losses are in percent, average turnover is in percent per month. The numbers in brackets are the Newey and West (1987) t-statistics for the null hypothesis of mean return being zero with number of lags according to Newey and West (1994). The sample is from January 1990 to December 2018.

Table D.III: Time-series asset pricing tests: restricted model

Portfolio	α	$R_m - R_f$	SMB	HML	RMW	CMA	MOM	$\bar{R}^2$
Panel A: Median Sorts								
P1	−2.17	1.02	0.05	0.05	0.06	0.09	−0.11	0.95
	[−3.49]	[56.97]	[2.06]	[1.63]	[1.25]	[1.37]	[−7.64]	
P2	2.96	1.04	0.09	0.05	0.06	0.01	0.05	0.93
	[3.65]	[44.18]	[2.56]	[1.01]	[1.05]	[0.17]	[2.31]	
P2-P1	5.13	0.02	0.04	−0.00	0.00	−0.07	0.16	0.17
	[5.16]	[0.67]	[0.88]	[−0.05]	[0.00]	[−0.71]	[6.78]	
Panel B: Decile Sorts								
P1	−6.02	1.00	0.08	−0.01	0.09	0.11	−0.30	0.82
	[−4.02]	[30.27]	[1.57]	[−0.17]	[0.93]	[0.89]	[−8.22]	
P2	−2.76	1.00	0.01	0.01	0.10	0.12	−0.13	0.87
	[−2.90]	[39.00]	[0.24]	[0.14]	[1.46]	[1.39]	[−5.81]	
P3	−1.28	1.02	0.04	0.08	0.07	0.09	−0.11	0.89
	[−1.36]	[37.71]	[1.16]	[1.91]	[1.21]	[0.92]	[−5.23]	
P4	−0.59	1.01	0.07	0.09	0.03	0.10	−0.02	0.89
	[−0.56]	[43.89]	[2.46]	[1.58]	[0.59]	[1.27]	[−1.07]	
P5	−0.25	1.07	0.03	0.08	−0.02	0.02	−0.01	0.92
	[−0.27]	[42.66]	[1.17]	[1.54]	[−0.48]	[0.26]	[−0.27]	
P6	0.41	1.08	0.11	0.04	0.03	0.10	0.07	0.91
	[0.39]	[50.20]	[3.63]	[0.90]	[0.65]	[1.32]	[2.77]	
P7	1.87	1.07	0.06	0.04	0.01	0.04	0.07	0.91
	[1.99]	[35.45]	[1.99]	[0.88]	[0.09]	[0.46]	[3.04]	
P8	2.34	1.03	0.09	0.09	0.00	−0.01	0.05	0.86
	[2.12]	[28.51]	[1.56]	[1.44]	[0.02]	[−0.12]	[1.24]	
P9	3.37	0.97	0.10	0.03	0.03	−0.03	0.08	0.82
	[2.92]	[34.09]	[1.68]	[0.45]	[0.29]	[−0.26]	[2.07]	
P10	6.78	1.02	0.07	0.03	0.21	−0.03	−0.02	0.80
	[4.96]	[29.37]	[1.27]	[0.56]	[3.89]	[−0.29]	[−0.59]	
P10-P1	12.81	0.02	−0.01	0.04	0.12	−0.14	0.28	0.13
	[5.49]	[0.52]	[−0.07]	[0.45]	[1.08]	[−0.87]	[5.72]	

The table reports results of time-series regressions for excess returns on portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The predicted probability is estimated from a restricted version of the model which suppresses effects of market state variables by setting them to their unconditional means – zeros – over the course of the test sample. The test portfolios are across rows and explanatory variables are across columns. The first column reports pricing errors in percent p.a.; the regressors include the five factors of Fama and French (2015) plus 12-1 momentum. The last column shows the adjusted R^2. Panels A and B report results for median and decile sorts respectively. The numbers in brackets are the Newey and West (1987) t-statistics with number of lags according to Newey and West (1994). The sample is from January 1990 to December 2018.

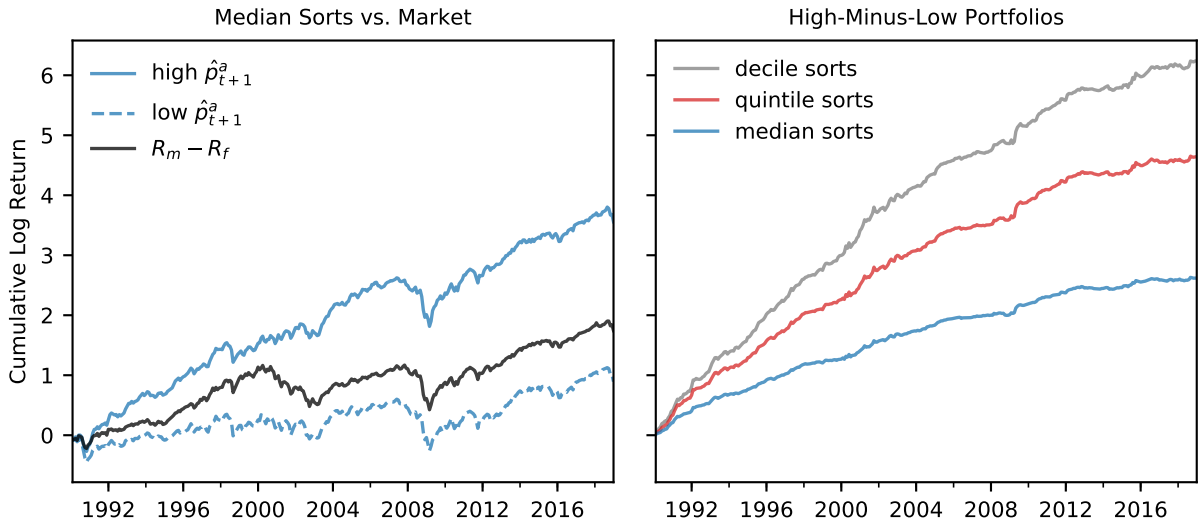
Internet Appendix

A. All CRSP Stocks

I construct a set of features for all CRSP stocks using the procedure outlined in Section II and use the ensemble of neural networks from the main text of the paper to predict returns on the test sample without any additional training. The number of stock-month observations in the test set increases more than sevenfold from over 170,000 to over 1.2 million. I then repeat the asset pricing analysis from Section IV of the main text for both equally- and value-weighted portfolios of stocks from the whole CRSP universe. See the notes below each table and figure for an extensive description.

