



Management Science

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

Deep Learning in Asset Pricing

Luyang Chen, Markus Pelger, Jason Zhu

To cite this article:

Luyang Chen, Markus Pelger, Jason Zhu (2023) Deep Learning in Asset Pricing. *Management Science*

Published online in Articles in Advance 20 Feb 2023

. <https://doi.org/10.1287/mnsc.2023.4695>

Full terms and conditions of use: <https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2023, INFORMS

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Deep Learning in Asset Pricing

Luyang Chen,^a Markus Pelger,^{b,*} Jason Zhu^b

^aInstitute for Computational and Mathematical Engineering, Stanford University, Stanford, California 94305; ^bDepartment of Management Science & Engineering, Stanford University, Stanford, California 94305

*Corresponding author

Contact: lych@alumni.stanford.edu (LC); mpelger@stanford.edu,  <https://orcid.org/0000-0001-7111-3588> (MP); jasonzhu@alumni.stanford.edu (JZ)

Received: September 15, 2020

Revised: July 23, 2021; January 21, 2022;
April 10, 2022

Accepted: May 19, 2022

Published Online in Articles in Advance:
February 20, 2023

<https://doi.org/10.1287/mnsc.2023.4695>

Copyright: © 2023 INFORMS

Abstract. We use deep neural networks to estimate an asset pricing model for individual stock returns that takes advantage of the vast amount of conditioning information, keeps a fully flexible form, and accounts for time variation. The key innovations are to use the fundamental no-arbitrage condition as criterion function to construct the most informative test assets with an adversarial approach and to extract the states of the economy from many macroeconomic time series. Our asset pricing model outperforms out-of-sample all benchmark approaches in terms of Sharpe ratio, explained variation, and pricing errors and identifies the key factors that drive asset prices.

History: Accepted by Agostino Capponi, finance.

Supplemental Material: The online appendix and data are available at <https://doi.org/10.1287/mnsc.2023.4695>.

Keywords: conditional asset pricing model • no arbitrage • stock returns • nonlinear factor model • cross-section of expected returns • machine learning • deep learning • big data • hidden states • GMM

1. Introduction

The fundamental question in asset pricing is to explain differences in average returns of assets. No-arbitrage pricing theory provides a clear answer: expected returns differ because assets have different exposure to the stochastic discount factor (SDF) or pricing kernel. The empirical quest in asset pricing for the last 40 years has been to estimate a stochastic discount factor that can explain the expected returns of all assets. There are four major challenges that the literature so far has struggled to overcome in a unified framework: First, the SDF could, by construction, depend on all available information, which means that the SDF is a function of a potentially very large set of variables. Second, the functional form of the SDF is unknown and likely complex. Third, the SDF can have a complex dynamic structure, and the risk exposure for individual assets can vary over time depending on economic conditions and changes in asset-specific attributes. Fourth, the risk premium of individual stocks has a low signal-to-noise ratio, which complicates the estimation of an SDF that explains the expected returns of all stocks.

In this paper, we estimate a general nonlinear asset pricing model with deep neural networks for all U.S. equity data based on a substantial set of macroeconomic and firm-specific information. Our crucial innovation is the use of the no-arbitrage condition as part of the neural network algorithm. We estimate the stochastic discount factor that explains all stock returns

from the conditional moment constraints implied by no arbitrage. It is a natural idea to use machine learning techniques, such as deep neural networks, to deal with the high dimensionality and complex functional dependencies of the problem. However, machine learning tools are designed to work well for prediction tasks in a high signal-to-noise environment. As asset returns in efficient markets seem to be dominated by unforeseeable news, it is hard to predict their risk premia with off-the-shelf methods. We show how to build better machine learning estimators by incorporating economic structure. Including the no-arbitrage constraint in the learning algorithm significantly improves the risk premium signal and makes it possible to explain individual stock returns. Empirically, our general model outperforms out-of-sample the leading benchmark approaches and provides a clear insight into the structure of the pricing kernel and the sources of systematic risk.

Our model framework answers three conceptional key questions in asset pricing. (1) What is the functional form of the SDF based on the information set? Popular models, for example, the Fama–French five-factor model, impose that the SDF depends linearly on a small number of characteristics. However, the linear model seems to be misspecified, and the factor zoo suggests that there are many more characteristics with pricing information. Our model allows for a general nonparametric form with a large number of characteristics. (2) What are the right test assets? The conventional approach is to calibrate and

evaluate asset pricing models on a small number of pre-specified test assets, for example, the 25 size and book-to-market double-sorted portfolios of Fama and French (1992). However, an asset pricing model that can explain well those 25 portfolios does not need to capture the pricing information of other characteristic-sorted portfolios or individual stock returns. Our approach constructs in a data-driven way the most informative test assets that are the hardest to explain and identify the parameters of the SDF. (3) What are the states of the economy? Exposure and compensation for risk should depend on economic conditions. A simple way to capture those would be, for example, National Bureau of Economic Research (NBER) recession indicators. However, this is a very coarse set of information given the hundreds of macroeconomic time series with complex dynamics. Our model extracts a small number of state processes that are based on the complete dynamics of a large number of macroeconomic time series and are the most relevant for asset pricing.

Our estimation approach combines no-arbitrage pricing and three neural network structures in a novel way. Each network is responsible for solving one of the three key questions outlined. First, we can explain the general functional form of the SDF as a function of the information set using a feedforward neural network. Second, we capture the time variation of the SDF as a function of macroeconomic conditions with a recurrent long short-term memory (LSTM) network that identifies a small set of macroeconomic state processes. Third, a generative adversarial network constructs the test assets by identifying the portfolios and states with the most unexplained pricing information. These three networks are linked by the no-arbitrage condition that helps to separate the risk premium signal from the noise and serves as a regularization to identify the relevant pricing information.

Our paper makes several methodological contributions. First, we introduce a nonparametric adversarial estimation approach to finance and show that it can be interpreted as a data-driven way to construct informative test assets. Estimating the SDF from the fundamental no-arbitrage moment equation is conceptionally a generalized method of moment (GMM) problem. The conditional asset pricing moments imply an infinite number of moment conditions. Our generative adversarial approach provides a method to find and select the most relevant moment conditions from an infinite set of candidate moments. Second, we introduce a novel way to use neural networks to extract economic conditions from complex time series. We are the first to propose LSTM networks to summarize the dynamics of a large number of macroeconomic time series in a small number of economic states. More specifically, our LSTM approach aggregates a large-dimensional panel cross-sectionally in a small number of time series and extracts from those a nonlinear time series model. The

key element is that it can capture short- and long-term dependencies that are necessary for detecting business cycles. Third, we propose a problem formulation that can extract the risk premium in spite of its low signal-to-noise ratio. The no-arbitrage condition identifies the components of the pricing kernel that carry a high risk premia but have only a weak variation signal. Intuitively, most machine learning methods in finance¹ fit a model that can explain as much variation as possible, which is essentially a second moment object. The no-arbitrage condition is based on explaining the risk premia, which is based on a first moment. We can decompose stock returns into a predictable risk premium part and an unpredictable martingale component. Most of the variation is driven by the unpredictable component that does not carry a risk premium. When considering average returns, the unpredictable component is diversified away over time, and the predictable risk premium signal is strengthened. However, the risk premia of individual stocks are time varying, and an unconditional mean of stock returns might not capture the predictable component. Therefore, we consider unconditional means of stock returns instrumented with all possible combinations of firm-specific characteristics and macroeconomic information. This serves the purpose of pushing up the risk premium signal and taking into account the time variation in the risk premium.

Our adversarial estimation approach is economically motivated by the seminal work of Hansen and Jagannathan (1997). They show that estimating an SDF that minimizes the largest possible pricing error is closest to an admissible true SDF in a least square distance. Our generative adversarial network builds on this idea and creates characteristic managed portfolios with the largest pricing errors for a candidate SDF, which are then used to estimate a better SDF. Our approach also builds on the insight of Bansal and Viswanathan (1993), who propose the use of neural networks as nonparametric estimators for the SDF from a given set of moment equations. Hence, the adversarial network element of our paper combines ideas of Hansen and Jagannathan (1997) and Bansal and Viswanathan (1993).

Our empirical analysis is based on a data set of all available U.S. stocks from the Center for Research in Security Prices (CRSP) with monthly returns from 1967 to 2016 combined with 46 time-varying, firm-specific characteristics and 178 macroeconomic time series. It includes the most relevant pricing anomalies and forecasting variables for the equity risk premium. Our approach outperforms out-of-sample all other benchmark approaches, which include linear models and deep neural networks that forecast risk premia instead of solving a GMM-type problem. We compare the models out-of-sample with respect to the Sharpe ratio implied by the pricing kernel, the explained variation, and explained average returns of individual stocks.

Our model has an annual out-of-sample Sharpe ratio of 2.6 compared with 1.7 for the linear special case of our model, 1.5 for the deep learning forecasting approach, and 0.8 for the Fama–French five-factor model. At the same time, we can explain 8% of the variation of individual stock returns and 23% of the expected returns of individual stocks, which is substantially higher than the other benchmark models. On standard test assets based on single- and double-sorted anomaly portfolios, our asset pricing model reveals an unprecedented pricing performance. In fact, on all 46 anomaly-sorted decile portfolios, we achieve a cross-sectional R^2 higher than 90%.

Our empirical main findings are threefold. First, we show and quantify the relevance of estimating machine learning models from the fundamental conditional no-arbitrage moments. This is possible through our adversarial GMM approach, which combines a pricing error objective with the selection of informative test assets. Even a flexible asset pricing model can only capture the asset pricing information that is included in the test assets on which it is calibrated. A flexible asset pricing model estimated on the optimal test assets constructed by the adversarial network has a 20% higher Sharpe ratio than one calibrated on individual stock returns without characteristic managed portfolios. By explaining the most informative test assets, we achieve a superior pricing performance on conventional sorted portfolios, for example, size and book-to-market single- or double-sorted portfolios. In fact, our model has an excellent pricing performance on all 46 anomaly-sorted decile portfolios. We confirm the Gu et al. (2020) insight that deep neural networks can explain more structure in stock returns because of their ability to fit flexible functional forms with many covariates. However, when used for asset pricing, off-the-shelf simple machine learning models that use a prediction instead of a pricing error objective can perform worse than even linear no-arbitrage models. It is the crucial innovation to combine a pricing objective with informative test assets and allow for a flexible model that allows us to detect the underlying SDF structure.

Second, macroeconomic states matter. Macroeconomic time series data have a low dimensional “factor” structure, which can be captured by four hidden state processes. The SDF structure depends on these economic states that are closely related to business cycles and times of economic crises. In order to find these states, we need to take into account the full time series dynamics of all macroeconomic variables. The conventional approach to deal with nonstationary macroeconomic time series is to use differenced data that capture changes in the time series. However, using only the last change as an input loses all dynamic information and renders the macroeconomic time series essentially useless. Even worse, prediction based on only the last change in a large panel

of macroeconomic variables leads to worse performance than leaving them out overall because they have lost most of their informational content, thus making it harder to separate the signal from the noise.

Third, we confirm that nonlinear and interaction effects matter as pointed out, among others, by Gu et al. (2020) and Bryzgalova et al. (2019). Our finding is more subtle and also explains why linear models, which are the workhorse models in asset pricing, perform so well. We find that, when considering firm-specific characteristics in isolation, the SDF depends approximately linearly on most characteristics. Thus, specific linear risk factors work well on certain single-sorted portfolios. The strength of the flexible functional form of deep neural networks reveals itself when considering the interaction between several characteristics. Although in isolation firm characteristics have a close to linear effect on the SDF, the multidimensional functional form is complex. Linear models and also nonlinear models that assume an additive structure in the characteristics (for example, additive splines or kernels) rule out interaction effects and cannot capture this structure.

Our conceptional framework is complementary to multifactor models. Multifactor models are based on the assumption that the SDF is spanned by those factors. We provide a unified framework to construct the SDF in a conditional multifactor model, which, in general, does not coincide with the unconditional mean–variance efficient combination of the factors. We combine the general conditional multifactor model of Kelly et al. (2019) with our model. We show that using the additional economic structure of spanning the SDF with instrumented principal components analysis (IPCA) factors and combining it with our SDF framework can lead to an even better asset pricing model.

Our findings are robust to the time periods under consideration, small capitalization stocks, the choice of the tuning parameters, and limits to arbitrage. The SDF structure is surprisingly stable over time. We estimate the functional form of the SDF with data from 1967 to 1986, which has an excellent out-of-sample performance for the test data from 1992 to 2016. The risk exposure to the SDF for individual stocks varies over time because the firm-specific characteristics and macroeconomic variables are time varying, but the functional form of the SDF and the risk exposure with respect to these covariates does not change. When allowing for a time-varying functional form by estimating the SDF on a rolling window, we find that it is highly correlated with the benchmark SDF and only leads to minor improvements. The estimation is robust to the choice of the tuning parameters. All of the best performing models selected on the validation data capture essentially the same asset pricing model. Our asset pricing model also performs well after excluding small and illiquid stocks from the test assets or the SDF.

1.1. Related Literature

Our paper contributes to an emerging literature that uses machine learning methods for asset pricing. In their pioneering work Gu et al. (2020) conduct a comparison of machine learning methods for predicting the panel of individual U.S. stock returns and demonstrate the benefits of flexible methods. Their estimates of the expected risk premia of stocks map into a cross-sectional asset pricing model. We use their best prediction model based on deep neural networks as a benchmark model in our analysis. We show that including the no-arbitrage constraint leads to better results for asset pricing and explained variation than a simple prediction approach. Furthermore, we clarify that it is essential to identify the dynamic pattern in macroeconomic time series before feeding them into a machine learning model, and we are the first to do this in an asset pricing context. Messmer (2017) and Feng et al. (2018) follow a similar approach as Gu et al. (2020) to predict stock returns with neural networks. Bianchi et al. (2021) provide a comparison of machine learning methods for predicting bond returns in the spirit of Gu et al. (2020).² Freyberger et al. (2020) use Lasso selection methods to estimate the risk premia of stock returns as a nonlinear additive function of characteristics. Feng et al. (2019) impose a no-arbitrage constraint by using a set of prespecified linear asset pricing factors and estimate the risk loadings with a deep neural network. Gu et al. (2021) extend the linear conditional factor model of Kelly et al. (2019) to a nonlinear factor model using an autoencoder neural network.³ We confirm their crucial insight that imposing economic structure on a machine learning algorithm can substantially improve the estimation. Bryzgalova et al. (2019) use decision trees to build a cross-section of asset returns, that is, a small set of basis assets that captures the complex information contained in a given set of stock characteristics and spans the conditional SDF. Their asset pricing trees generalize the concept of conventional sorting and are pruned by a novel dimension-reduction approach based on no-arbitrage arguments.

Our work is complementary to the literature on machine learning investment. Our estimated SDF directly maps into a conditionally mean-variance efficient portfolio. Avramov et al. (2022) raise the concern that the performance of machine learning portfolios could deteriorate in the presence of trading frictions.⁴ We discuss how their important insight can be taken into account when constructing machine learning investment portfolios. Cong et al. (2020) introduce an alternative reinforcement learning framework to optimize general portfolio objectives that also allows for trading frictions. They show the benefits of directly optimizing a trading objective and allowing for a time- and state-varying allocation rule through reinforcement learning. Furthermore, they also incorporate dynamic information from sequences of characteristics. Guijarro-Ordonez et al. (2019) design optimal statistical

arbitrage strategies under trading friction constraints with neural networks specialized for time series data.

The workhorse models in equity asset pricing are based on linear factor models exemplified by Fama and French (1993, 2015). Recently, new methods have been developed to study the cross-section of returns in the linear framework but accounting for the large amount of conditioning information. Lettau and Pelger (2020) extend principal component analysis (PCA) to account for no arbitrage. They show that a no-arbitrage penalty term makes it possible to overcome the low signal-to-noise ratio problem in financial data and find the information that is relevant for the pricing kernel. Our paper is based on a similar intuition, and we show that this result extends to a nonlinear framework. Kozak et al. (2020) estimate the SDF based on characteristic sorted factors with a modified elastic net regression.⁵ Kelly et al. (2019) apply PCA to stock returns projected on characteristics to obtain a conditional multifactor model in which the loadings are linear in the characteristics. Pelger (2020) combines high-frequency data with PCA to capture nonparametrically the time variation in factor risk. Pelger and Xiong (2021b) show that macroeconomic states are relevant to capture time variation in PCA-based factors.