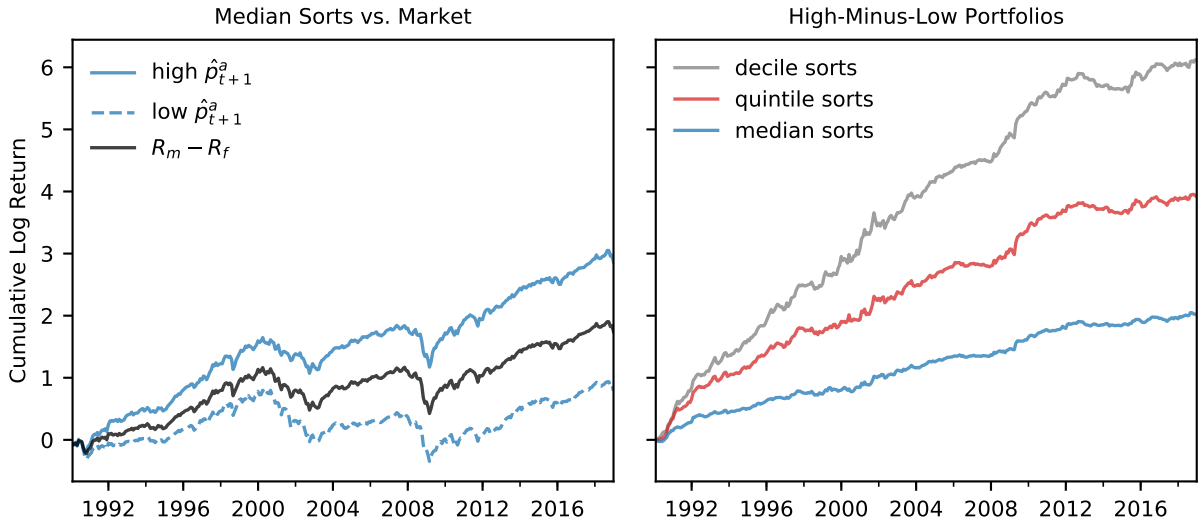
Table I reports descriptive statistics for the 'low', 'high', and 'high-minus-low' portfolios for median, quintile and decile sorts. The equally- and value-weighted portfolios are in Panels A and B respectively. Similarly, Table II reports descriptive statistics for individual portfolios from the decile sorts. Figure 1 plots cumulative log returns on the equally-weighted ensemble portfolios. The left panel shows the returns of the high and low portfolios sorted by median $\hat{p}^a_{t+1}$ (solid and dashed blue lines) along with the return on the CRSP value weighted index in black. The right panel displays the returns on the long-short portfolios: median, quintile, and decile sorts are drawn in blue, red, and gray respectively. Figure 2 repeats this analysis for the value-weighted portfolios. Tables III and IV repeat the seven factor model time-series asset pricing tests from Table VI in the main text using equally and value-weighted long-short portfolios of stocks sorted on SDF loadings of Chen et al. (2019) as the seventh pricing factor in addition to the Fama and French (2015) factors and momentum.

Figure 1: Return on predicted probability portfolios: all stocks, equal weights



The figure plots out-of-sample performance of equally-weighted portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The left plot shows cumulative log excess returns on portfolios investing in stocks with 50% highest and lowest $\hat{p}^a_{t+1}$ (solid and dashed blue lines) and on the CRSP value weighted index (solid black line). The right plot displays cumulative log returns on long-short predicted probability portfolios: median, quintile, and decile sorts are drawn in blue, red, and gray respectively. All portfolios are rebalanced monthly. The sample is from January 1990 to December 2018 and includes all CRSP stocks.

Figure 2: Return on predicted probability portfolios: all stocks, value weights



The figure plots out-of-sample performance of value-weighted portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The left plot shows cumulative log excess returns on portfolios investing in stocks with 50% highest and lowest $\hat{p}^a_{t+1}$ (solid and dashed blue lines) and on the CRSP value weighted index (solid black line). The right plot displays cumulative log returns on long-short predicted probability portfolios: median, quintile, and decile sorts are drawn in blue, red, and gray respectively. All portfolios are rebalanced monthly. The sample is from January 1990 to December 2018 and includes all CRSP stocks.

Table I: Descriptive statistics, all stocks: long-short portfolios

	Median			Quintile			Decile		
Portfolio	1	2	2-1	1	5	5-1	1	10	10-1
Panel A: Equal Weights									
Mean	4.53	13.73	9.20	0.03	16.52	16.50	−3.54	18.78	22.31
[t-stat]	[1.37]	[4.01]	[8.99]	[0.01]	[4.52]	[9.63]	[−0.98]	[4.91]	[10.52]
Median	10.30	18.20	7.11	7.89	20.04	13.50	3.55	22.02	18.80
Std	16.81	16.65	5.16	18.01	17.78	8.78	19.11	18.84	11.04
Skew	−0.76	−0.58	0.74	−0.79	−0.46	0.86	−0.70	−0.34	0.76
Kurtosis	1.55	1.92	2.57	1.87	2.38	2.93	1.82	2.32	2.44
Sharpe	0.27	0.82	1.78	0.00	0.93	1.88	−0.19	1.00	2.02
MaxDD	57.63	55.48	4.52	71.32	56.40	6.59	86.91	56.76	8.51
Max 1 M loss	20.11	20.19	4.06	22.90	21.62	6.07	24.40	22.40	7.48
Turnover	80.44	77.93	158.37	127.03	120.96	247.99	147.29	141.15	288.44
Panel B: Value Weights									
Mean	3.89	11.08	7.19	0.16	14.51	14.34	−4.45	18.11	22.55
[t-stat]	[1.30]	[3.72]	[6.04]	[0.05]	[4.39]	[6.29]	[−1.24]	[5.10]	[7.63]
Median	11.08	14.38	4.49	7.04	17.34	11.13	0.80	18.59	17.08
Std	15.05	15.18	6.88	16.68	16.89	12.64	18.72	18.86	16.18
Skew	−0.84	−0.47	0.81	−1.06	−0.07	0.81	−0.87	0.11	0.65
Kurtosis	1.77	1.41	2.89	3.85	2.53	3.49	3.32	2.88	2.23
Sharpe	0.26	0.73	1.05	0.01	0.86	1.14	−0.24	0.96	1.39
MaxDD	68.17	48.94	7.03	77.51	51.33	16.11	92.40	47.45	26.13
Max 1 M loss	19.81	16.16	5.30	27.16	17.68	9.50	28.42	19.49	11.64
Turnover	93.80	90.75	184.55	149.09	136.64	285.73	165.97	155.42	321.39

The table reports descriptive statistics of excess returns on portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. Each triplet of columns shows descriptive statistics of a low, high, and high-minus-low portfolio for median, quintile and decile sorts. Means, medians, standard deviations are in percent p.a., Sharpe ratios are annualized, maximum drawdowns and maximum 1-month losses are in percent, average turnover is in percent per month. The numbers in brackets are the Newey and West (1987) t-statistics for the null hypothesis of mean return being zero with number of lags according to Newey and West (1994). Panels A and B report results for equally- and value-weighted portfolios respectively. The sample is from January 1990 to December 2018 and includes all CRSP stocks.