Our approach uses a similar insight as Bansal and Viswanathan (1993) and Chen and Ludvigson (2009), who propose using a given set of conditional GMM equations to estimate the SDF with neural networks but restrict themselves to a small number of conditioning variables. In order to deal with the infinite number of moment conditions, we extend the classic GMM setup of Hansen (1982) and Chamberlain (1987) by an adversarial network to select the optimal moment conditions. Independently, Lewis and Syrgkanis (2018) propose to use an adversarial approach to solve a conditional method of moment problem. They focus on the theory of identification, whereas we show how to use this idea conceptually and empirically in the context of asset pricing.⁶ Our problem is also similar in spirit to the Wasserstein generative adversarial network (GAN) in Arjovsky et al. (2017) that provides a robust fit to moments. The GAN approach was first proposed by Goodfellow et al. (2014) for image recognition. In order to find the hidden states in macroeconomic time series, we propose the use of recurrent neural networks (RNNs) with LSTM. LSTMs are designed to find patterns in time series data and were first proposed by Hochreiter and Schmidhuber (1997).⁷ They are among the most successful commercial AIs and are heavily used for sequences of data, such as speech (e.g., Google with speech recognition for Android, Apple with Siri, and the “QuickType” function on the iPhone or Amazon with Alexa).

The rest of the paper is organized as follows. Section 2 introduces the model framework, and Section 3 elaborates on the estimation approach. The empirical results

are collected in Section 4. Section 5 concludes. The implementation details are delegated to the appendix, whereas the online appendix collects an overview of conditional SDF models, the application of the adversarial perspective to multifactor models, empirical robustness results, and a simulation.

2. Model

2.1. No-Arbitrage Asset Pricing

Our goal is to explain the differences in the cross-section of returns R for individual stocks. Let $R_{t+1,i}$ denote the return of asset i at time $t+1$. The fundamental no-arbitrage assumption is equivalent to the existence of a strictly positive SDF M_{t+1} such that, for any return in excess of the risk-free rate $R_{t+1,i}^e = R_{t+1,i} - R_{t+1}^f$, it holds

$$\begin{aligned}\mathbb{E}_t[M_{t+1}R_{t+1,i}^e] &= 0 \Leftrightarrow \mathbb{E}_t[R_{t+1,i}^e] \\ &= \underbrace{\left(-\frac{\text{Cov}_t(R_{t+1,i}^e, M_{t+1})}{\text{Var}_t(M_{t+1})}\right)}_{\beta_{t,i}} \cdot \underbrace{\frac{\text{Var}_t(M_{t+1})}{\mathbb{E}_t[M_{t+1}]}}_{\lambda_t},\end{aligned}$$

where $\beta_{t,i}$ is the exposure to systematic risk, λ_t is the price of risk, and $E_t[\cdot]$ denotes the expectation conditional on the information at time t . The SDF is an affine transformation of the tangency portfolio.⁸ Without loss of generality we consider the SDF formulation

$$M_{t+1} = 1 - \sum_{i=1}^N \omega_{t,i} R_{t+1,i}^e = 1 - \omega_t^\top R_{t+1}^e.$$

The fundamental pricing equation $\mathbb{E}_t[R_{t+1}^e M_{t+1}] = 0$ implies the SDF weights

$$\omega_t = \mathbb{E}_t[R_{t+1}^e R_{t+1}^{e \top}]^{-1} \mathbb{E}_t[R_{t+1}^e], \quad (1)$$

which are the portfolio weights of the conditional mean–variance efficient portfolio.⁹ We define the tangency portfolio as $F_{t+1} = \omega_t^\top R_{t+1}^e$ and refer to this traded factor as the SDF. The asset pricing equation can now be formulated as

$$\mathbb{E}_t[R_{t+1,i}^e] = \frac{\text{Cov}_t(R_{t+1,i}^e, F_{t+1})}{\text{Var}_t(F_{t+1})} \cdot \mathbb{E}_t[F_{t+1}] = \beta_{t,i} \mathbb{E}_t[F_{t+1}].$$

Hence, no arbitrage implies a one-factor model

$$R_{t+1,i}^e = \beta_{t,i} F_{t+1} + \epsilon_{t+1,i}$$

with $\mathbb{E}_t[\epsilon_{t+1,i}] = 0$ and $\text{Cov}_t(F_{t+1}, \epsilon_{t+1,i}) = 0$. Conversely, the factor model formulation implies the stochastic discount factor formulation. Furthermore, if the idiosyncratic risk $\epsilon_{t+1,i}$ is diversifiable and the factor F is systematic,¹⁰ then knowledge of the risk loadings is sufficient to construct the SDF:

$$(\beta_t^\top \beta_t)^{-1} \beta_t^\top R_{t+1}^e = F_{t+1} + (\beta_t^\top \beta_t)^{-1} \beta_t^\top \epsilon_{t+1} = F_{t+1} + o_p(1).$$

The fundamental problem is to find the SDF portfolio weights ω_t and risk loadings β_t . Both are time varying

and general functions of the information set at time t . The knowledge of ω_t and β_t solves three problems: (1) We can explain the cross-section of individual stock returns. (2) We can construct the conditional mean–variance efficient tangency portfolio. (3) We can decompose stock returns into their predictable systematic component and their nonsystematic unpredictable component.

Whereas Equation (1) gives an explicit solution for the SDF weights in terms of the conditional second and first moment of stock returns, it becomes infeasible to estimate without imposing strong assumptions. Without restrictive parametric assumptions, it is practically not possible to estimate reliably the inverse of a large-dimensional conditional covariance matrix for thousands of stocks. Even in the unconditional setup, the estimation of the inverse of a large-dimensional covariance matrix is already challenging. In the next section, we introduce an adversarial problem formulation that allows us to sidestep the problem of solving explicitly an infeasible conditional mean–variance optimization.

2.2. Generative Adversarial Methods of Moments

Finding the SDF weights is equivalent to solving a method of moment problem.¹¹ The conditional no-arbitrage moment condition implies infinitely many unconditional moment conditions

$$\mathbb{E}[M_{t+1} R_{t+1,i}^e g(I_t, I_{t,i})] = 0 \quad (2)$$

for any function $g(\cdot) : \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}^D$, where $I_t \times I_{t,i} \in \mathbb{R}^p \times \mathbb{R}^q$ denotes all the variables in the information set at time t and D is the number of moment conditions. We denote by I_t all p macroeconomic conditioning variables that are not asset-specific, for example, inflation rates or the market return, whereas $I_{t,i}$ are q firm-specific characteristics, for example, the size or book-to-market ratio of firm i at time t . The unconditional moment conditions can be interpreted as the pricing errors for a choice of portfolios and times determined by $g(\cdot)$. The challenge lies in finding the relevant moment conditions to identify the SDF.

A well-known formulation includes 25 moments that correspond to pricing the 25 size and value double-sorted portfolios of Fama and French (1992). For this special case, each g corresponds to an indicator function if the size and book-to-market values of a company are in a specific quantile. Another special case is to consider only unconditional moments, that is, setting g to a constant. This corresponds to minimizing the unconditional pricing error of each stock.

The SDF portfolio weights $\omega_{t,i} = \omega(I_t, I_{t,i})$ and risk loadings $\beta_{t,i} = \beta(I_t, I_{t,i})$ are general functions of the information set, that is, $\omega : \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}$ and $\beta : \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}$. For example, the SDF weights and loadings in the Fama–French three-factor model are a special case in which both functions are approximated by a two-

dimensional kernel function that depends on the size and book-to-market ratio of firms. The Fama–French three-factor model only uses firm-specific information but no macroeconomic information; for example, the loadings cannot vary based on the state of the business cycle.

We use an adversarial approach to select the moment conditions that lead to the largest mispricing:

$$\min_{\omega} \max_g \frac{1}{N} \sum_{j=1}^N \left\| \mathbb{E} \left[\left(1 - \sum_{i=1}^N \omega(I_t, I_{t,i}) R_{t+1,i}^e \right) R_{t+1,j}^e g(I_t, I_{t,j}) \right] \right\|^2, \quad (3)$$

where the function ω and g are normalized functions chosen from a specified functional class. This is a minimax optimization problem. These types of problems can be modeled as a zero-sum game in which one player, the asset pricing modeler, wants to choose an asset pricing model, whereas the adversary wants to choose conditions under which the asset pricing model performs badly. This can be interpreted as first finding portfolios or times that are the most mispriced and then correcting the asset pricing model to also price these assets. The process is repeated until all pricing information is taken into account; that is, the adversary cannot find portfolios with large pricing errors. Note that this is a data-driven generalization for the research protocol conducted in asset pricing in the last decades. Assume that the asset pricing modeler uses the Fama–French five-factor model; that is, M is spanned by those five factors. The adversary might propose momentum-sorted test assets; that is, g is a vector of indicator functions for different quantiles of past returns. As these test assets have significant pricing errors with respect to the Fama–French five factors, the asset pricing modeler needs to revise the candidate SDF, for example, by adding a momentum factor to M . Next, the adversary searches for other mispriced anomalies or states of the economy, which the asset pricing modeler exploits in the SDF model.

Our adversarial estimation with a minimax objective function is economically motivated and based on the insights of Hansen and Jagannathan (1997). They show that, if the SDF implied by an asset pricing model is only a proxy that does not price all possible assets in the economy, then minimizing the largest possible pricing error corresponds to estimating the SDF that is the closest to an admissible true SDF in a least square distance.¹² In our case, the SDF is implicitly constrained by the fact that it can only depend on stock-specific characteristics $I_{i,t}$ but not the identity of the stocks themselves and by a regularization in the estimation as specified in Section 3.5. Hence, even in-sample, the SDF has nonzero pricing errors for some stocks and their characteristic managed

portfolios, which puts us into the setup of Hansen and Jagannathan (1997).

Choosing the conditioning function g corresponds to finding optimal instruments in a GMM estimation. The conventional GMM approach assumes a finite number of moments that identify a finite dimensional set of parameters. The moments are selected to achieve the most efficient estimator within this class. Nagel and Singleton (2011) use this argument to build optimal managed portfolios for a particular asset pricing model. Their approach assumes that the set of candidate test assets identifies all the parameters of the SDF, and they can, therefore, focus on which test asset provides the most efficient estimator. Our problem is different in two ways that rule out using the same approach. First, we have an infinite number of candidate moments without the knowledge of which moments identify the parameters. Second, our parameter set is also of infinite dimension, and we, consequently, do not have an asymptotic normal distribution with a feasible estimator of the covariance matrix. In contrast, our approach selects the moments based on robustness.¹³ By controlling the worst possible pricing error, we aim to choose the test assets that can identify all parameters of the SDF and provide a robust fit. Hansen and Jagannathan (1997) discuss the estimation of the SDF based on the minimax objective function and compare it with the conventional efficient GMM estimation for parametric models with a low-dimensional parameter set. They conclude that the minimax estimation has desirable properties when models are misspecified, and the resulting SDFs have substantially less variation than with the conventional GMM approach.

Our conditioning function g generates a very large number of test assets to identify a complex SDF structure. The cross-sectional average is taken over the moment deviations, that is, pricing errors for $R_{t+1,j}^e g(I_t, I_{t,j})$, instead of considering the moment deviation for the D time series $\frac{1}{N} \sum_{i=1}^N R_{t+1,i}^e g(I_t, I_{t,j})$, which corresponds to the traditionally characteristic managed test assets. In the second case, we only have D test assets, whereas our approach yields $D \cdot N$ test assets. Note that there is no gain from using only D portfolios as the average squared moment deviations of all $D \cdot N$ instrumented stocks provide an upper bound for the pricing errors of the D portfolios. However, it turns out that the use of the larger number of test assets substantially accelerates the convergence, and the minimax problem can empirically already converge after three steps.¹⁴

Our objective function in Equation (3) uses only an approximate arbitrage condition in the sense of Ross (1976) and Chamberlain and Rothschild (1983). Our moment conditions are averaged over the sample of all instrumented stocks; that is, the objective is $1/N \sum_{i=1}^N \sum_{d=1}^D \alpha_{i,d}^2$,

where the moment deviation $\alpha_{i,d} = E[M_{t+1}R_i g_d(I_t, I_{t,i})]$ can be interpreted as the pricing error of stock i instrumented by the element g_d of the vector-valued function $g(\cdot)$. Note that the instruments g_d are normalized to be in $[-1, 1]$. In our benchmark model, we consider $n = 10,000$ stocks and $D = 8$ instruments and, therefore, average in total more than 80,000 instrumented assets. Hence, our SDF depends only on information that affects a very large proportion of the stocks, that is, systematic mispricing. This also implies that the adversarial approach only selects instruments that lead to mispricing for most stocks.

Once we obtain the SDF factor weights, the loadings are proportional to the conditional moments $E_t[F_{t+1} R_{t+1,i}^e]$. A key element of our approach is to avoid estimating directly conditional means of stock returns. Our empirical results show that we can better estimate the conditional comovement of stock returns with the SDF factors, which is a second moment, than the conditional first moment. Note, that, in the no-arbitrage one-factor model, the loadings are proportional to $\text{Cov}_t(R_{t+1,i}^e, F_{t+1})$ and $E_t[F_{t+1} R_{t+1,i}^e]$, where the last one has the advantage that we avoid estimating the first conditional moment.

2.3. Alternative Models

We consider two special cases as alternatives: One model can take a flexible functional form but does not use the no-arbitrage constraint in the estimation. The other model is based on the no-arbitrage framework but considers a linear functional form.

Instead of minimizing the violation of the no-arbitrage condition, one can directly estimate the conditional mean. Note that the conditional expected returns $\mu_{t,i}$ are proportional to the loadings in the one-factor formulation:

$$\mu_{t,i} := E_t[R_{t+1,i}^e] = \beta_{t,i} E_t[F_{t+1}].$$

Hence, up to a time-varying proportionality constant, the SDF weights and loadings are equal to $\mu_{t,i}$. This reduces the cross-sectional asset pricing problem to a simple forecasting problem. Hence, we can use the forecasting approach pursued in Gu et al. (2020) for asset pricing.

The second benchmark model assumes a linear structure in the factor portfolio weights $\omega_{t,i} = \theta^\top I_{t,i}$ and linear conditioning in the test assets:

$$\begin{aligned} & \frac{1}{N} \sum_{j=1}^N E \left[\left(1 - \frac{1}{N} \sum_{i=1}^N \theta^\top I_{t,i} R_{t+1,i}^e \right) R_{t+1,j}^e I_{t,j} \right] \\ &= 0 \Leftrightarrow E[(1 - \theta^\top \tilde{F}_{t+1}) \tilde{F}_{t+1}^\top] = 0, \end{aligned}$$

where $\tilde{F}_{t+1} = \frac{1}{N} \sum_{i=1}^N I_{t,i} R_{t+1,i}^e$ are q characteristic managed factors. Such characteristic managed factors based on linearly projecting onto quantiles of characteristics are exactly the input to PCA in Kelly et al. (2019) or the elastic net mean-variance optimization in Kozak et al.

(2020).¹⁵ The solution to minimizing the sum of squared errors in these moment conditions is a simple mean-variance optimization for the q characteristic managed factors; that is, $\theta = (\mathbb{E}[\tilde{F}_{t+1} \tilde{F}_{t+1}^\top])^{-1} \mathbb{E}[\tilde{F}_{t+1}]$ are the weights of the tangency portfolio based on these factors.¹⁶ We choose this specific linear version of the model as it maps directly into the linear approaches that are already successfully used in the literature. This linear framework essentially captures the class of linear factor models. The online appendix provides a detailed overview of the various models for conditional SDFs and their relationship to our framework.

3. Estimation

3.1. Loss Function and Model Architecture

The empirical loss function of our model minimizes the weighted sample moments, which can be interpreted as weighted sample mean pricing errors:

$$L(\omega | \hat{g}, I_t, I_{t,i}) = \frac{1}{N} \sum_{i=1}^N \frac{T_i}{T} \left\| \frac{1}{T_i} \sum_{t \in T_i} M_{t+1} R_{t+1,i}^e \hat{g}(I_t, I_{t,i}) \right\|^2. \quad (4)$$

This is for a given conditioning function $\hat{g}(\cdot)$ and information set. We deal with an unbalanced panel in which the number of time series observations T_i varies for each asset. As the convergence rates of the moments under suitable conditions is $1/\sqrt{T_i}$, we weight each cross-sectional moment condition by $\sqrt{T_i}/\sqrt{T}$, which assigns a higher weight to moments that are estimated more precisely and down-weights the moments of assets that are observed only for a short time period.

For a given conditioning function $\hat{g}(\cdot)$ and choice of information set, the SDF portfolio weights are estimated by a feedforward network that minimizes the pricing error loss

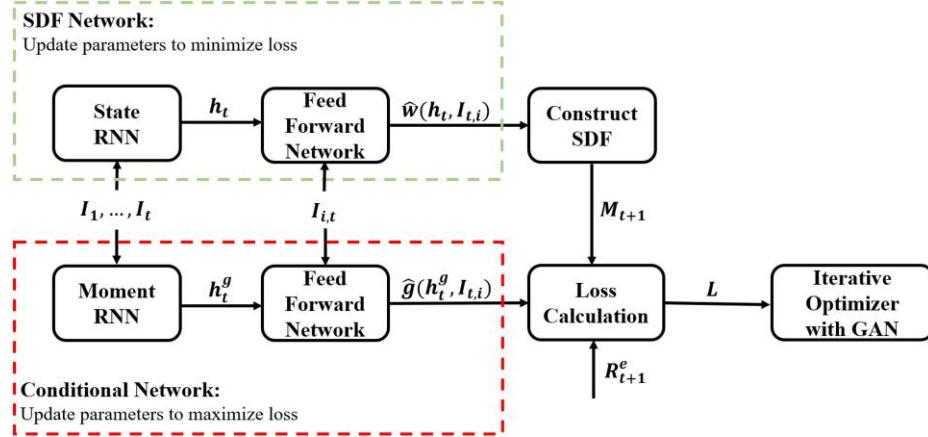
$$\hat{\omega} = \min_{\omega} L(\omega | \hat{g}, I_t, I_{t,i}).$$

We refer to this as the SDF network.

We construct the conditioning function \hat{g} via a conditional network with a similar neural network architecture. The conditional network serves as an adversary and competes with the SDF network to identify the assets and portfolio strategies that are the hardest to explain. The macroeconomic information dynamics are summarized by macroeconomic state variables h_t , which are obtained by an RNN with long short-term memory units. The model architecture is summarized in Figure 1, and each of the different components are described in detail in the next sections.

In contrast, forecasting returns similar to Gu et al. (2020) uses only a feedforward network and is labeled as FFN. It estimates conditional means $\mu_{t,i} = \mu(I_t, I_{t,i})$ by minimizing the average sum of squared prediction

Figure 1. (Color online) GAN Model Architecture



Notes. This figure shows the model architecture of GAN with RNN with LSTM cells. The SDF network has two parts: (1) An LSTM estimates a small number of macroeconomic states. (2) These states together with the firm characteristics are used in an FFN to construct a candidate SDF for a given set of test assets. The conditioning network also has two networks: (1) it creates its own set of macroeconomic states, (2) which it combines with the firm characteristics in an FFN to find mispriced test assets for a given SDF M . These two networks compete until convergence; that is, neither the SDF nor the test assets can be improved.

errors:

$$\hat{\mu} = \min_{\mu} \frac{1}{T} \sum_{t=1}^T \frac{1}{N_t} \sum_{i=1}^{N_t} \left(R_{t+1,i}^e - \mu(I_{t,i}) \right)^2.$$

We only include the best performing feedforward network from the Gu et al. (2020) comparison study. Within their framework, this model outperforms tree learning approaches and other linear and nonlinear prediction models. In order to make the results more comparable with Gu et al. (2020), we follow the same procedure as outlined in their paper. Thus, the simple forecasting approach does not include an adversarial network or LSTM to condense the macroeconomic dynamics.

3.2. FFN

An FFN¹⁷ is a flexible nonparametric estimator for a general functional relationship $y = f(x)$ between the covariates x and a variable y . In contrast to conventional nonparametric estimators, such as kernel regressions or splines, FFNs not only estimate nonlinear relationships, but are also designed to capture interaction effects between a large dimensional set of covariates. We consider four different FFNs: for the covariates $x = [I_t, I_{t,i}]$, we estimate (1) the optimal weights in our GAN network ($y = \omega$), (2) the optimal instruments for the moment conditions in our GAN network ($y = g$), (3) the conditional mean return ($y = \mathbb{E}_t[R_{t+1,i}^e]$), and (4) the second moment ($y = \mathbb{E}_t[R_{t+1,i}^e F_{t+1}]$) to obtain the SDF loadings $\beta_{t,i}$.

We start with a one-layer neural network as illustrated in Figure 2. It combines the original covariates $x = x^{(0)} \in \mathbb{R}^{K^{(0)}}$ linearly and applies a nonlinear transformation. This nonlinear transformation is based on an element-wise operating activation function. We choose the popular

function known as the rectified linear unit (ReLU),¹⁸ which component-wise thresholds the inputs and is defined as

$$\text{ReLU}(x_k) = \max(x_k, 0).$$

The result is the hidden layer $x^{(1)} = (x_1^{(1)}, \dots, x_{K^{(1)}}^{(1)})$ of dimension $K^{(1)}$, which depends on the parameters $W^{(0)} = (w_1^{(0)}, \dots, w_{K^{(0)}}^{(0)})$ and the bias term $w_0^{(0)}$. The output layer is simply a linear transformation of the output from the hidden layer:

$$x^{(1)} = \text{ReLU}(W^{(0)\top} x^{(0)} + w_0^{(0)}) = \text{ReLU}\left(w_0^{(0)} + \sum_{k=1}^{K^{(0)}} w_k^{(0)} x_k^{(0)}\right)$$

$$y = W^{(1)\top} x^{(1)} + w_0^{(1)} \quad \text{with } x^{(1)} \in \mathbb{R}^{K^{(1)}}, W^{(0)} \in \mathbb{R}^{K^{(1)} \times K^{(0)}},$$

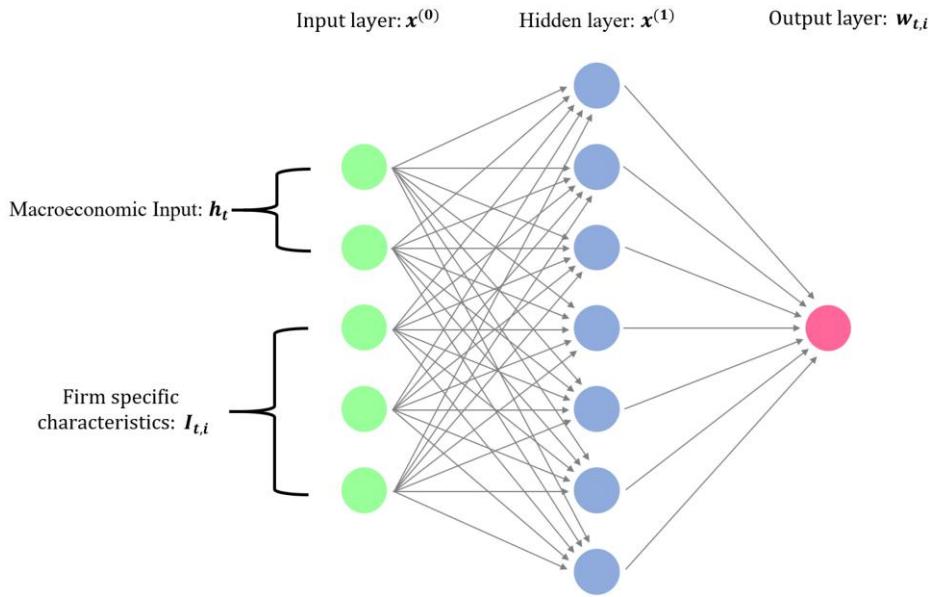
$$W^{(1)} \in \mathbb{R}^{K^{(1)}}.$$

Note that, without the nonlinearity in the hidden layer, the one-layer network reduces to a generalized linear model. A deep neural network combines several layers by using the output of one hidden layer as an input to the next hidden layer. The details are explained in Appendix A.1. The multiple layers allow the network to capture nonlinearities and interaction effects in a more parsimonious way.

3.3. RNN with LSTM

An RNN with LSTM estimates the hidden macroeconomic state variables. Instead of directly passing macroeconomic variables I_t as covariates to the feedforward network, we extract their dynamic patterns with a specific RNN and only pass on a small number of hidden states capturing these dynamics.

Figure 2. (Color online) Illustration of Feedforward Network with Single Hidden Layer



Many macroeconomic variables themselves are not stationary. Hence, we need to first perform transformations as suggested in McCracken and Ng (2016), which typically take the form of some difference of the time series. There is no reason to assume that the pricing kernel has a Markovian structure with respect to the macroeconomic information, in particular, after transforming them into stationary increments. For example, business cycles can affect pricing, but the gross domestic product growth of the last period is insufficient to learn if the model is in a boom or a recession. Hence, we need to include lagged values of the macroeconomic variables and find a way to extract the relevant information from a potentially large number of lagged values.

As an illustration, we show in Figure 3 three examples of the complex dynamics in the macroeconomic time series that we include in our empirical analysis. We plot the time series of the U.S. unemployment rate, the S&P 500 price, and the oil price together with the standard transformations proposed by McCracken and Ng (2016) to remove the obvious nonstationarities. Using only the last observation of the differenced data obviously results in a loss of information and cannot identify the cyclical dynamic patterns.

Formally, we have a sequence of stationary vector-valued processes $\{x_0, \dots, x_t\}$, where we set x_t to the stationary transformation of I_t at time t , that is, typically an increment. Our goal is to estimate a functional mapping h that transforms the time series x_t into “state processes” $h_t = h(x_0, \dots, x_t)$ for $t = 1, \dots, T$. The simplest transformation is to simply take the last increment, that is, $h_t^\Delta = h^\Delta(x_0, \dots, x_t) = x_t$. This approach is used in most papers, including Gu et al. (2020) and neglects the serial dependency structure in x_t .

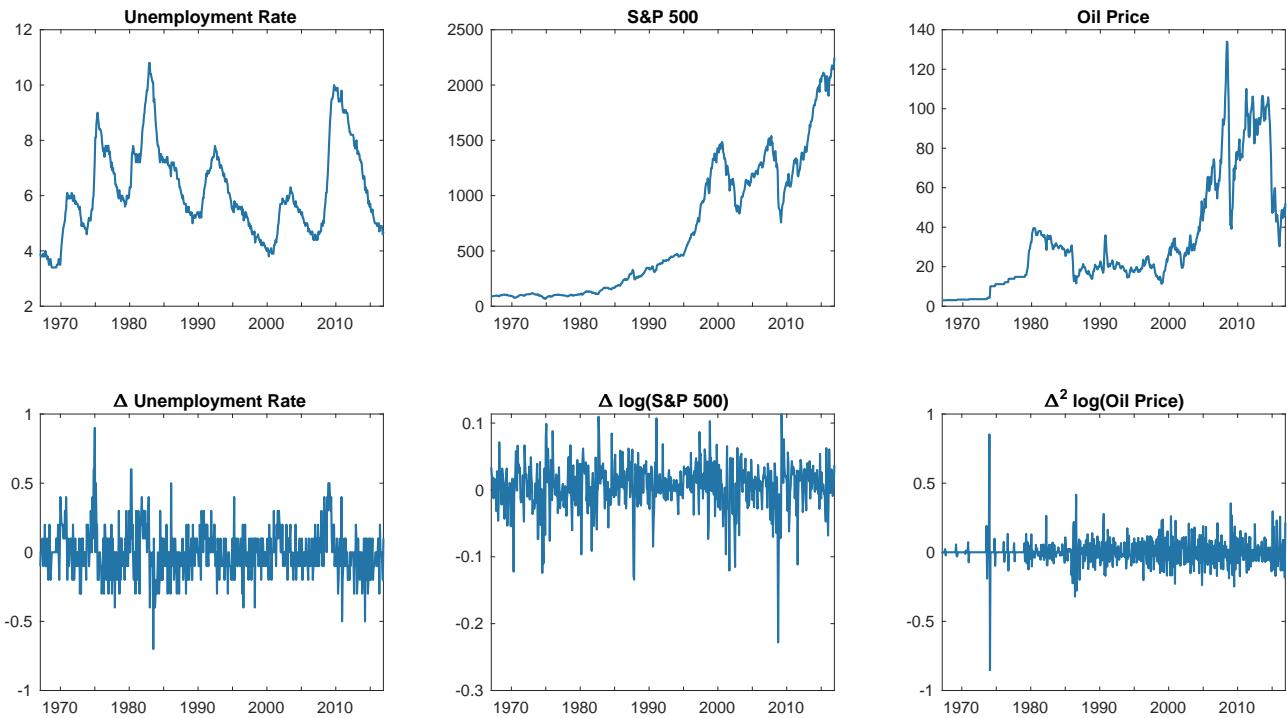
Macroeconomic time series variables are strongly cross-sectionally dependent; that is, there is redundant information that could be captured by some form of factor model. A cross-sectional dimension reduction is necessary as the number of time series observations in our macroeconomic panel is of a similar magnitude as the number of cross-sectional observations. Ludvigson and Ng (2007) advocate the use of PCA to extract a small number K_h of factors, that is, a special case of the function $h^{\text{PCA}}(x_0, \dots, x_t) = W_x x_t$ for $W_x \in \mathbb{R}^{p \times K_h}$. This aggregates the time series to a small number of latent factors that explain the correlation in the innovations in the time series, but PCA cannot identify the current state of the economic system that depends on the dynamics.

RNNs are a family of neural networks for processing sequences of data. They estimate nonlinear time series dependencies for vector-valued sequences in a recursive form. A vanilla RNN model takes the current input variable x_t and the previous hidden state h_{t-1}^{RNN} and performs a nonlinear transformation to get the current state h_t^{RNN} :

$$h_t^{\text{RNN}} = h^{\text{RNN}}(x_0, \dots, x_t) = \sigma(W_h h_{t-1}^{\text{RNN}} + W_x x_t + w_0),$$

where σ is the nonlinear activation function. Intuitively, a vanilla RNN combines two steps: First, it summarizes cross-sectional information by linearly combining a large vector x_t into a lower dimensional vector. Second, it is a nonlinear generalization of an autoregressive process in which the lagged variables are transformations of the lagged observed variables. This type of structure is powerful if only the immediate past is relevant, but it is not suitable if the time series dynamics are driven by

Figure 3. (Color online) Examples of Macroeconomic Variables



Note. This figure shows examples of macroeconomic time series with standard transformations proposed by McCracken and Ng (2016).

events that are further back in the past. Conventional RNNs can encounter problems with exploding and vanishing gradients when considering longer time lags. This is why we use the more complex long short-term memory cells. The LSTM is designed to deal with lags of unknown and potentially long duration in the time series, which makes it well-suited to detect business cycles.

Our LSTM approach can deal with both the large dimensionality of the system and a very general functional form of the states and allow for long-term dependencies. Appendix A.2 provides a detailed explanation of the estimation method. Intuitively, an LSTM uses different RNN structures to model short- and long-term dependencies and combines them with a nonlinear function. We can think of an LSTM as a flexible hidden state-space model for a large-dimensional system. On the one hand, it provides a cross-sectional aggregation similar to a latent factor model. On the other hand, it extracts dynamics similar in spirit to state-space models, for example, the simple linear Gaussian state-space model estimated by a Kalman filter. The strength of the LSTM is that it combines both elements in a general nonlinear model. In our simulation example in the online appendix, we illustrate that an LSTM can successfully extract a business cycle pattern that essentially captures deviations of a local mean from a long-term mean. Similarly, the state processes in our empirical analysis also seem to be based on the

relationship between the short- and long-term averages of the macroeconomic increments and, hence, represent a business cycle-type behavior.

The output of the LSTM is a small number of state processes $h_t = h^{\text{LSTM}}(x_0, \dots, x_t)$, which we use instead of the macroeconomic variables I_t as an input to our SDF network. Note, that each state h_t depends only on current and past macroeconomic increments and has no look-ahead bias.

3.4. GAN

The conditioning function g is the output of a second feedforward network. Inspired by GANs, we choose the moment conditions that lead to the largest pricing discrepancy by having two networks compete against each other. One network creates the SDF M_{t+1} , and the other network creates the conditioning function.

We take three steps to train the model. Our initial first step SDF minimizes the unconditional loss. Second, given this SDF, we maximize the loss by optimizing the parameters in the conditional network. Finally, given the conditional network, we update the SDF network to minimize the conditional loss.¹⁹ The logic behind this idea is that, by minimizing the largest conditional loss among all possible conditioning functions, the loss for any function is small. Note that both the SDF and conditional networks each use an FFN network combined with an LSTM that estimates the macroeconomic hidden state variables; that is, instead of

directly using I_t as an input, each network summarizes the whole macroeconomic time series information in the state process h_t (respectively, h_t^g for the conditional network). The two LSTMs are based on the criteria function of the two networks; that is, h_t are the hidden states that can minimize the pricing errors, whereas h_t^g generate the test assets with the largest mispricing.²⁰

$$\{\hat{\omega}, \hat{h}_t, \hat{g}, \hat{h}_t^g\} = \min_{\omega, h_t} \max_{g, h_t^g} L(\omega | \hat{g}, h_t^g, h_t, I_{t,i}).$$

3.5. Hyperparameters and Ensemble Learning

Because of the high dimensionality and nonlinearity of the problem, training a deep neural network is a complex task. Here, we summarize the implementation and provide additional details in Appendix A.3. We prevent the model from overfitting and deal with the large number of parameters by using “dropout,” which is a form of regularization that has generally better performance than conventional l_1/l_2 regularization. We optimize the objective function accurately and efficiently by employing an adaptive learning rate for a gradient-based optimization.

We obtain robust and stable fits by ensemble averaging over several fits of the models. A distinguishing feature of neural networks is that the estimation results can depend on the starting value used in the optimization. The standard practice, which is also used by Gu et al. (2020), is to train the models separately with different initial values chosen from an optimal distribution. Averaging over multiple fits achieves two goals: First, it diminishes the effect of a local suboptimal fit. Second, it reduces the estimation variance of the estimated model. All our neural networks, including the forecasting approach, are averaged over nine model fits.²¹ Let $\hat{\omega}^{(j)}$ and $\hat{\beta}^{(j)}$ be the optimal portfolio weights, respectively, SDF loadings given by the j th model fit. The ensemble model is an average of the outputs from models with the same architecture but different starting values for the optimization, that is, $\hat{\omega} = \frac{1}{9} \sum_{j=1}^9 \hat{\omega}^{(j)}$ and $\hat{\beta} = \frac{1}{9} \sum_{j=1}^9 \hat{\beta}^{(j)}$. Note that for vector-valued functions, for example, the conditioning function g and macroeconomic states h , it is not meaningful to report their model averages as different entries in the vectors are not necessarily reflecting the same object in each fit.