Table II: Descriptive statistics, all stocks: decile portoflios

Portfolio	1	2	3	4	5	6	7	8	9	10
Panel A: Equal Weights										
Mean	−3.54	3.59	5.84	7.89	8.87	11.07	12.31	12.23	14.27	18.78
[t-stat]	[−0.98]	[1.05]	[1.77]	[2.43]	[2.77]	[3.42]	[3.70]	[3.57]	[4.04]	[4.91]
Median	3.55	9.63	11.88	12.98	15.70	16.16	18.52	18.65	19.48	22.02
Std	19.11	17.25	16.80	16.18	16.27	16.00	16.06	16.52	17.00	18.84
Skew	−0.70	−0.85	−0.75	−0.67	−0.65	−0.59	−0.58	−0.66	−0.53	−0.34
Kurtosis	1.82	2.02	1.71	1.18	1.25	1.31	1.72	1.91	2.33	2.32
Sharpe	−0.19	0.21	0.35	0.49	0.55	0.69	0.77	0.74	0.84	1.00
MaxDD	86.91	61.09	57.66	53.55	52.19	52.76	54.68	57.28	56.09	56.76
Max 1 M loss	24.40	21.63	20.15	18.11	17.92	18.64	19.27	19.82	20.83	22.40
Turnover	147.29	169.01	173.54	174.96	175.57	175.41	174.73	172.45	167.33	141.15
Panel B: Value Weights										
Mean	−4.45	2.98	4.17	5.08	7.19	8.16	10.50	10.10	11.82	18.11
[t-stat]	[−1.24]	[0.91]	[1.34]	[1.73]	[2.18]	[2.64]	[3.63]	[3.17]	[3.59]	[5.10]
Median	0.80	10.39	12.96	11.74	11.76	14.19	14.39	14.88	14.43	18.59
Std	18.72	16.51	15.70	15.39	15.95	15.08	15.63	16.51	16.23	18.86
Skew	−0.87	−1.03	−0.83	−0.64	−0.68	−0.55	−0.54	−0.70	−0.33	0.11
Kurtosis	3.32	3.80	1.62	1.21	1.48	1.01	1.51	1.87	1.46	2.88
Sharpe	−0.24	0.18	0.27	0.33	0.45	0.54	0.67	0.61	0.73	0.96
MaxDD	92.40	72.55	67.70	62.90	59.59	56.83	47.27	52.75	53.86	47.45
Max 1 M loss	28.42	26.81	21.02	15.26	19.96	17.24	16.55	18.94	16.23	19.49
Turnover	165.97	174.14	175.54	174.58	174.78	176.03	176.18	175.82	171.44	155.42

The table reports descriptive statistics of excess returns on decile portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. Means, medians, standard deviations are in percent p.a., Sharpe ratios are annualized, maximum drawdowns and maximum 1-month loss are in percent, average turnover is in percent per month. The numbers in brackets are the Newey and West (1987) t-statistics for the null hypothesis of mean return being zero with number of lags according to Newey and West (1994). Panels A and B report results for equally- and value-weighted portfolios respectively. The sample is from January 1990 to December 2018 and includes all CRSP stocks.

Table III: Time-series asset pricing tests: FF-5 + MOM + Chen et al. (2019) HML EW

	Top 500 MV			All CRSP EW			All CRSP VW		
Portfolio	α	SDF	$\bar{R}^2$	α	SDF	$\bar{R}^2$	α	SDF	$\bar{R}^2$
Panel A: Median Sorts									
P1	−1.83	−0.04	0.94	−3.67	0.01	0.97	−2.81	−0.02	0.94
	[−1.91]	[−3.06]		[−3.93]	[0.68]		[−3.68]	[−1.81]	
P2	3.82	0.01	0.92	2.57	0.07	0.96	2.50	0.03	0.94
	[2.93]	[0.41]		[2.44]	[4.69]		[2.92]	[1.56]	
P2-P1	5.65	0.05	0.05	6.24	0.06	0.09	5.31	0.05	0.05
	[3.28]	[1.92]		[4.18]	[2.67]		[3.53]	[2.00]	
Panel B: Decile Sorts									
P1	−8.65	−0.03	0.74	−10.77	0.01	0.86	−9.15	−0.08	0.68
	[−3.27]	[−0.81]		[−5.12]	[0.24]		[−3.28]	[−1.99]	
P2	−4.52	−0.03	0.85	−4.58	−0.01	0.93	−4.00	−0.04	0.79
	[−3.10]	[−1.56]		[−3.59]	[−0.46]		[−2.29]	[−1.32]	
P3	0.11	−0.01	0.88	−2.48	0.00	0.96	−3.13	0.01	0.87
	[0.09]	[−0.48]		[−2.89]	[0.26]		[−2.45]	[0.51]	
P4	0.65	−0.05	0.91	−0.57	0.02	0.96	−1.64	−0.00	0.86
	[0.65]	[−3.15]		[−0.71]	[1.76]		[−1.17]	[−0.10]	
P5	3.17	−0.09	0.91	0.06	0.02	0.97	−0.12	−0.01	0.90
	[2.62]	[−6.15]		[0.08]	[1.37]		[−0.09]	[−0.37]	
P6	1.97	−0.01	0.88	1.56	0.03	0.96	3.24	−0.02	0.88
	[1.47]	[−0.56]		[1.60]	[1.46]		[2.54]	[−0.64]	
P7	2.76	−0.05	0.89	2.82	0.03	0.95	2.97	0.00	0.89
	[1.87]	[−2.70]		[2.59]	[1.99]		[2.41]	[0.01]	
P8	4.33	0.00	0.85	1.19	0.07	0.94	0.03	0.05	0.86
	[2.61]	[0.01]		[1.00]	[3.78]		[0.02]	[1.45]	
P9	3.61	0.03	0.85	1.46	0.09	0.92	0.51	0.06	0.82
	[2.09]	[1.08]		[1.01]	[5.27]		[0.31]	[2.11]	
P10	6.40	0.06	0.80	5.81	0.11	0.90	8.41	0.04	0.74
	[3.05]	[1.67]		[3.13]	[5.33]		[3.01]	[0.83]	
P10-P1	15.06	0.09	0.04	16.57	0.11	0.05	17.56	0.12	0.01
	[4.11]	[1.60]		[4.91]	[2.25]		[3.60]	[1.52]	

The table reports results of time-series regressions for excess returns on portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The test portfolios are across rows and explanatory variables include five Fama and French (2015) factors, momentum and long-short equally-weighted decile portfolio of Chen et al. (2019) sorted on SDF loadings. Each triplet of columns reports pricing error in percent p.a., regression coefficient on the SDF and adjusted R^2 of the regression. The first triplet reports estimates for the large cap sample with test assets being equally-weighted portfolios constructed in the universe of five hundred largest stocks. The second triplet and third triplets repeat the exercise for the entire CRSP sample with equal- and value-weighted test portfolios respectively. Panels A and B report results for median and decile sorts respectively. The numbers in brackets are the Newey and West (1987) t-statistics with number of lags according to Newey and West (1994). The sample is from January 1992 to December 2016.