We split the data into training, validation, and testing samples. The validation set is used to tune the hyperparameters, and it includes the depth of the network (number of layers), the number of basis functions in each layer (nodes), the number of macroeconomic states, the number of conditioning instruments, and the structure of the conditioning network. We choose the best configuration among all possible combinations of hyperparameters by maximizing the Sharpe ratio of the SDF on the validation data.²² The optimal model is evaluated on

the test data. Our optimal model has two layers, four economic states, and eight instruments for the test assets. Our results are robust to the tuning parameters as discussed in Section 4.8. In particular, our results do not depend on the structure of the network, and the best performing networks on the validation data provide essentially an identical model with the same relative performance on the test data. The FFN for the forecasting approach uses the optimal hyperparameters selected by Gu et al. (2020). This has the additional advantage of making our results directly comparable to their results.

3.6. Model Comparison

We evaluate the performance of our model by calculating the Sharpe ratio of the SDF, the amount of explained variation and the pricing errors. We compare our GAN model with its linear special case, which is a linear factor model, and the deep-learning forecasting approach. The one factor representation yields three performance metrics to compare the different model formulations. First, the SDF is by construction on the globally efficient frontier and should have the highest conditional Sharpe ratio. We use the unconditional Sharpe ratio of the SDF portfolio $SR = \frac{\mathbb{E}[F]}{\sqrt{\text{Var}[F]}}$ as a measure to assess the pricing performance of models. The second metric measures the variation explained by the SDF. The explained variation is defined as $1 - \frac{\sum_{i=1}^N \mathbb{E}[\epsilon_i^2]}{\sum_{i=1}^N \mathbb{E}[R_i^2]}$, where ϵ_i is the residual

of a cross-sectional regression on the loadings. As in Kelly et al. (2019), we do not demean returns because of their nonstationarity and noise in the mean estimation. Our explained variation measure can be interpreted as a time series R^2 . The third performance measure is the average pricing error normalized by the average mean return to obtain a cross-sectional R^2 measure

$$1 - \frac{\frac{1}{N} \sum_{i=1}^N \mathbb{E}[\epsilon_i]^2}{\frac{1}{N} \sum_{i=1}^N \mathbb{E}[R_i]}.$$

The outputs for our GAN model are the SDF factor weights $\hat{\omega}_{GAN}$. We obtain the risk exposure $\hat{\beta}_{GAN}$ by fitting a feedforward network to predict $R_{t+1}^e F_{t+1}$ and, hence, estimate $\mathbb{E}_t[R_{t+1}^e F_{t+1}]$. Note that this loading estimate $\hat{\beta}_{GAN}$ is only proportional to the population value β , but this is sufficient for projecting on the systematic and nonsystematic component. The conventional forecasting approach, which we label FFN, yields the conditional mean $\hat{\mu}_{FFN}$, which is proportional to β and, hence, is used as $\hat{\beta}_{FFN}$ in the projection. At the same time, $\hat{\mu}_{FFN}$ is proportional to the SDF factor portfolio weights and, hence, also serves as $\hat{\omega}_{FFN}$. Hence, the fundamental difference between GAN and FFN is that GAN estimates a conditional second moment $\beta_{GAN} = \mathbb{E}_t[R_{t+1}^e F_{t+1}]$,

whereas FFN estimates a conditional first moment $\beta_{FFN} = \mathbb{E}_t[R_{t+1}^e]$ for cross-sectional pricing.

Note that the linear model, labeled as LS, is a special case with an explicit solution:

$$\begin{aligned}\hat{\theta}_{LS} &= \left(\frac{1}{T} \sum_{t=1}^T \left(\frac{1}{N} \sum_{i=1}^N R_{t+1,i}^e I_{t,i} \right) \left(\frac{1}{N} \sum_{i=1}^N R_{t+1,i}^e I_{t,i} \right)^\top \right)^{-1} \\ &\quad \left(\frac{1}{NT} \sum_{t=1}^T \sum_{i=1}^N R_{t+1,i}^e I_{t,i} \right) \\ &= \left(\frac{1}{T} \sum_{t=1}^T \tilde{F}_{t+1} \tilde{F}_{t+1}^\top \right)^{-1} \left(\frac{1}{T} \sum_{t=1}^T \tilde{F}_{t+1}^\top \right)\end{aligned}$$

and SDF factor portfolio weights $\omega_{LS} = \hat{\theta}_{LS}^\top I_{t,i}$. The risk exposure $\hat{\beta}_{LS}$ is obtained by a linear regression of $R_{t+1}^e F_{t+1}$ on $I_{t,i}$. As the number of characteristics is very large in our setup, the linear model is likely to suffer from overfitting. The nonlinear models include a form of regularization to deal with the large number of characteristics. In order to make the model comparison valid, we add a regularization to the linear model as well. The regularized linear model EN adds an elastic net penalty to the regression to obtain $\hat{\theta}_{EN}$ and to the predictive regression for $\hat{\beta}_{EN}$:²³

$$\begin{aligned}\hat{\theta}_{EN} &= \arg \min_{\theta} \left(\frac{1}{T} \sum_{t=1}^T \tilde{F}_{t+1} - \frac{1}{T} \sum_{t=1}^T \tilde{F}_{t+1} \tilde{F}_{t+1}^\top \theta \right)^2 \\ &\quad + \lambda_2 \|\theta\|_2^2 + \lambda_1 \|\theta\|_1.\end{aligned}$$

The linear approach with elastic net is closely related to Kozak et al. (2020), who perform mean-variance optimization with an elastic net penalty on characteristic-based factors.²⁴ In addition, we also report the maximum Sharpe ratios for the tangency portfolios based on the Fama-French three- and five-factor models.²⁵

For the four models GAN, FFN, EN, and LS, we obtain estimates of ω for constructing the SDF and estimates of β for calculating the residuals ϵ . We obtain the systematic and nonsystematic return components by projecting returns on the estimated risk exposure $\hat{\beta}$:

$$\hat{\epsilon}_{t+1} = (I_N - \hat{\beta}_t (\hat{\beta}_t^\top \hat{\beta}_t)^{-1} \hat{\beta}_t^\top) R_{t+1}^e.$$

For each model, we report (1) the unconditional Sharpe ratio of the SDF factor, (2) the explained variation in individual stock returns, and (3) the cross-sectional mean²⁶ R^2 :

$$SR = \frac{\hat{\mathbb{E}}[F_t]}{\sqrt{\hat{Var}(F_t)}}, EV = 1 - \frac{\left(\frac{1}{T} \sum_{t=1}^T \frac{1}{N_i} \sum_{i=1}^{N_t} (\hat{\epsilon}_{t+1,i})^2 \right)}{\left(\frac{1}{T} \sum_{t=1}^T \frac{1}{N_i} \sum_{i=1}^{N_t} (R_{t+1,i}^e)^2 \right)},$$

$$XS-R^2 = 1 - \frac{\frac{1}{N} \sum_{i=1}^N \frac{T_i}{T} \left(\frac{1}{T_i} \sum_{t \in T_i} \hat{\epsilon}_{t+1,i} \right)^2}{\frac{1}{N} \sum_{i=1}^N \frac{T_i}{T} \left(\frac{1}{T_i} \sum_{t \in T_i} \hat{R}_{t+1,i} \right)^2}.$$

These are generalizations of the standard metrics used in linear asset pricing.

We also evaluate our models on conventional characteristic sorted portfolios. The portfolio loadings are the average of the stock loadings weighted by the portfolio weights. In more detail, our models provide risk loading $\beta_{t,i}$ s for each individual stock i . The risk loading β_t s for the portfolios are obtained by aggregating the corresponding stock-specific loadings. We obtain the portfolio error from a cross-sectional regression of the portfolio returns on the portfolio β_t at each point in time. This is similar to a standard cross-sectional Fama-MacBeth regression in a linear model with the main difference that the β_t s are obtained from our SDF models on individual stocks. The measures EV and $XS-R^2$ for portfolios follow the same procedure as for individual stocks but use portfolio instead of stock returns. For the individual quantiles, we also report the pricing error $\hat{\alpha}_i$ normalized by the root-mean-squared average returns of all corresponding quantile-sorted portfolios, that is, $\hat{\alpha}_i =$

$$\frac{\hat{\mathbb{E}}[\hat{\epsilon}_{t,i}]}{\sqrt{\frac{1}{N} \sum_{i=1}^N \hat{\mathbb{E}}[R_{t,i}]^2}}.$$

The online appendix includes a simulation that illustrates that all three evaluation metrics (SR, EV, and $XS-R^2$) are necessary to assess the quality of an SDF. A model such as FFN can achieve high Sharpe ratios by loading on some extreme portfolios, but it does not imply that it captures the loading structure correctly.²⁸ Similarly, linear factors can achieve high Sharpe ratios but by construction cannot capture nonlinear and interaction effects in the SDF loadings, which is reflected in lower EV and $XS-R^2$. It does not matter how flexible the model is (e.g., FFN); by conditioning only on the most recent macroeconomic observations, general macroeconomic dynamics are ruled out, which seems to be the most strongly reflected in the Sharpe ratio. The no-arbitrage condition in the GAN model helps to deal with a low signal-to-noise ratio and correctly estimate the SDF loadings of stocks that have small risk premia, which is reflected in the $XS-R^2$.

4. Empirical Results for U.S. Equities

4.1. Data

We collect monthly equity return data for all securities on CRSP. The sample period spans January 1967 to December 2016, totaling 50 years. We divide the full data into 20 years of training sample (1967–1986), 5 years of validation sample (1987–1991), and 25 years of out-of-sample testing sample (1992–2016). We use the one-month Treasury bill rates from the Kenneth French Data Library as the risk-free rate to calculate excess returns.

In addition, we collect the 46 firm-specific characteristics either listed on the Kenneth French Data Library

or used by Freyberger et al. (2020).²⁹ All these variables are constructed from either accounting variables from the CRSP/Compustat database or past returns from CRSP. We follow the standard conventions in the variable definitions, construction, and their updating. Yearly updated variables are updated at the end of each June following the Fama–French convention, whereas monthly changing variables are updated at the end of each month for use in the next month. The full details on the construction of these variables are in the online appendix. In Table B.1, we sort the characteristics into the six categories past returns, investment, profitability, intangibles, value, and trading frictions.

The number of all available stocks from CRSP is around 31,000. As in Kelly et al. (2019) or Freyberger et al. (2020), we are limited to the returns of stocks that have all firm characteristic information available in a certain month, which leaves us with around 10,000 stocks. This is the largest possible data set that can be used for this type of analysis.³⁰

For each characteristic variable in each month, we rank them cross-sectionally and convert them into quantiles. This is a standard transformation to deal with the different scales and is also used in Kelly et al. (2019), Kozak et al. (2020), and Freyberger et al. (2020) among others. In the linear model, the projection $\tilde{F}_{t+1} = \frac{1}{N} \sum_{i=1}^N I_{t,i} R_{t+1,i}^e$ results in long-short factors with an increasing positive weight for stocks that have a characteristic value above the median and a decreasing negative weight for below-median values.³¹ We increase the flexibility of the linear model by including the positive and negative legs separately for each characteristic; that is, we take the rank-weighted average of the stocks with above-median characteristic values and similarly for the below-median values. This results in two factors for each characteristic. Note, that our model includes the conventional long-short factors as a special case in which the long and short legs receive the same absolute weight of opposite sign in the SDF. These factors are still zero-cost portfolios as they are based on excess returns.³²

We collect 178 macroeconomic time series from three sources. We take 124 macroeconomic predictors from the FRED-MD database as detailed in McCracken and Ng (2016). Next, we add the cross-sectional median time series for each of the 46 firm characteristics. The quantile distribution combined with the median level for each characteristic are close to representing the same information as the raw characteristic information but in a normalized form. Third, we supplement the time series with the eight macroeconomic predictors from Welch and Goyal (2007), which are suggested as predictors for the equity premium and are not already included in the FRED-MD database.

We apply standard transformations to the time series data. We use the transformations suggested in

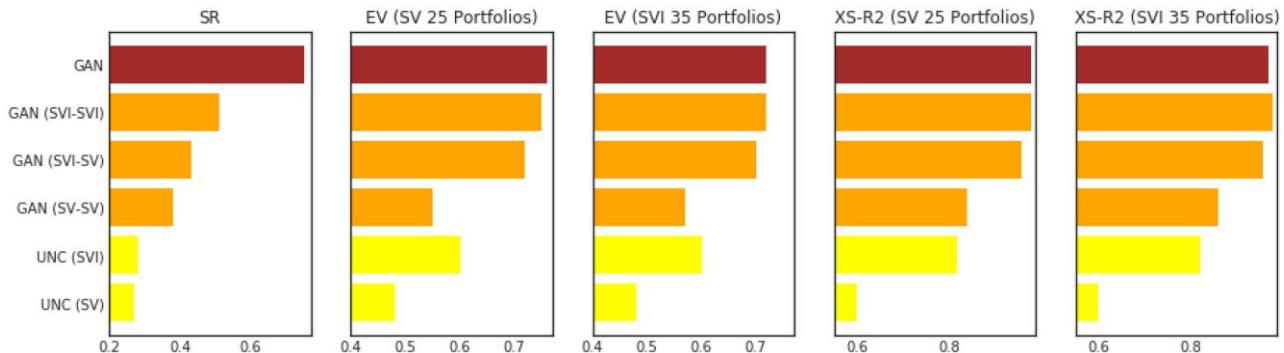
McCracken and Ng (2016) and define transformations for the 46 median and 8 time series from Welch and Goyal (2007) to obtain stationary time series. A detailed description of the macroeconomic variables as well as their corresponding transformations are collected in the online appendix.

4.2. An Illustrative Example of GAN

We illustrate how GAN works with a simple example that uses only the three characteristics: size (LME), book-to-market ratio (BEME), and investment (Investment) for all stocks in our sample but leaves out the macroeconomic information. We show that it is not only crucial which characteristics are included in the SDF weights ω , but also which are included in the test assets constructed by the conditioning function g . UNC denotes the model that is unconditional in the test assets; that is, the test assets are the individual stock returns, and the objective function is based on the unconditional moments. We allow the SDF weights to depend on size and value information denoted by UNC (SV) or also include investment for UNC (SVI). The GAN model allows for test assets that depend on the characteristics, that is, g is a nontrivial function. We first list the characteristics included in the SDF weights ω and then those included in the test assets modeled by g . For example, GAN (SVI-SV) uses size, value, and investment in ω but only size and value in g . In order to keep this simple example interpretable, we restrict g to a scalar function. The loadings β depend on the same information as the SDF weights ω . We also include the benchmark model that is estimated on all the data and discussed in more detail in the next section. We evaluate the asset pricing performance on two well-known sets of test assets: 25 portfolios double-sorted on size and book-to-market (SV 25) and 35 portfolios that include, in addition, 10 decile portfolios sorted on investment (SVI 35). We infer the portfolio's β from the SDF loadings of the individual stocks and the portfolio weights.

Figure 4 shows the out-of-sample Sharpe ratio, explained variation, and cross-sectional R^2 . First, not surprisingly, including more information in the SDF weights ω leads to a better asset pricing model. UNC (SVI) explains more variation and mean returns for portfolios sorted on investment and also for the size and book-to-market portfolios compared with UNC (SV). However, the key insight is that the information in the test assets matters crucially for the SDF recovery. The test assets of GAN (SVI-SVI) include investment information, which results in a better model than GAN (SVI-SV) that has only size and book-to-market information for the test assets. Depending on the metric, GAN (SVI-SVI) is roughly twice as good as UNC (SVI). The top bar is the full benchmark model. Not surprisingly, the higher SR confirms that there is substantially more information that can be extracted by including the other characteristics

Figure 4. (Color online) GAN Illustration



Notes. This figure shows the out-of-sample monthly Sharpe ratio (SR), explained time series variation (EV), and cross-sectional R^2 for different SDF models. UNC (SV) and UNC (SVI) are unconditional models with respect to the test assets; that is, they use only size and value, respectively, size, value, and investment, for the SDF weights but set g to a constant. The GAN models use a nontrivial g . GAN (SVI-SV) allows the SDF weight ω to depend on size, value, and investment but the test asset function g to depend only on size and investment. The model labeled as GAN is our benchmark model estimated with all characteristics and macroeconomic information. We evaluate the model on 25 double-sorted size and book-to-market portfolios (SV 25), and we add another 10 decile portfolios sorted on investment (SVI 35). The portfolios are value weighted.

and macroeconomic time series. However, if the goal is to simply explain the 25 Fama–French double-sorted portfolios, GAN (SVI -SVI) already provides a good model.