Table IV: Time-series asset pricing tests: FF-5 + MOM + Chen et al. (2019) HML VW

	Top 500 MV			All CRSP EW			All CRSP VW		
Portfolio	α	SDF	$\bar{R}^2$	α	SDF	$\bar{R}^2$	α	SDF	$\bar{R}^2$
Panel A: Median Sorts									
P1	−2.84	−0.03	0.94	−3.50	0.01	0.97	−3.20	−0.02	0.94
	[−3.74]	[−2.63]		[−5.25]	[0.94]		[−4.90]	[−3.00]	
P2	3.88	0.01	0.92	4.19	0.04	0.95	3.00	0.03	0.94
	[3.82]	[0.65]		[4.70]	[3.86]		[4.90]	[2.27]	
P2-P1	6.72	0.04	0.04	7.69	0.04	0.07	6.20	0.06	0.06
	[5.30]	[1.63]		[7.00]	[2.43]		[5.18]	[3.00]	
Panel B: Decile Sorts									
P1	−8.78	−0.05	0.74	−10.71	0.01	0.86	−10.72	−0.08	0.68
	[−4.23]	[−1.80]		[−7.23]	[0.50]		[−4.72]	[−2.56]	
P2	−5.33	−0.01	0.85	−4.90	0.00	0.93	−4.32	−0.06	0.79
	[−4.20]	[−0.57]		[−5.12]	[0.03]		[−3.19]	[−2.33]	
P3	−0.29	0.00	0.88	−2.30	−0.00	0.96	−2.65	−0.01	0.87
	[−0.30]	[0.29]		[−3.54]	[−0.27]		[−2.07]	[−0.57]	
P4	−0.85	−0.03	0.91	−0.05	0.02	0.96	−1.97	0.01	0.86
	[−0.88]	[−1.67]		[−0.07]	[1.80]		[−1.57]	[0.75]	
P5	0.99	−0.06	0.91	0.45	0.02	0.97	−0.25	−0.01	0.90
	[0.95]	[−3.56]		[0.76]	[1.48]		[−0.20]	[−0.51]	
P6	1.85	−0.02	0.88	2.47	0.01	0.96	2.68	−0.01	0.88
	[1.45]	[−0.77]		[3.32]	[0.51]		[2.74]	[−0.33]	
P7	1.43	−0.02	0.89	3.56	0.02	0.95	2.69	0.01	0.89
	[1.10]	[−1.29]		[3.81]	[1.33]		[2.74]	[0.79]	
P8	4.70	−0.02	0.85	2.91	0.04	0.94	0.91	0.05	0.87
	[3.56]	[−0.84]		[2.96]	[2.90]		[0.91]	[1.84]	
P9	3.94	0.04	0.85	3.66	0.07	0.92	1.88	0.05	0.82
	[2.82]	[1.54]		[2.91]	[4.66]		[1.41]	[1.81]	
P10	7.45	0.06	0.80	8.32	0.09	0.89	8.94	0.05	0.74
	[4.43]	[1.81]		[5.59]	[4.83]		[4.62]	[1.42]	
P10-P1	16.23	0.11	0.06	19.02	0.08	0.04	19.66	0.13	0.01
	[5.81]	[2.66]		[7.91]	[2.55]		[5.67]	[2.39]	

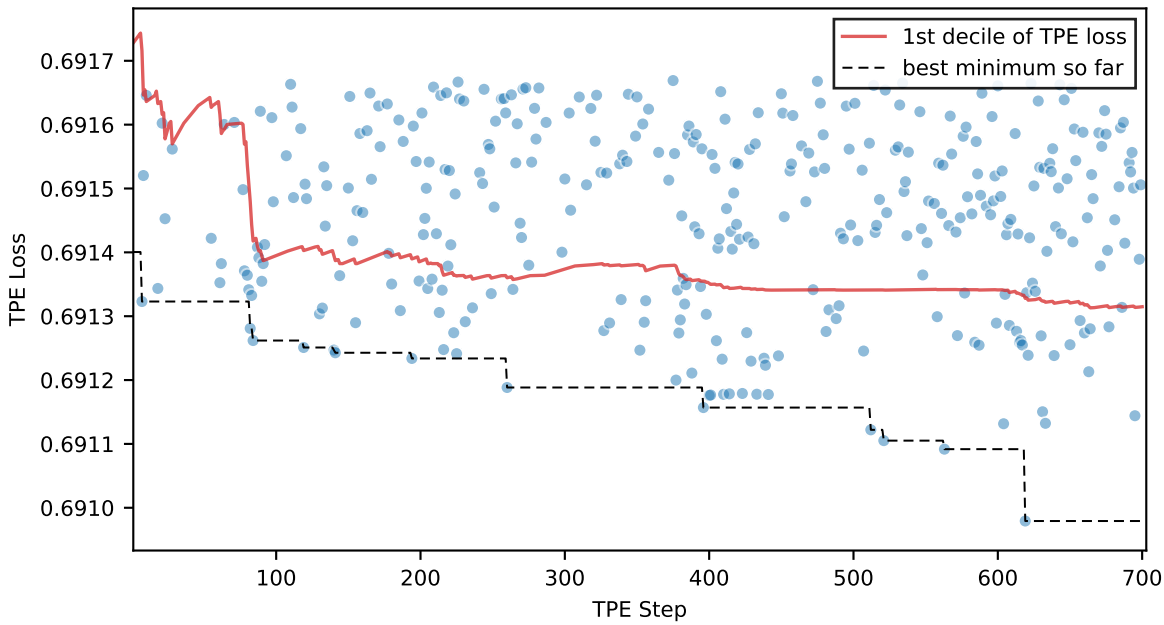
The table reports results of time-series regressions for excess returns on portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The test portfolios are across rows and explanatory variables include five Fama and French (2015) factors, momentum and long-short value-weighted decile portfolio of Chen et al. (2019) sorted on SDF loadings. Each triplet of columns reports pricing error in percent p.a., regression coefficient on the SDF and adjusted R^2 of the regression. The first triplet reports estimates for the large cap sample with test assets being equally-weighted portfolios constructed in the universe of five hundred largest stocks. The second triplet and third triplets repeat the exercise for the entire CRSP sample with equal- and value-weighted test portfolios respectively. Panels A and B report results for median and decile sorts respectively. The numbers in brackets are the Newey and West (1987) t-statistics with number of lags according to Newey and West (1994). The sample is from January 1992 to December 2016.

B. Total Returns as Prediction Targets

I repeat the key results estimating the model using total returns both to construct input variables and as prediction targets. Overall, using total returns results in a higher ensemble loss on the validation and test sets, and, as a result, slightly more volatile returns on the out-of-sample ensemble portfolios, with mean returns virtually unchanged. The results of asset pricing tests are similar to those reported in the main text.