Figure 5 explains why we observe these findings. The top figures show heatmaps for the scalar conditioning function g . For GAN (SV-SV) the test assets become long-short portfolios with extreme weights in small value stocks and large growth stocks.³³ When we add investment information, the test assets become essentially long-short portfolios with extreme weights for small, conservative value stocks and large, aggressive growth stocks. GAN has in a data-driven way discovered the structure of the Fama–French type test assets! Not surprisingly, an asset pricing model trained on these test assets better explains portfolios sorted on these characteristics. The bottom figures show the model-implied average excess returns from a cross-sectional regression and average excess returns for the 25 and 35 sorted portfolios for GAN (SVI-SVI) and UNC (SVI). In an ideal model, the points line up on the 45° line. UNC (SVI) fails in explaining small value stocks, whereas the GAN formulation captures mean returns very well for all quantiles. The online appendix contains the detailed results for all the models and test assets.

In summary, the simple example illustrates that the problem of estimating an asset pricing model cannot be separated from the problem of choosing informative test assets. In the next section, we move to our main analysis that includes all firm characteristics and macroeconomic information.

4.3. Cross-Section of Individual Stock Returns

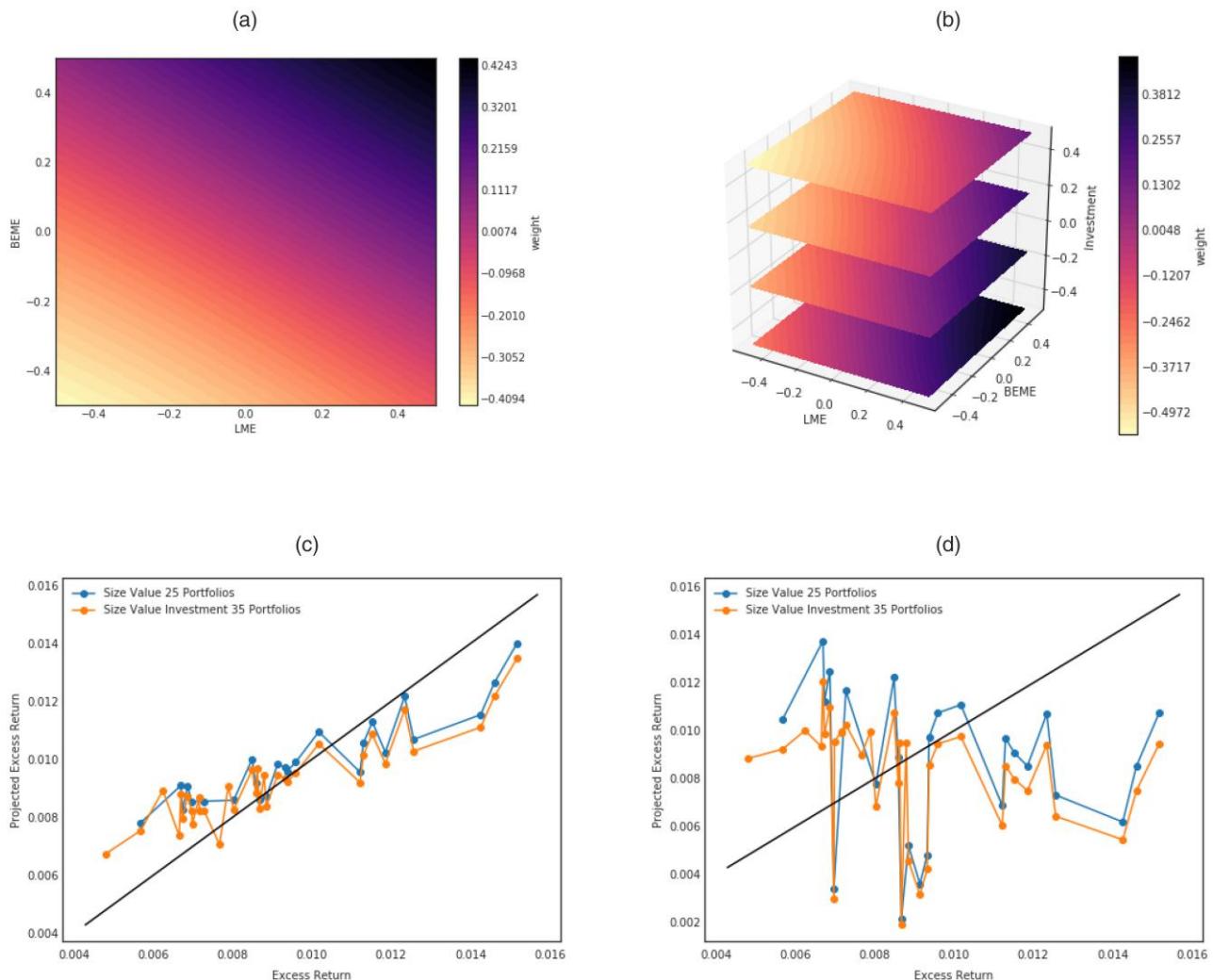
The GAN SDF has a higher out-of-sample Sharpe ratio, explaining more variation and pricing than the other benchmark models. Table 1 reports the three main

performance measures, Sharpe ratio, explained variation, and cross-sectional R^2 , for the four model specifications. The annual out-of-sample Sharpe ratio of GAN is around 2.6 and almost twice as high as with the simple forecasting approach FFN. The nonlinear and interaction structure that GAN can capture results in a 50% increase compared with the regularized linear model. Hence, the more flexible form matters, but an appropriately designed linear model can already achieve an impressive performance. The nonregularized linear model has the worst performance in terms of explained variation and pricing error. GAN explains 8% of the variation of individual stock returns, which is twice as large as the other models. Similarly, the cross-sectional R^2 of 23% is substantially higher than for the other models. Interestingly, the regularized linear model based on the no-arbitrage objective function explains the time series and cross-section of stock returns at least as well as the flexible neural network without the no-arbitrage condition. Each model here uses the optimal set of hyperparameters to maximize the validation Sharpe ratio. In case of the LS, EN, and FFN, this implies leaving out the macroeconomic variables.³⁴

The benchmark criteria differ on the out-of-sample test and in-sample training data. It is important to keep in mind that risk premia and risk exposure of individual stocks are time varying.³⁵ Hence, there is fundamentally no reason to expect the benchmark numbers on different time windows to be the same. Nevertheless, the higher benchmark numbers on the in-sample data suggest a certain degree of overfitting. Thus, the relevant metric is the relative out-of-sample performance between different models as also emphasized among others by Martin and Nagel (2022) and Gu et al. (2020).

Figure 6 summarizes the effect of conditioning on the hidden macroeconomic state variables. First, we add

Figure 5. (Color online) GAN Conditioning Function g and Portfolio Pricing



Notes. (a) Conditioning function g for GAN (SV-SV). (b) Conditioning function g for GAN (SVI-SVI). (c) Portfolio pricing for GAN (SVI-SVI). (d) Portfolio pricing for UNC (SVI). This figure shows the composition of the conditioning function g and average versus model-implied average returns for the sorted portfolios. Panel (a) shows g based only on size and value, whereas panel (b) also includes investment. Panels (c) and (d) show the average excess returns and model-implied average excess returns from a cross-sectional regression for the 25 and 35 sorted portfolios. The tuning parameters are chosen optimally on the validation data and are different from the general benchmark GAN. All results are evaluated out-of-sample.

the 178 macroeconomic variables as predictors to all networks without reducing them to the hidden state variables. The performance for the out-of-sample Sharpe

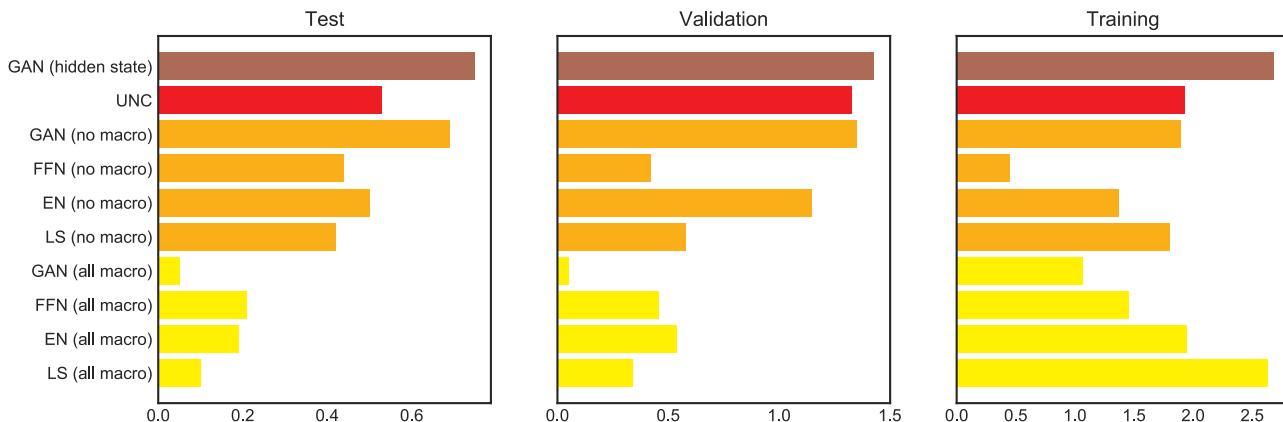
Table 1. Performance of Different SDF Models

Model	SR			EV			XS-R ²		
	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
LS	1.80	0.58	0.42	0.09	0.03	0.03	0.15	0.00	0.14
EN	1.37	1.15	0.50	0.12	0.05	0.04	0.17	0.02	0.19
FFN	0.45	0.42	0.44	0.11	0.04	0.04	0.14	-0.00	0.15
GAN	2.68	1.43	0.75	0.20	0.09	0.08	0.12	0.01	0.23

Note. This table shows the monthly Sharpe ratio (SR) of the SDF, explained time series variation (EV), and cross-sectional mean R^2 for the GAN, FFN, EN, and LS models.

ratio of the LS, EN, FFN, and GAN model completely collapses. First, conditioning only on the last normalized observation of the macroeconomic variables, which is usually an increment, does not allow us to detect a dynamic structure, for example, a business cycle. The decay in the Sharpe ratio indicates that using only the past macroeconomic information results in a loss of valuable information. Even worse, including the large number of irrelevant variables actually lowers the performance compared with a model without macroeconomic information. Although the models use a form of regularization, a too large number of irrelevant variables makes it harder to select those that are actually relevant. The results for the in-sample training data illustrate the complete overfitting when the large

Figure 6. (Color online) Performance of Models with Different Macroeconomic Variables



Notes. This figure shows the Sharpe ratio of SDFs for different inclusions of the macroeconomic information. The GAN (hidden states) is our reference model. UNC is a special version of our model that uses only unconditional moments (but includes LSTM macroeconomic states in the FFN network for the SDF weights). GAN (no macro), FFN (no macro), EN (no macro), and LS (no macro) use only firm-specific information as conditioning variables but no macroeconomic variables. GAN (all macro), FFN (all macro), EN (all macro), and LS (all macro) include all 178 macro variables as predictors (respectively, conditioning variables) without using LSTM to transform them into macroeconomic states.

number of macroeconomic variables is included. FFN, EN, and LS without macroeconomic information perform better, and that is why we choose them as the comparison benchmark models. GAN without the macroeconomic but only firm-specific variables has an out-of-sample Sharpe ratio that is around 10% lower than with the macroeconomic hidden states. This is another indication that it is relevant to include the dynamics of the time series. The UNC model uses only unconditional moments as the objective function; that is, we use a constant conditioning function g but include the LSTM hidden states in the factor weights. The Sharpe ratio is around 20% lower than the GAN with hidden states. These results confirm the insights from the last section. Hence, it is not only important to include all characteristics and the hidden states in the weights and loadings of SDF, but also in the conditioning function g to identify the assets and times that matter for pricing.

4.4. Predictive Performance

The no-arbitrage factor representation implies a connection between average returns of stocks and their risk exposure to the SDF measured by β . The fundamental equation

$$\mathbb{E}_t[R_{t+1,i}^e] = \beta_{t,i} \mathbb{E}_t[F_{t+1}]$$

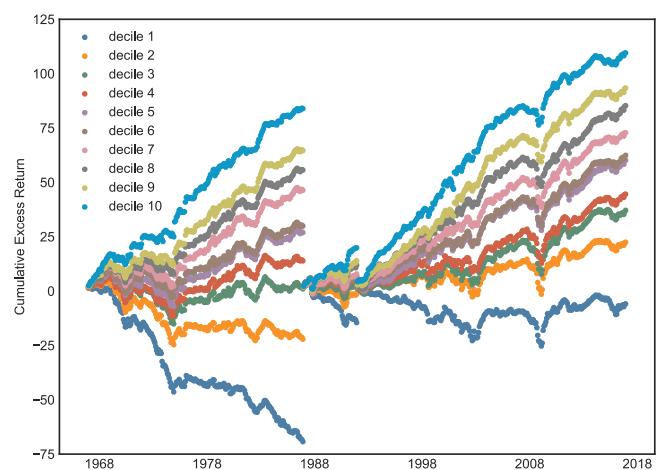
implies that, as long as the conditional risk premium $\mathbb{E}_t[F_{t+1}]$ is positive, which is required by no arbitrage, assets with a higher risk exposure $\beta_{t,i}$ should have higher expected returns. We test the predictive power of our model by sorting stocks into decile portfolios based on their risk loadings.

In Figure 7, we plot the cumulative excess return of decile-sorted portfolios based on risk loading β .

Portfolios based on higher β s have higher subsequent returns. This clearly indicates that risk loadings predict future stock returns. In particular, the highest and lowest deciles clearly separate. The online appendix collects the corresponding results for the other estimation approaches with qualitatively similar findings; that is, the risk loadings predict future returns.

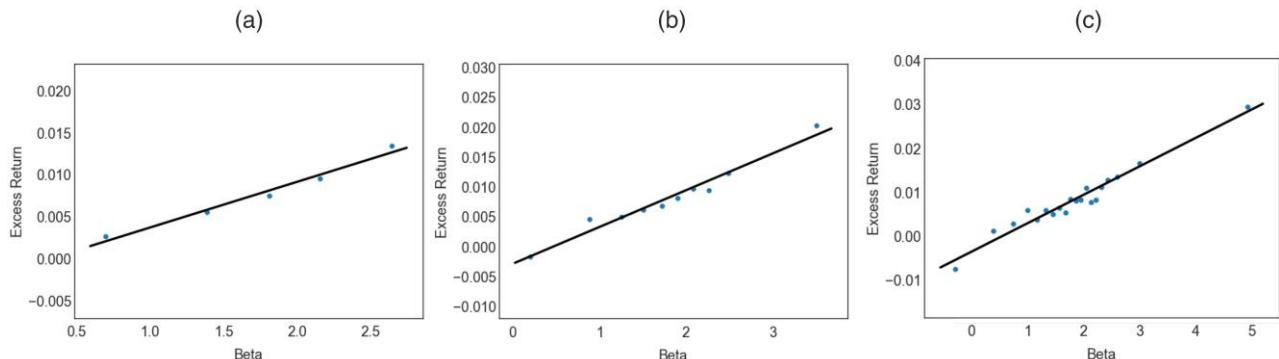
The no-arbitrage condition not only applies a monotonic, but a linear relationship between stock β s and conditional expected returns. An unconditional one-factor model, such as the capital asset pricing model (CAPM), can be tested by evaluating the fit of the

Figure 7. (Color online) Cumulative Excess Return of Decile-Sorted Portfolios with GAN



Notes. This figure shows the cumulative excess return of decile-sorted portfolios based on the risk loadings β . The first portfolio is based on the smallest decile of risk loadings, whereas the last decile portfolio is constructed with the largest loading decile. Within each decile, the stocks are equally weighted.

Figure 8. (Color online) Expected Excess Returns of β -Sorted Portfolios as a Function of β



Notes. (a) Five β -sorted quintiles. (b) Ten β -sorted deciles. (c) Twenty β -sorted quantiles. This figure shows expected excess returns of β -sorted portfolios for GAN on the test sample. Stocks are sorted into 5, 10, or 20 quantiles every month. We plot them against the β of each portfolio that averages the individual stock $\beta_{t,i}$ over stocks and time within each portfolio. The linear line denotes a linear regression with intercept, which yields an R^2 of 0.98 (quintiles), 0.97 (deciles), and 0.95 (20 quantiles). The SDF itself has a β equal to one.

security market line, that is, the relationship between expected returns and the β s of the assets. In our case, the stock-specific $\beta_{t,i}$ are time varying, but by construction the β -sorted portfolios should have a close to constant slope with respect to the SDF.³⁶ In Figure 8, we plot the expected excess returns of the 10 β -sorted deciles as well as for 5 and 20 β -sorted quantile portfolios against their average β s. No arbitrage imposes a linear relationship and a zero intercept. Indeed, for all three plots, the relationship is almost perfectly linear with a R^2 of 0.98, 0.97, and 0.95, respectively. However, the intercept seems to be slightly below zero. This indicates a very good but not perfect fit.