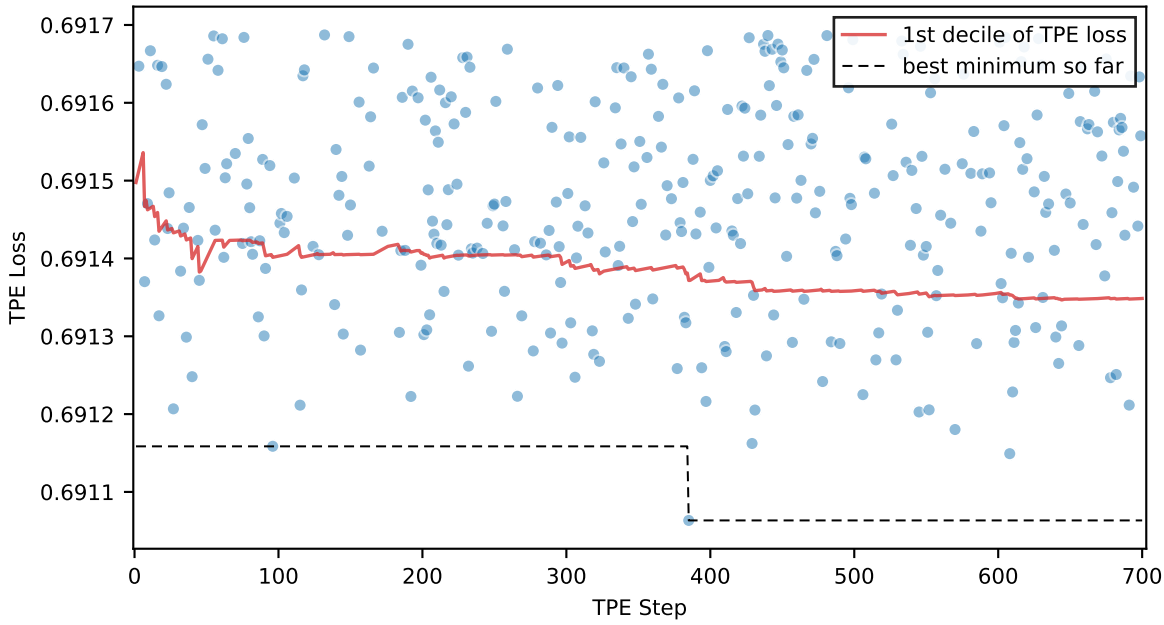
I start with repeating hyperparameter and ensemble optimization procedures outlined in Section III of the main text using the same hyperparameter priors for the first and second passes of the TPE optimization. Figures 3 and 4 report the progress of the TPE algorithm for the first and second passes respectively. Figure 5 reports the loss of the optimized ensemble and Figure 6 plots the ensemble's loss on the test set. Figure 7 plots the realized return as a function of the predicted probability (left panel) and distribution of the predicted probabilities (right panel). Table V reports descriptive statistics of median, quintile and decile portfolios sorted by the predicted probability, for the samples of largest 500 stocks with equal weights (Panel A) and of all CRSP stocks with equal and value weights (Panels B, C). Figures 8, 9 and 8 plot out-of-sample performance of the ensemble portfolios for the corresponding samples. Finally, Table VI repeats analysis of Table VI reporting results of time-series asset pricing tests for the seven-factor model (five factors of Fama and French (2015), momentum and the SDF portfolio of Chen et al. (2019)).

Figure 3: TPE hyperparameter optimization progress



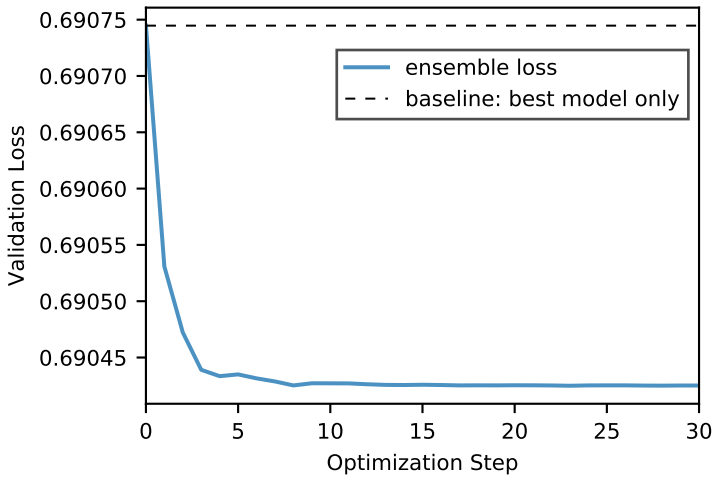
This figure depicts progress of the TPE hyperparameter optimization. The iterations and the TPE objective function values are along the horizontal and vertical axis respectively. The blue dots show the best 50% of the TPE evaluations, the solid red line is the expanding first decile of the TPE loss and the dashed black line tracks the best TPE loss at each iteration.

Figure 4: TPE hyperparameter optimization progress: deep architectures



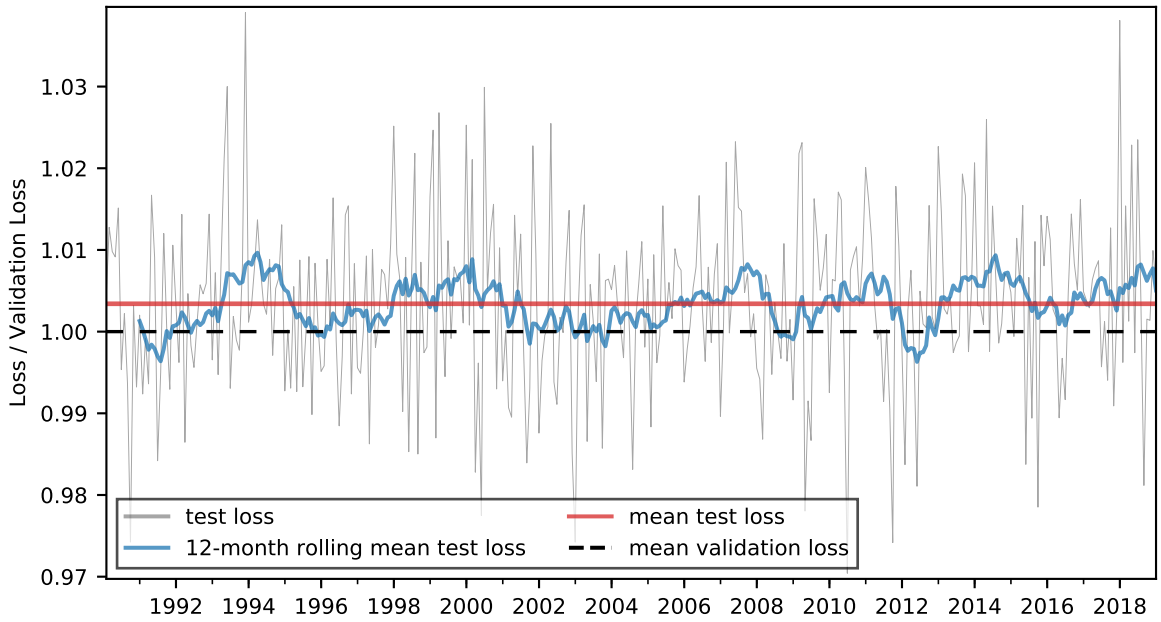
This figure depicts progress of the TPE hyperparameter optimization. The iterations and the TPE objective function values are along the horizontal and vertical axis respectively. The blue dots show the best 50% of the TPE evaluations, the solid red line is the expanding first decile of the TPE loss and the dashed black line tracks the best TPE loss at each iteration.