The systematic return difference of the β -sorted portfolios is not explained by the market or Fama–French

factors. Table 2 reports the time series pricing errors with corresponding t -statistics for the 10 decile-sorted portfolios for the three factor models. Obviously, the pricing errors are highly significant, and expected returns of almost all decile portfolios are not explained by the Fama–French factors. The GRS test clearly rejects the null hypothesis that either of the factor models prices this cross-section. These β -sorted portfolios equally weight the stocks within each decile. The online appendix shows that the findings extend to value-weighted β -sorted portfolios.

4.5. Pricing of Characteristic Sorted Portfolios

Our approach achieves an unprecedented pricing performance on standard test portfolios. Asset pricing testing is

Table 2. Time Series Pricing Errors for β -Sorted Portfolios

Decile	Market-Rf				Fama–French 3				Fama–French 5					
	Average returns		Full		Test		Full		Test		Full		Test	
	Full	Test	α	t	α	t	α	t	α	t	α	t	α	t
1	-0.12	-0.02	-0.19	-8.92	-0.11	-3.43	-0.21	-12.77	-0.13	-5.01	-0.20	-11.99	-0.12	-4.35
2	-0.00	0.05	-0.07	-4.99	-0.04	-1.56	-0.09	-8.79	-0.05	-3.22	-0.09	-8.29	-0.05	-2.68
3	0.04	0.08	-0.02	-2.01	-0.00	-0.16	-0.04	-5.18	-0.02	-1.40	-0.04	-4.87	-0.01	-1.05
4	0.07	0.09	-0.00	-0.03	0.01	0.68	-0.02	-2.30	-0.00	-0.35	-0.02	-2.86	-0.01	-0.54
5	0.10	0.12	0.03	2.75	0.04	2.50	0.01	2.08	0.03	2.46	0.01	1.36	0.03	2.17
6	0.11	0.12	0.04	3.16	0.05	2.77	0.02	2.75	0.03	2.85	0.01	1.51	0.02	2.20
7	0.14	0.15	0.07	5.62	0.07	3.92	0.05	6.61	0.05	4.39	0.04	5.16	0.04	3.41
8	0.18	0.18	0.11	7.41	0.10	5.12	0.08	9.32	0.08	5.83	0.07	8.05	0.07	4.86
9	0.22	0.21	0.15	7.83	0.13	5.37	0.11	9.16	0.11	5.71	0.11	8.58	0.11	5.39
10	0.37	0.37	0.29	9.22	0.27	6.05	0.24	10.03	0.25	6.27	0.25	10.43	0.27	6.59
10 – 1	0.48	0.39	0.47	18.93	0.38	10.29	0.45	18.50	0.38	10.14	0.46	18.13	0.39	9.96
GRS asset pricing test	GRS	p	GRS	p	GRS	p	GRS	p	GRS	p	GRS	p	GRS	p
	42.23	0.00	11.58	0.00	39.72	0.00	11.25	0.00	37.64	0.00	10.75	0.00		

Notes. This table shows the average returns, time series pricing errors, and corresponding t -statistics for β -sorted decile portfolios based on GAN. The pricing errors are based on the CAPM and Fama–French three- and five-factor models. Returns are annualized. The GRS test is under the null hypothesis of correctly pricing all decile portfolios and includes the p -values. We consider the full time period and the test period. Within each decile, the stocks are equally weighted.

usually conducted on characteristic sorted portfolios that isolate the pricing effect of a small number of characteristics. We sort the stocks into value-weighted decile and double-sorted 25 portfolios based on the characteristics.³⁷ Table C.1 starts with four sets of decile-sorted portfolios. We choose short-term reversal and momentum as these are among the two most important variables as discussed in the next sections and size and book-to-market sorted portfolios, which are well-studied characteristics. GAN can substantially better capture the variation and mean return for short-term reversal and momentum-sorted decile portfolios. EN and FFN have a very similar performance. The better GAN results are driven by explaining the extreme decile portfolios (the 10th decile for short-term reversal and the 1st decile for momentum). All approaches perform very similarly for the middle portfolios. It turns out that book-to-market and size-sorted portfolios are very “easy” to price. All models have time series R^2 above 70% and cross-sectional R^2 close to one. Hence, all models seem to capture this pricing information very well although the GAN results are still slightly better than for the other models.

Table C.2 repeats the same analysis on short-term reversal and momentum double-sorted and size and book-to-market double-sorted portfolios. The takeaways are similar to the decile-sorted portfolios. GAN outperforms FFN and EN on the momentum-related portfolios, whereas all three models are able to explain the size and value double-sorted portfolios. Importantly, the linear

EN becomes worse on the double-sorted reversal and momentum portfolios. This is due to the extreme corner portfolios, which are, in particular, low-momentum and high short-term reversal stocks. This implies that the linear model cannot capture the interaction between characteristics, whereas the GAN model successfully identifies the potentially nonlinear interaction effects.

Our findings generalize to other decile-sorted portfolios. Table 3 collects the explained variation and cross-sectional R^2 for all decile-sorted portfolios. It is striking that GAN is always better than the other two models in explaining variation. At the same time, GAN achieves a cross-sectional R^2 higher than 90% for all characteristics. In the few cases in which the other models have a slightly higher cross-sectional R^2 , this number is very close to one; that is, all models can essentially perfectly explain the pricing information in the deciles. In summary, GAN strongly dominates the other methods in explaining sorted portfolios. The results show that (1) the nonlinearities and interactions matter as GAN is better than EN and (2) the no-arbitrage condition extracts additional information as GAN is better than FFN.

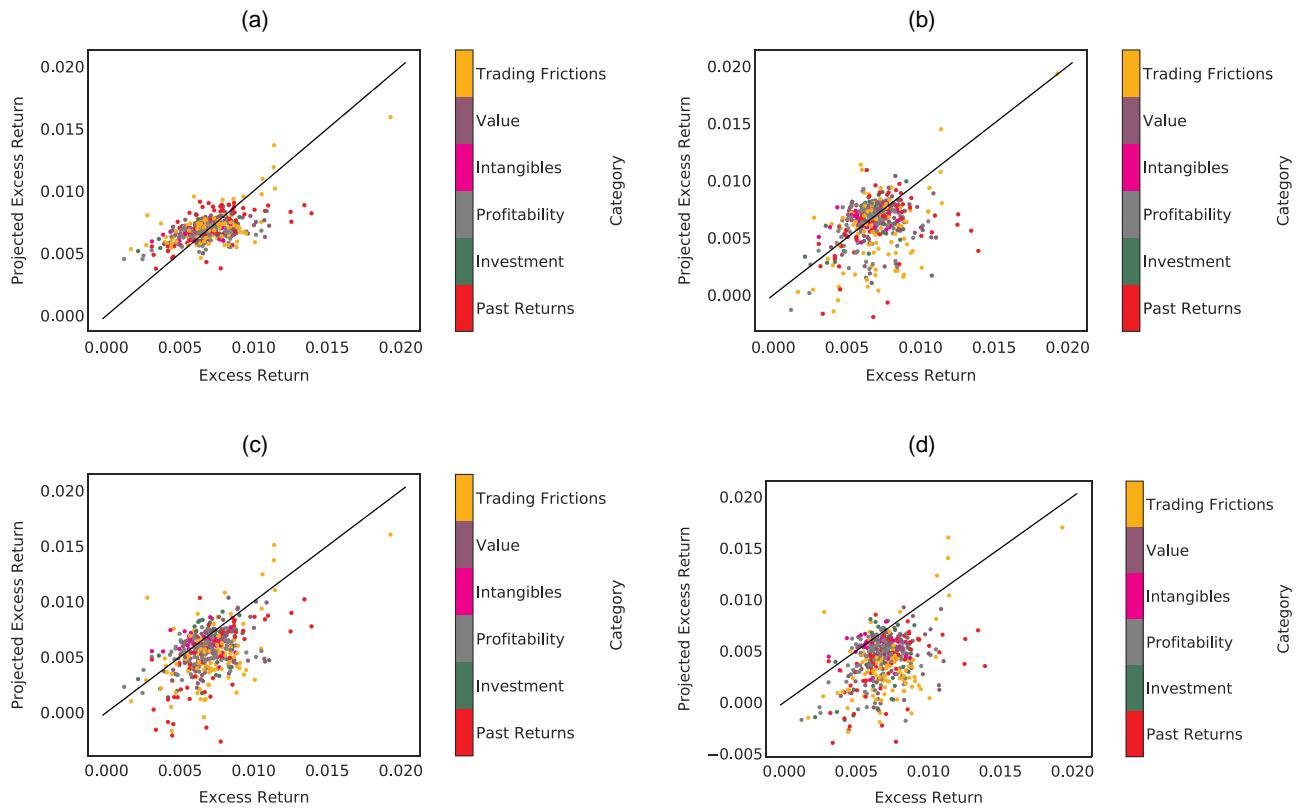
Figure 9 visualizes the ability of GAN to explain the cross-section of expected returns for all value-weighted characteristic sorted deciles. We plot the average excess return and the model implied average excess return. The GAN SDF captures the correct monotonic behavior, but its prediction is biased toward the mean. In contrast, the predictions of the other three models show a

Table 3. Explained Variation and Pricing Errors for Decile Sorted Portfolios

Charact.	Explained variation			Cross-sectional R^2			Charact.	Explained variation			Cross-sectional R^2		
	EN	FFN	GAN	EN	FFN	GAN		EN	FFN	GAN	EN	FFN	GAN
ST_REV	0.43	0.58	0.70	0.45	0.79	0.94	Q	0.68	0.70	0.78	0.97	0.92	0.96
SUV	0.42	0.75	0.83	0.64	0.97	0.99	Investment	0.54	0.65	0.75	0.91	0.94	0.98
r12_2	0.26	0.27	0.54	0.66	0.71	0.93	PM	0.52	0.42	0.68	0.90	0.86	0.93
NOA	0.58	0.69	0.78	0.94	0.96	0.95	DPI2A	0.57	0.70	0.78	0.90	0.95	0.97
SGA2S	0.52	0.63	0.73	0.93	0.95	0.96	ROE	0.59	0.56	0.76	0.91	0.86	0.97
LME	0.83	0.78	0.86	0.96	0.95	0.97	S2P	0.69	0.79	0.82	0.98	0.98	0.97
RNA	0.50	0.48	0.69	0.93	0.87	0.96	FC2Y	0.56	0.71	0.76	0.91	0.94	0.95
LTurnover	0.52	0.57	0.68	0.88	0.89	0.96	AC	0.63	0.79	0.82	0.96	0.98	0.98
Lev	0.52	0.63	0.73	0.90	0.92	0.95	CTO	0.59	0.73	0.79	0.92	0.96	0.97
Resid_Var	0.52	0.27	0.65	0.84	0.73	0.97	LT_Rev	0.60	0.59	0.72	0.93	0.85	0.94
ROA	0.51	0.44	0.70	0.92	0.93	0.98	OP	0.56	0.48	0.74	0.97	0.88	0.98
E2P	0.48	0.44	0.67	0.86	0.80	0.95	PROF	0.58	0.62	0.76	0.91	0.98	0.95
D2P	0.47	0.51	0.72	0.82	0.85	0.94	IdioVol	0.43	0.27	0.66	0.79	0.72	0.97
Spread	0.49	0.32	0.60	0.76	0.71	0.92	r12_7	0.37	0.42	0.66	0.84	0.86	0.93
CF2P	0.46	0.47	0.66	0.90	0.89	0.99	Beta	0.45	0.46	0.62	0.83	0.87	0.97
BEME	0.70	0.75	0.82	0.97	0.94	0.98	OA	0.65	0.78	0.83	0.88	0.92	0.93
Variance	0.48	0.27	0.61	0.74	0.72	0.90	ATO	0.58	0.70	0.77	0.96	0.98	0.99
D2A	0.57	0.71	0.78	0.96	0.96	0.97	MktBeta	0.44	0.44	0.64	0.81	0.85	0.97
PCM	0.66	0.79	0.82	0.97	0.98	0.99	OL	0.60	0.73	0.78	0.95	0.97	0.97
A2ME	0.72	0.79	0.83	0.97	0.96	0.98	C	0.51	0.65	0.73	0.90	0.93	0.95
AT	0.77	0.70	0.83	0.77	0.89	0.92	r36_13	0.54	0.53	0.69	0.92	0.82	0.93
Rel2High	0.46	0.33	0.60	0.90	0.83	0.97	NI	0.51	0.60	0.75	0.88	0.96	0.99
CF	0.61	0.64	0.78	0.89	0.85	0.96	r2_1	0.51	0.52	0.69	0.87	0.90	0.95

Note. This table shows the out-of-sample explained variation and cross-section R^2 for 46 decile-sorted and value-weighted portfolios.

Figure 9. (Color online) Predicted Returns for Value-Weighted Characteristic Sorted Portfolios



Notes. (a) GAN. (b) FFN. (c) EN. (d) LS. This figure shows the predicted and average excess returns for value-weighted characteristic sorted decile portfolios for the four SDF models. We have in total 460 decile portfolios in six different anomaly categories.

larger discrepancy that holds for characteristics of all groups. Figure 10 shows the prediction results for equally weighted decile portfolios. All models seem to perform slightly better, but the general findings are the same.

4.6. Variable Importance

What is the structure of the SDF factor? As a first step in Table E.1, we compare the GAN SDF with the Fama–French five-factor model. None of the five factors has a high correlation with our SDF, whereas the profitability factor has the highest correlation with 17%. The market factor has only a correlation of 10%. Next, we run a time series regression to explain the GAN SDF with the Fama–French five factors. Only the profitability factor is significant. The strongly significant pricing error indicates that these factors fail to capture the pricing information in our SDF portfolio.

We rank the importance of firm-specific and macroeconomic variables for the pricing kernel based on the sensitivity of the SDF weight ω with respect to these variables. Our sensitivity analysis is similar to Sadhwani et al. (2021) and Horel and Giesecke (2020) and based on the average absolute gradient. More specifically, we

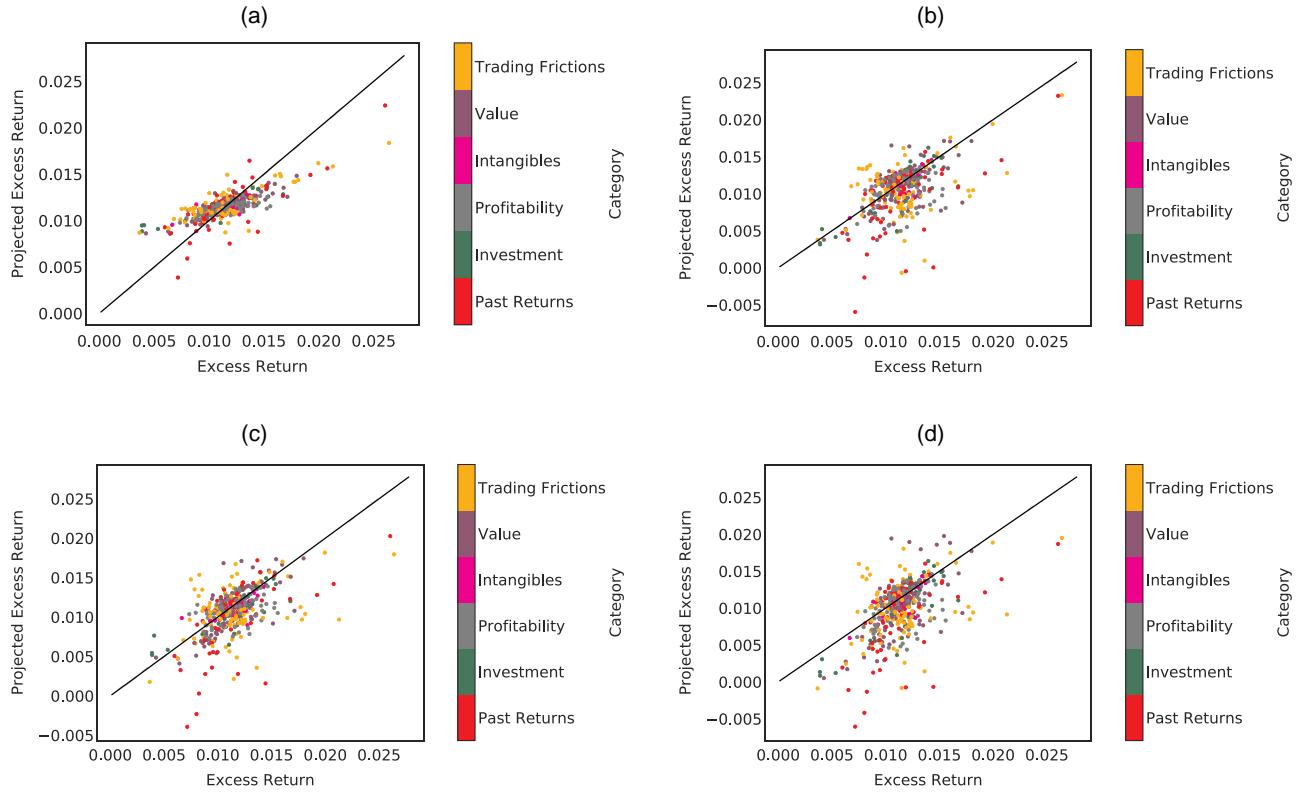
define the sensitivity of a particular variable as the average absolute derivative of the weight w with respect to this variable:

$$\text{Sensitivity}(x_j) = \frac{1}{C} \sum_{i=1}^N \sum_{t=1}^T \left| \frac{\partial w(I_t, I_{t,i})}{\partial x_j} \right|,$$

where C a normalization constant. This simplifies to the standard slope coefficient in the special case of a linear regression framework. A larger sensitivity means that a variable has a larger effect on the SDF weight ω .