Figure 5: Validation loss during ensemble optimization



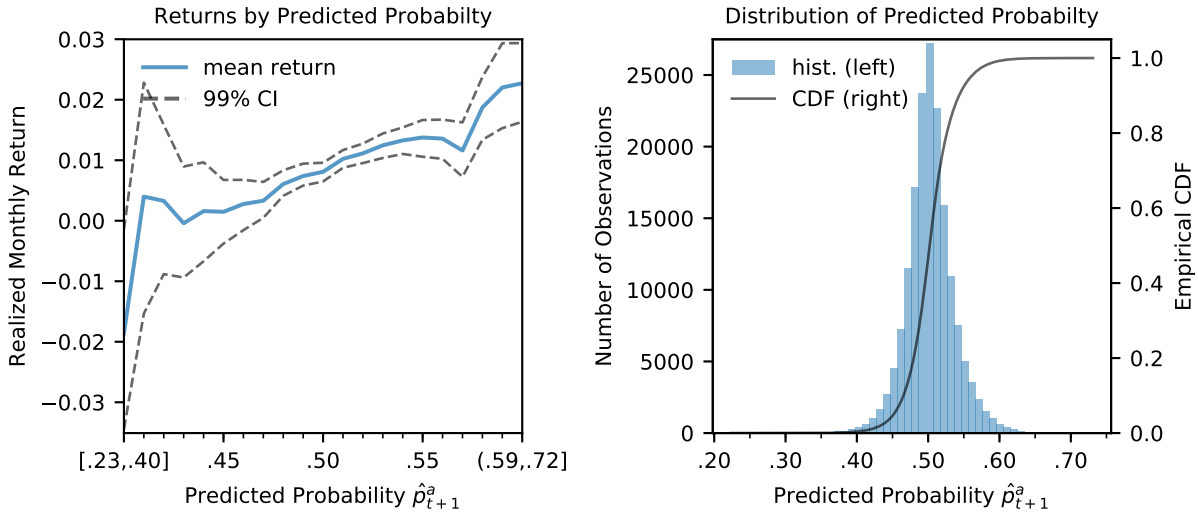
This figure depicts the validation loss of the ensemble over the course of the Caruana et al. (2004) optimization procedure. The black line shows the validation loss of the best model and the blue solid line plots the ensemble loss as optimization progresses.

Figure 6: Out-of-sample performance



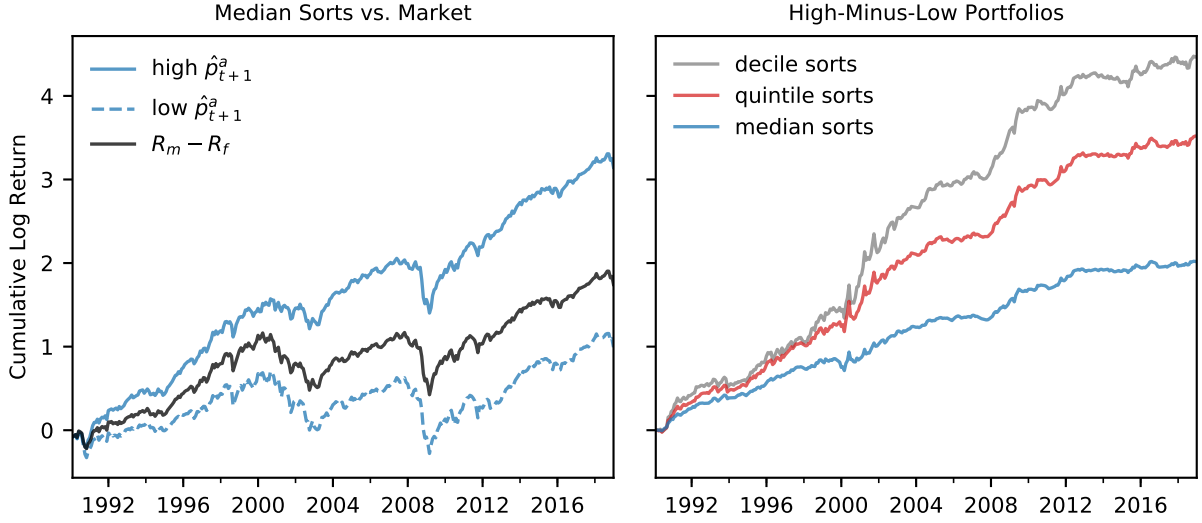
This figure displays out-of-sample performance of the neural network ensemble relative to its performance on the validation set (dashed black line normalized to 1). The red line depicts the average loss on the test set, the gray line shows cross-sectional average the loss at each month, and the blue line draws the 12-month rolling mean of this average.