Figure 11 ranks the variable importance of the 46 firm-specific characteristics for GAN. The sum of all sensitivities is normalized to one. Figures 12, D.2, and D.3 collect the corresponding results for FFN, EN, and LS. All three models, GAN, FFN, and EN, select trading frictions and past returns as being the most relevant categories. The most important variables for GAN are short-term reversal (ST_REV), standard unexplained volume (SUV), and momentum (r12_2). Importantly, for GAN, all six categories are represented among the first 20 variables, which includes value, intangibles, investment, and profitability characteristics. The SDF composition is different for FFN, for which the first 14 characteristics are almost only in the trading friction

Figure 10. (Color online) Predicted Returns for Equally Weighted Characteristic Sorted Portfolios



Notes. (a) GAN. (b) FFN. (c) EN. (d) LS. This figure shows the predicted and average excess returns for equally weighted characteristic sorted decile portfolios for the four SDF models. We have in total 460 decile portfolios in six different anomaly categories.

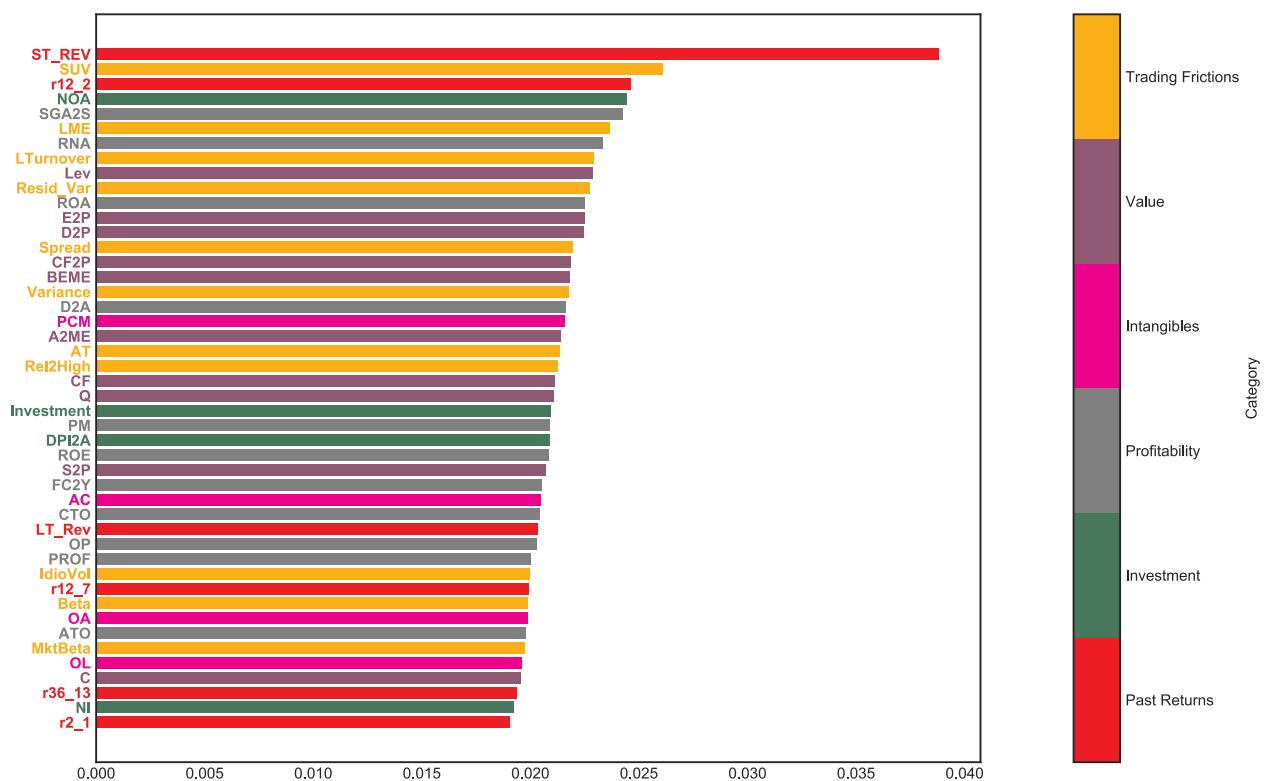
and past return category. More specifically, this SDF loads heavily on short-term reversal, illiquidity measured by unexplained volume and size, which raises the suspicion that a simple forecasting approach might focus mainly on illiquid penny stocks. The no-arbitrage condition with informative test assets seems to be necessary to discipline the model to capture the pricing information in other characteristics. Figure D.4 shows the variable importance ranking for the conditioning vector g . The GAN test assets depend on all six major anomaly categories. These test assets ensure that the GAN SDF also reflects this information. The linear model with regularization also selects variables from all six categories among the first nine variables. Note that the elastic net penalty removes characteristics that are close substitutes, for example, as the dividend-price ratio (D2P) and book-to-market ratio (BEME) capture similar information, the regularized model only selects one of them. The linear model without regularization cannot handle the large number of variables and, not surprisingly, results in a different ranking.

Figure D.1 shows the importance of the macroeconomic variables for the GAN model. These variables are first summarized into the four hidden state processes before they enter the weights of the SDF. First, it is apparent that most macroeconomic variables have

a very similar importance. This is in line with a model in which there is a strong dependency between the macroeconomic time series that is driven by a low-dimensional nonlinear factor structure. A simple example is the factor model in Ludvigson and Ng (2009) in which the information in a macroeconomic data set very similar to ours is summarized by a small number of PCA factors. As the first PCA factor is likely to pick up a general economic market trend, it affects all variables. If the SDF structure depends on this PCA factor, all macroeconomic variables appear to be important (of potentially similar magnitude). It is important to keep in mind that a simple PCA analysis of the macroeconomic variables does not work in our asset pricing context. The reason is that the PCA factors are mainly based on increments of the macroeconomic time series and, hence, would not capture the dynamic pattern.³⁸ The two most relevant variables that stand out in our importance ranking are the median bid-ask spread (Spread) and the federal fund rate (FEDFUNDS). These can be interpreted as capturing the overall economic activity level and overall market volatility.

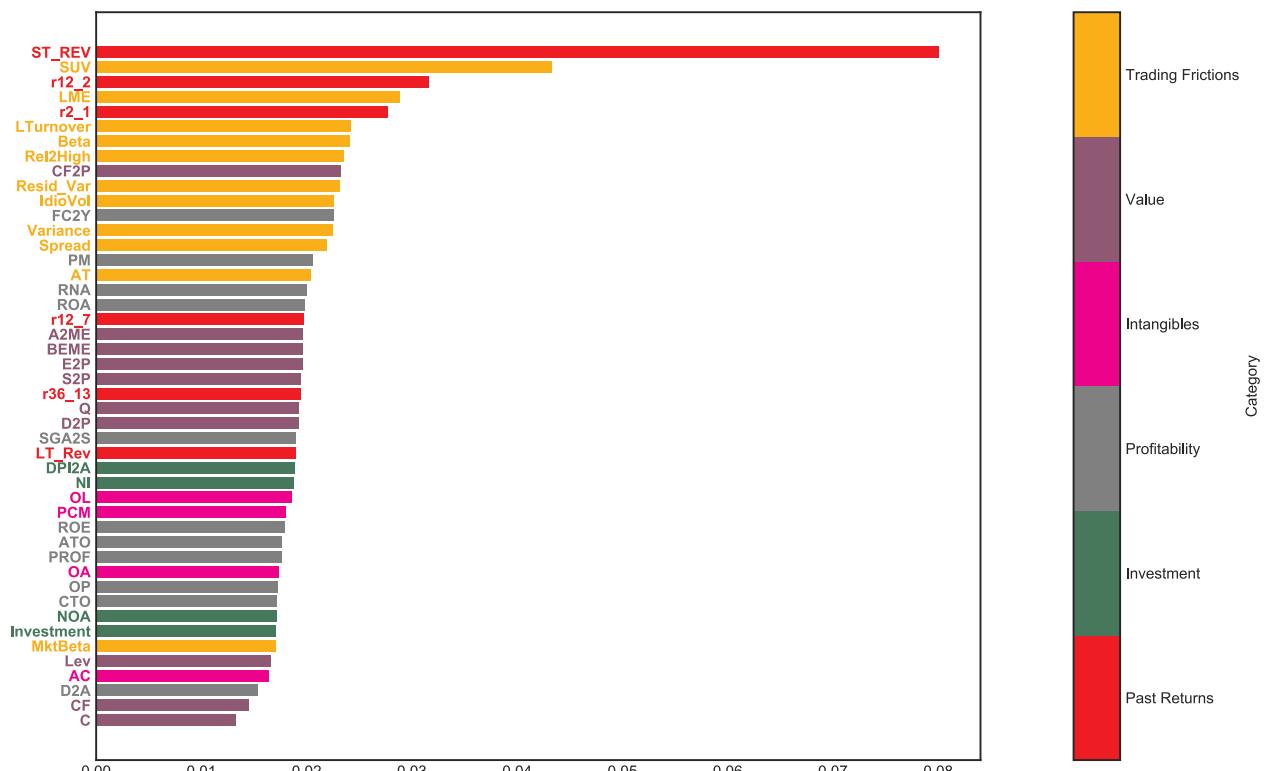
We show that the hidden macroeconomic states are closely linked to business cycles and overall economic activity. Figure 13 plots the time series of the four hidden macroeconomic state variables. These variables are the outputs from the LSTM that encodes the history of

Figure 11. (Color online) Characteristic Importance for GAN SDF



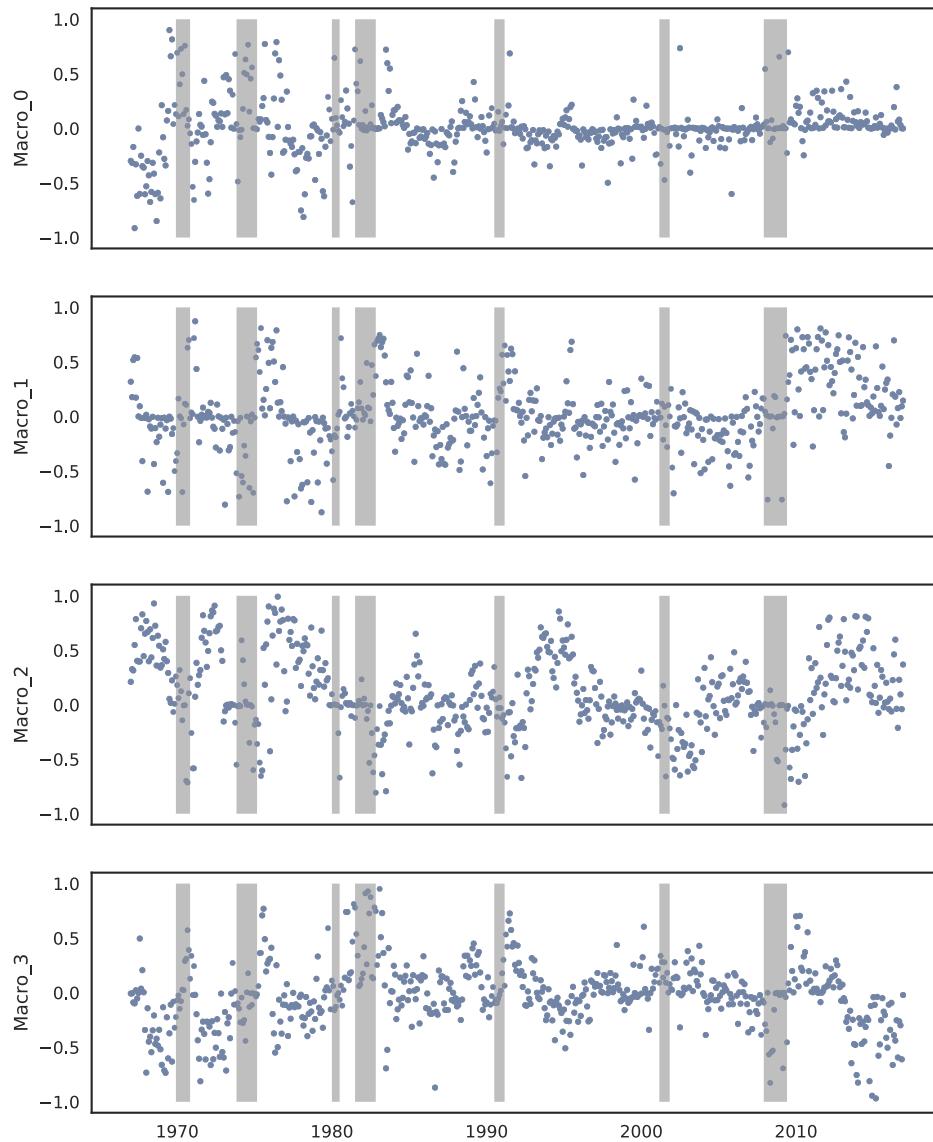
Notes. The figure shows the GAN variable importance ranking of the 46 firm-specific characteristics in terms of average absolute gradient (VI) on the test data. The values are normalized to sum up to one.

Figure 12. (Color online) Characteristic Importance for FFN SDF



Notes. The figure shows the FFN variable importance ranking of the 46 firm-specific characteristics in terms of average absolute gradient (VI) on the test data. The values are normalized to sum up to one.

Figure 13. (Color online) Macroeconomic Hidden State Processes (LSTM Outputs)



Notes. The figure shows the four macroeconomic hidden state processes extracted with LSTM in the GAN model. The gray areas mark NBER recession periods. The time series are one representative estimation in the multiple ensemble estimations.

macroeconomic information. Here, we report one representative fit of the LSTM from the nine ensemble estimates.³⁹ The gray shaded areas indicate NBER recessions.⁴⁰ First, it is apparent that the state variables, in particular, for the third and fourth state, peak during times of recessions. Second, the state processes seem to have a cyclical behavior, which confirms our intuition that the relevant macroeconomic information is likely to be related to business cycles. The cycles and peaks of the different state variables do not coincide at all times, indicating that they capture different macroeconomic risks.

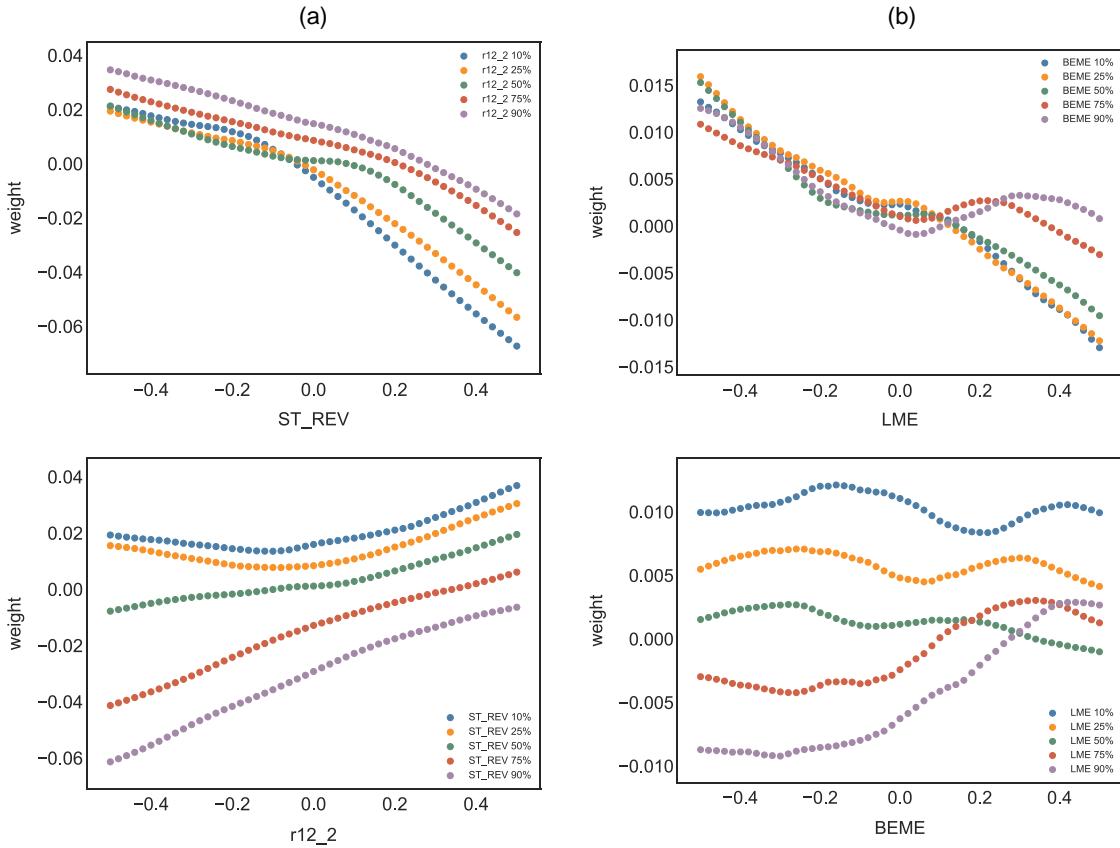
4.7. SDF Structure

We study the structure of the SDF weights and betas as a function of the characteristics. Our main findings are

twofold: Surprisingly, individual characteristics have an almost linear effect on the pricing kernel and the risk loadings; that is, nonlinearities matter less than expected for individual characteristics. Second, the better performance of GAN is explained by nonlinear interaction effects; that is, the general functional form of our model is necessary for capturing the dependency between multiple characteristics.