Figure 7: Out-of-sample predicted probabilities and realized returns



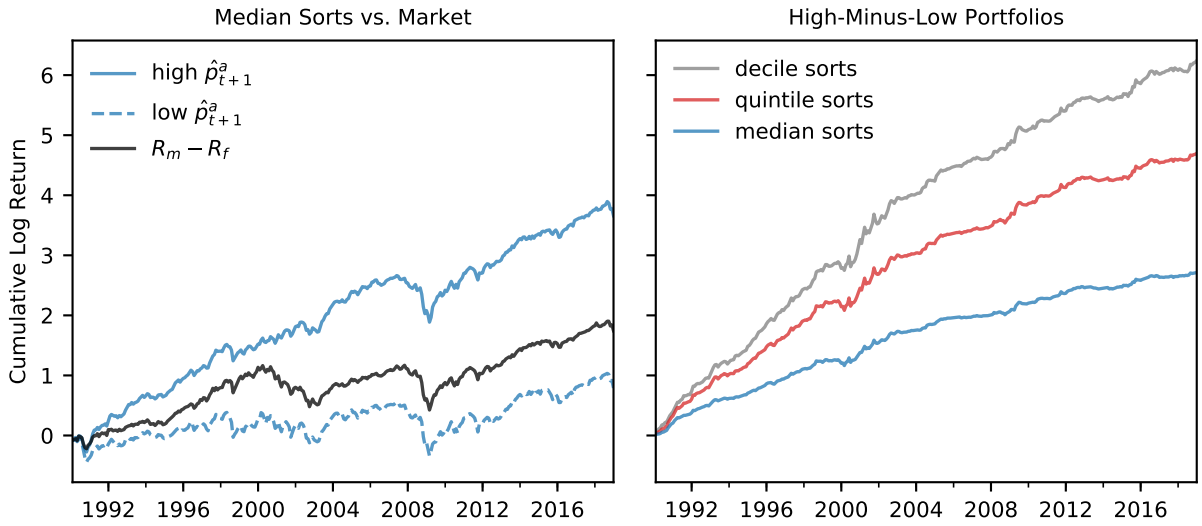
The left plot displays realized monthly return as a function of predicted probability of stock return being above the cross-sectional median return in the next month. Each (x, y) point shows the average return over all stocks and months for the estimated probability bin $\hat{p}^a_{t+1} \in (x - 0.01, x]$ except for the left- and right-most points which report the averages over extremes of the empirical distribution of predicted probabilities. The dashed black lines draw the corresponding 99% confidence bounds. The right plot shows the histogram of the predicted probabilities (in blue, against the left y-axis) and their empirical CDF (solid black line, against the right y-axis). The sample is from January 1990 to December 2018.

Figure 8: Return on predicted probability portfolios: largest 500 stocks, equal weights



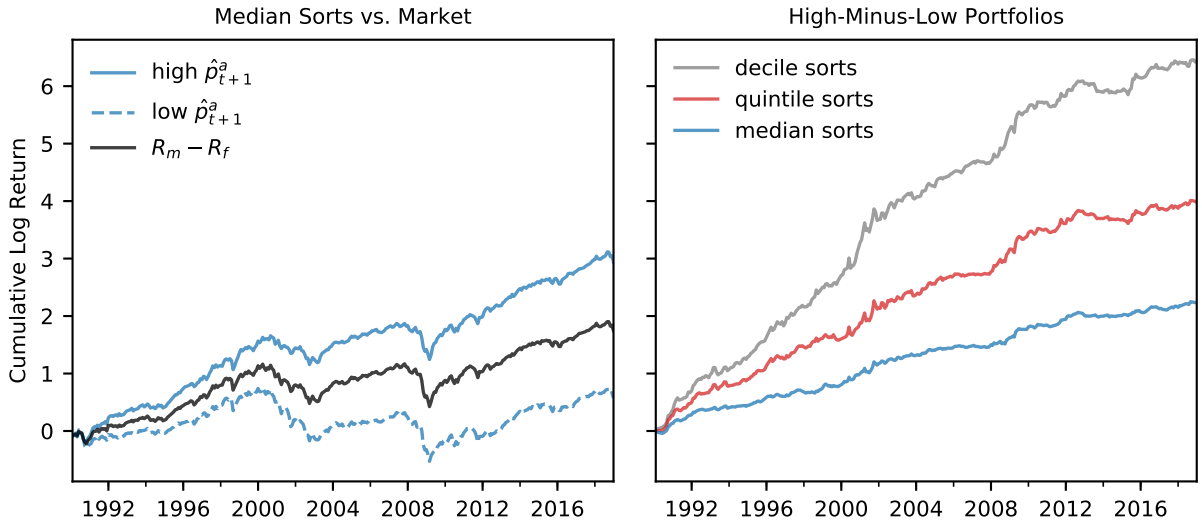
The figure plots out-of-sample performance of equally-weighted portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The left plot shows cumulative log excess returns on portfolios investing in stocks with 50% highest and lowest $\hat{p}^a_{t+1}$ (solid and dashed blue lines) and on the CRSP value weighted index (solid black line). The right plot displays cumulative log returns on long-short predicted probability portfolios: median, quintile, and decile sorts are drawn in blue, red, and gray respectively. All portfolios are rebalanced monthly. The sample is from January 1990 to December 2018 and the investment universe is the largest 500 stocks by market capitalization available at each rebalance date.

Figure 9: Return on predicted probability portfolios: all stocks, equal weights



The figure plots out-of-sample performance of equally-weighted portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The left plot shows cumulative log excess returns on portfolios investing in stocks with 50% highest and lowest $\hat{p}^a_{t+1}$ (solid and dashed blue lines) and on the CRSP value weighted index (solid black line). The right plot displays cumulative log returns on long-short predicted probability portfolios: median, quintile, and decile sorts are drawn in blue, red, and gray respectively. All portfolios are rebalanced monthly. The sample is from January 1990 to December 2018 and includes all CRSP stocks.

Figure 10: Return on predicted probability portfolios: all stocks, value weights



The figure plots out-of-sample performance of value-weighted portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The left plot shows cumulative log excess returns on portfolios investing in stocks with 50% highest and lowest $\hat{p}^a_{t+1}$ (solid and dashed blue lines) and on the CRSP value weighted index (solid black line). The right plot displays cumulative log returns on long-short predicted probability portfolios: median, quintile, and decile sorts are drawn in blue, red, and gray respectively. All portfolios are rebalanced monthly. The sample is from January 1990 to December 2018 and includes all CRSP stocks.

Table V: Descriptive statistics: long-short portfolios

	Median			Quintile			Decile		
Portfolio	1	2	2-1	1	5	5-1	1	10	10-1
Panel A: Largest 500 Stocks, Equal Weights									
Mean	4.79	12.00	7.21	1.31	14.17	12.87	−1.19	15.49	16.68
[t-stat]	[1.48]	[4.11]	[6.22]	[0.37]	[4.79]	[6.97]	[−0.29]	[4.80]	[6.39]
Median	12.29	17.38	4.11	8.34	16.19	9.51	2.58	18.74	12.01
Std	16.34	14.79	6.91	18.57	15.30	11.62	21.49	16.41	15.44
Skew	−0.69	−0.59	0.42	−0.60	−0.30	0.17	−0.66	−0.23	0.30
Kurtosis	1.66	2.03	4.05	2.10	2.08	4.26	2.60	2.33	3.65
Sharpe	0.29	0.81	1.04	0.07	0.93	1.11	−0.06	0.94	1.08
MaxDD	62.67	48.06	13.35	79.10	47.10	19.27	89.97	50.00	20.19
Max 1 M loss	21.00	19.14	8.47	24.02	19.10	16.35	28.35	21.62	19.95
Turnover	79.38	75.63	155.01	120.29	121.13	241.42	134.71	143.08	277.79
Panel B: All Stocks, Equal Weights									
Mean	4.37	13.94	9.57	−0.47	16.38	16.85	−4.43	18.15	22.58
[t-stat]	[1.25]	[4.28]	[8.40]	[−0.12]	[4.89]	[8.94]	[−1.05]	[5.18]	[9.50]
Median	10.54	16.73	7.48	6.26	19.46	12.31	3.05	22.45	17.67
Std	17.87	15.76	5.98	20.44	16.18	10.47	22.58	17.00	13.30
Skew	−0.77	−0.56	0.28	−0.65	−0.36	0.29	−0.55	−0.28	0.37
Kurtosis	1.75	1.86	2.84	1.75	1.96	3.01	1.84	2.15	2.97
Sharpe	0.24	0.88	1.60	−0.02	1.01	1.61	−0.20	1.07	1.70
MaxDD	59.44	53.99	9.53	78.04	52.53	14.93	91.91	55.53	16.27
Max 1 M loss	22.16	18.92	7.39	25.63	18.55	12.86	27.64	19.93	16.27
Turnover	78.85	75.12	153.96	121.35	118.89	240.24	139.75	140.03	279.79
Panel C: All Stocks, Value Weights									
Mean	3.28	11.23	7.96	−0.10	14.55	14.65	−5.45	18.34	23.79
[t-stat]	[1.04]	[3.90]	[6.28]	[−0.03]	[4.96]	[6.75]	[−1.30]	[5.76]	[7.65]
Median	10.87	14.78	6.32	4.48	15.97	10.69	1.65	19.58	19.00
Std	15.96	14.44	7.03	19.37	15.69	13.29	22.54	17.01	17.15
Skew	−0.93	−0.50	0.83	−0.73	−0.04	0.71	−0.67	0.19	0.58
Kurtosis	2.34	1.29	2.88	3.16	2.35	3.67	2.73	2.93	2.75
Sharpe	0.21	0.78	1.13	−0.01	0.93	1.10	−0.24	1.08	1.39
MaxDD	72.23	46.69	7.89	83.42	44.96	20.44	95.43	41.84	21.27
Max 1 M loss	22.66	15.66	6.22	27.18	14.69	12.84	29.31	17.94	16.43
Turnover	97.31	81.80	179.11	148.43	132.66	281.09	162.35	152.73	315.09

The table reports descriptive statistics of excess returns on portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. Each triplet of columns shows descriptive statistics of a low, high, and high-minus-low portfolio for median, quintile and decile sorts. Means, medians, standard deviations are in percent p.a., Sharpe ratios are annualized, maximum drawdowns and maximum 1-month losses are in percent, average turnover is in percent per month. The numbers in brackets are the Newey and West (1987) t-statistics for the null hypothesis of mean return being zero with number of lags according to Newey and West (1994). Panel A reports results for equally-weighted portfolios in the universe of the largest 500 stocks by market capitalization. Panels B and C report results for equally- and value-weighted portfolios respectively. The sample is from January 1990 to December 2018 and includes all CRSP stocks.

Table VI: Time-series asset pricing tests: FF-5 + MOM + Chen et al. (2019) SDF

	Top 500 MV			All CRSP EW			All CRSP VW		
Portfolio	α	SDF	$\bar{R}^2$	α	SDF	$\bar{R}^2$	α	SDF	$\bar{R}^2$
Panel A: Median Sorts									
P1	−2.54	−0.07	0.94	−4.33	0.09	0.97	−4.87	0.06	0.94
	[−2.20]	[−0.86]		[−3.78]	[0.99]		[−3.93]	[0.62]	
P2	3.81	−0.00	0.91	2.83	0.30	0.95	3.32	0.05	0.94
	[3.03]	[−0.04]		[2.51]	[4.43]		[2.87]	[0.52]	
HML	6.35	0.07	0.20	7.16	0.21	0.33	8.19	−0.01	0.04
	[3.42]	[0.51]		[4.12]	[1.67]		[3.57]	[−0.04]	
Panel B: Decile Sorts									
P1	−10.86	0.34	0.80	−12.19	0.18	0.89	−15.49	0.28	0.76
	[−3.14]	[1.19]		[−4.45]	[0.71]		[−4.03]	[0.91]	
P2	−4.23	−0.01	0.85	−5.36	0.09	0.94	−7.02	0.35	0.78
	[−2.80]	[−0.11]		[−3.61]	[0.70]		[−3.12]	[1.97]	
P3	−0.74	−0.21	0.88	−2.80	−0.02	0.96	−5.75	0.13	0.86
	[−0.51]	[−2.17]		[−2.26]	[−0.23]		[−3.27]	[0.96]	
P4	3.08	−0.38	0.89	−2.14	0.10	0.97	−3.18	−0.16	0.87
	[2.01]	[−4.60]		[−2.03]	[1.29]		[−2.04]	[−1.02]	
P5	−0.01	−0.08	0.90	0.83	0.09	0.96	−0.64	0.02	0.92
	[−0.01]	[−0.78]		[0.89]	[1.24]		[−0.65]	[0.18]	
P6	3.33	−0.23	0.88	2.25	0.07	0.96	1.67	0.02	0.89
	[2.17]	[−1.82]		[2.09]	[0.94]		[1.19]	[0.13]	
P7	3.20	−0.12	0.88	2.42	0.15	0.94	2.51	−0.21	0.85
	[2.07]	[−1.17]		[1.93]	[1.79]		[1.14]	[−1.06]	
P8	3.77	−0.10	0.86	1.75	0.31	0.93	2.72	0.23	0.84
	[2.46]	[−0.75]		[1.33]	[3.99]		[1.81]	[1.60]	
P9	3.45	0.11	0.83	2.98	0.36	0.91	3.37	0.06	0.81
	[2.15]	[0.95]		[2.15]	[4.19]		[1.93]	[0.40]	
P10	5.32	0.31	0.76	4.75	0.61	0.88	9.28	0.21	0.72
	[2.47]	[1.57]		[2.98]	[5.59]		[3.36]	[0.80]	
P10-P1	16.17	−0.02	0.27	16.94	0.43	0.34	24.77	−0.07	0.14
	[3.96]	[−0.08]		[4.44]	[1.46]		[4.59]	[−0.16]	

The table reports results of time-series regressions for excess returns on portfolios sorted by predicted probability $\hat{p}^a_{t+1}$ of stock return being above the cross-sectional median return in the next month. The test portfolios are across rows and explanatory variables include five Fama and French (2015) factors, momentum and SDF portfolio of Chen et al. (2019). Each triplet of columns reports pricing error in percent p.a., regression coefficient on the SDF portfolio and adjusted R^2 of the regression. The first triplet reports estimates for the large cap sample with test assets being equally-weighted portfolios constructed in the universe of five hundred largest stocks. The second triplet and third triplets repeat the exercise for the entire CRSP sample with equal- and value-weighted test portfolios respectively. Panels A and B report results for median and decile sorts respectively. The numbers in brackets are the Newey and West (1987) t-statistics with number of lags according to Newey and West (1994). The sample is from January 1992 to December 2016.