Figure E.1 plots the one-dimensional relationship between the SDF weights ω and one specific characteristic. The other variables are fixed at their mean values.⁴¹ In the case of a linear model, these plots simply show the slope of a linear regression coefficient. As we include a separate long and short leg for the linear model, we allow for a kink at the median value. Otherwise, the linear

Figure 14. (Color online) SDF Weight ω as a Function of Characteristics for GAN



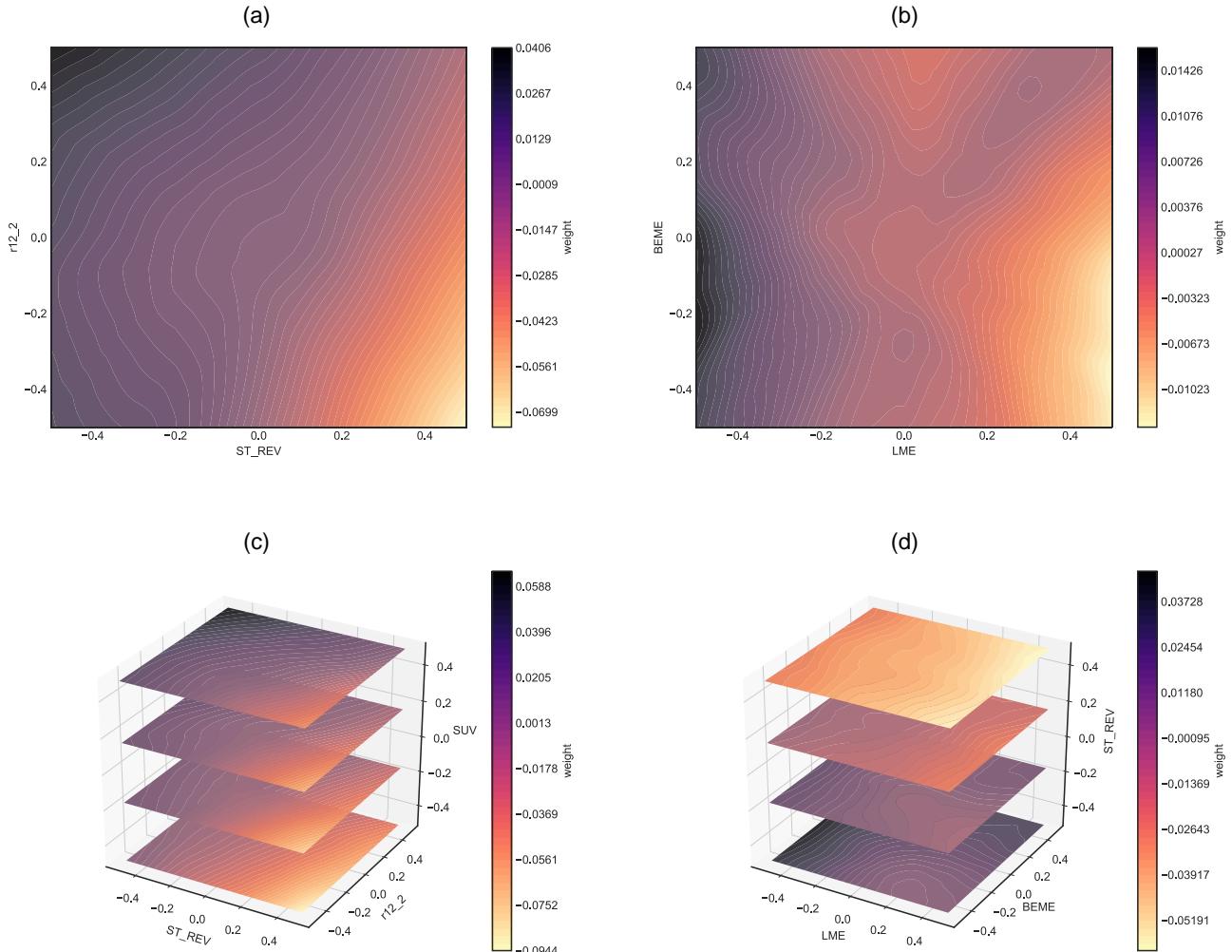
Notes. (a) Interaction between short-term reversal (ST_REV) and momentum (r12_2). (b) Interaction between size (LME) and book to market ratio (BEME). These figures show the SDF weight ω as function of short-term reversal, momentum, size, and book-to-market ratio for different quantiles of the second variable, keeping the remaining variables at their mean level.

model is simply a straight line. For the nonlinear GAN and FFN, the one-dimensional relationship can take any functional form. We show the univariate functional form for the three most relevant characteristics in Figure E.1, whereas the online appendix collects the results for the other characteristics. It is striking how close the functional form of the SDF for GAN and FFN is to a linear function. This explains why linear models are actually so successful in explaining single-sorted characteristics. For a small number of characteristics, for example, short-term reversal, GAN has some nonlinearities around the median. These are exactly the decile-sorted portfolios for which GAN performs better than FFN and EN. However, for most characteristics, the pricing kernel depends almost linearly on the characteristics as long as we consider a one-dimensional relationship. However, it seems to be relevant to allow the low and high quantiles to have different linear slopes. The linear model without regularization obtains a relationship for some characteristics that is completely out of line with the other models. Given the worse overall performance of LS, this suggests that LS suffers from severe overfitting.

Figures 14 and 15 show the crucial finding for this section: nonlinearities matter for interactions. Figure 14 plots the SDF weight of one characteristic conditioned on a quantile of a second characteristic that can be different than the median. In an additive model without interaction, all lines would be parallel shifts. This is exactly what we see for the two linear models.⁴² Interestingly, for size and value, the FFN model also has almost parallel shifts in the SDF weights, implying that it does not capture interactions. However, for GAN, small stocks have a very different exposure to value than large-cap stocks. The line plots for GAN reveal more complex interaction patterns than for the other models. In general, the shape seems to become more nonlinear when conditioning the second characteristic on an extreme quantile.

Instead of conditioning on only five quantiles for the second characteristic, we plot the two-dimensional pricing kernel for GAN in Figure 15. It confirms that the combined size and book-to-market characteristics have a highly nonlinear effect on the GAN pricing kernel. The triple interaction in Figure 15 shows that low short-term reversal, high momentum, and high explained volume

Figure 15. (Color online) SDF Weight ω as a Function of Characteristics for GAN



Notes. (a) Interaction between short-term reversal (ST_REV) and momentum (r_{12_2}). (b) Interaction between size (LME) and book to market ratio (BEME). (c) Interaction between short-term reversal (ST_REV), momentum (r_{12_2}), and standard unexplained volume (SUV). (d) Interaction between size (LME), book to market ratio (BEME), and short-term reversal (ST_REV). These figures show the SDF weight ω as two- and three-dimensional function of characteristics, keeping the remaining variables at their mean level.

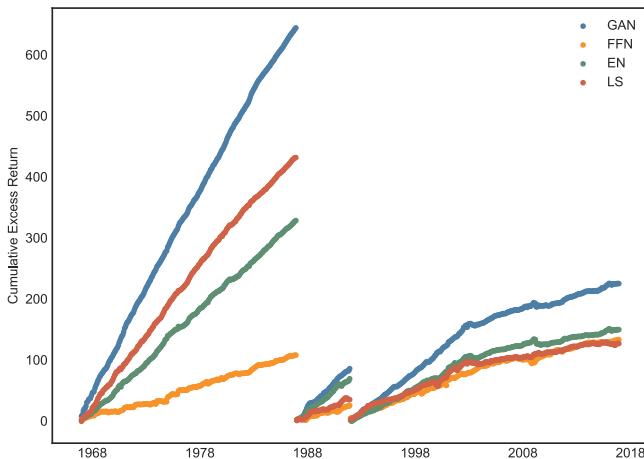
has the highest positive weight, whereas high reversal, low momentum, and low unexplained volume has the largest negative weight in the kernel when conditioning on these three characteristics. Low reversal and low momentum or high reversal and high momentum have an almost neutral effect independent of unexplained volume. The interaction effect for size, book-to-market, and short-term reversal is even more complicated.

4.8. Robustness Results and Extensions

The online appendix collects an extensive set of robustness results and extensions. First, in Online Appendix IA.B, we show that our GAN framework is complementary to conditional and unconditional multifactor models. Multifactor models are based on the assumption that the SDF is a linear combination of the multiple factors. In

an unconditional multifactor model with constant factor loadings and pricing errors, the SDF can easily be constructed as the unconditional mean-variance efficient combination of the factors. The time series regression pricing error in such a multifactor model is identical to that of a one-factor regression on the SDF. However, this relationship does not hold out-of-sample for unconditional models and breaks down in-sample and out-of-sample for conditional multifactor models. So far, the factor literature mainly focuses on extracting the factors and their loadings but is largely silent on the construction of a coherent conditional SDF framework based on a conditional factor structure. Our GAN framework can help to close this gap. We combine the general conditional multifactor model of Kelly et al. (2019) with our model. We show that using the additional economic

Figure 16. (Color online) Cumulative Excess Returns of SDF



Notes. The figure shows the cumulative excess returns for the SDF for GAN, FFN, EN, and LS. Each factor is normalized by its standard deviation for the time interval under consideration.

structure of spanning the SDF with IPCA factors and combining it with our SDF framework can lead to an even better asset pricing model.

Second, we show in Online Appendix IA.C that our findings are robust to small-cap stocks, the choice of tuning parameters, and the time period under consideration and are not exploiting limits to arbitrage. When allowing for a time-varying functional form by estimating the SDF on a rolling window, we find that it is highly correlated with the benchmark SDF and only leads to minor improvements. The estimation is robust to the choice of the tuning parameters. All of the best performing models selected on the validation data capture essentially the same asset pricing model. Our asset pricing model also performs well after excluding small and illiquid stocks from the test assets or the SDF. Thus, our GAN does not simply target pricing information,

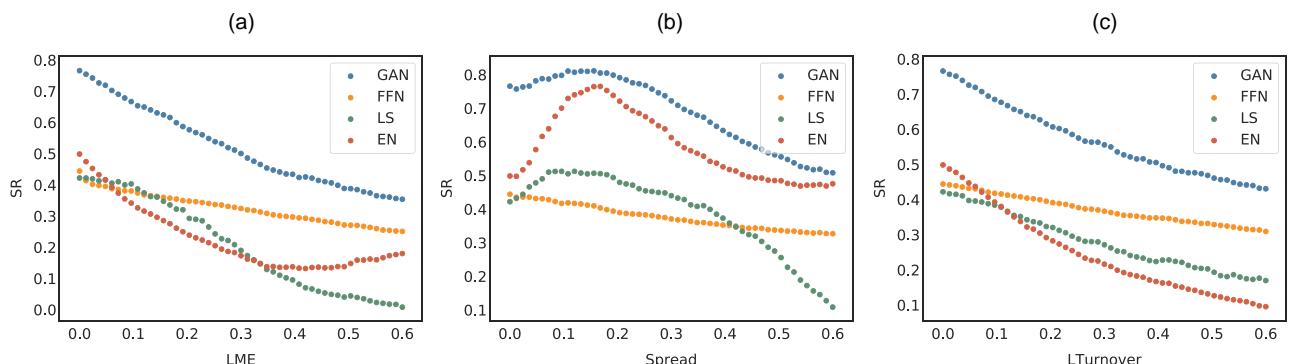
which is subject to limits to arbitrage and, therefore, hard to exploit.

4.9. Machine Learning Investment

The GAN SDF is a tradeable portfolio with an attractive risk–return trade-off. Besides the Sharpe ratio, we calculate the maximum one-month loss and maximum drawdown of the four benchmark models and also the Fama–French three-3 and five-factor models.⁴³ The number of consecutive losses as measured by drawdown and the maximum loss for the GAN model are comparable to the other models, whereas the Sharpe ratio is by far the highest. Figure 16 plots the cumulative return for each model normalized by the standard deviation. As suggested by the risk measures, the GAN return exceeds the other models, whereas it avoids fluctuations and large losses. In addition, the GAN factor has a comparable or even lower turnover than the other SDF portfolios. This suggests that all approaches are exposed to similar transaction costs, and it is valid to directly compare their risk-adjusted returns.⁴⁴

Avramov et al. (2022) raise the concern that the performance of machine learning portfolios could deteriorate in the presence of trading costs because of high turnover or extreme positions.⁴⁵ This important insight can be taken into account when constructing machine learning investment portfolios. Figure 17 shows the out-of-sample Sharpe ratios of the SDF portfolios after we set the SDF weights ω to zero for stocks with either market capitalization, bid–ask spread or turnover below a specified cross-sectional quantile at the time of portfolio construction. The idea is to remove stocks that are more prone to trading frictions. There is a clear trade-off between trading frictions and achievable Sharpe ratios. However, this indicates that a machine learning portfolio can be estimated to optimally trade off the trading frictions and a high risk-adjusted return. For example,

Figure 17. (Color online) Trading Friction Cutoffs



Notes. (a) Size cutoff. (b) Spread cutoff. (c) Turnover cutoff. This figure shows the out-of-sample Sharpe ratios of the SDF for GAN, FFN, EN, and LS after we have set the portfolio weights ω to zero if either the market capitalization (LME), bid–ask spread (Spread), or turnover (Lturnover) at the time of investment is below a specified cross-sectional quantile.

GAN without 40% of the smallest stocks still has an annual SR of 1.73, without 40% of the highest bid-ask spreads the SR is still 2.07, and without 40% of the stocks with the least trading activity measured by turnover the SR is 1.87. Note that these are all lower bounds as GAN has not been reestimated without these stocks, but we just set the portfolio weights of the stocks below the cutoffs to zero.

A more fundamental point is that machine learning methods should be designed to extract the signals for the actual objective about which the user cares. In particular, a signal that is optimal for prediction might not be optimal for asset pricing or investment. This paper estimates in one step the optimal signal for the SDF construction, which by construction is the conditionally mean-variance efficient portfolio and, hence, maximizes a conditional Sharpe ratio objective. A step further is to include trading frictions directly in this estimation. A promising step into this direction is presented in Bryzgalova et al. (2019), who estimate mean-variance efficient portfolios with decision trees that can easily incorporate constraints. Investors might also have alternative trading objectives. Cong et al. (2020) introduce a reinforcement learning framework to directly optimize portfolio objectives, including and beyond Sharpe ratios and allowing for trading frictions. Guijarro-Ordonez et al. (2019) design optimal statistical arbitrage strategies with various objectives under trading friction constraints.

5. Conclusion

We propose a new way to estimate asset pricing models for individual stock returns that can take advantage of the vast amount of conditioning information, keep a fully flexible form, and account for time variation. For this purpose, we combine three different deep neural network structures in a novel way: a feedforward network to capture nonlinearities, a recurrent (LSTM) network to find a small set of economic state processes, and a generative adversarial network to identify the portfolio strategies with the most unexplained pricing information. Our crucial innovation is the use of the no-arbitrage condition as part of the neural network algorithm. We estimate the stochastic discount factor that explains all stock returns from the conditional moment constraints implied by no arbitrage. Our SDF is a portfolio of all traded assets with time-varying portfolio weights that are general functions of the observable firm-specific and macroeconomic variables. Our model allows us to understand what are the key factors that drive asset prices, identify mispricing of stocks, and generate the conditional mean-variance efficient portfolio.

Our primary conclusions are threefold. First, our adversarial approach allows us to show and quantify the importance of including a no-arbitrage condition in the estimation of machine learning asset pricing models.

The “kitchen sink” prediction approach with deep learning does not outperform a linear model with no-arbitrage constraints. This illustrates that a successful use of machine learning methods in finance requires both subject-specific domain knowledge and a state-of-the-art technical implementation. Second, financial data have a time dimension that has to be taken into account accordingly. Even the most flexible model cannot compensate for the problem that macroeconomic data seems to be uninformative for asset pricing if only the last increments are used as input. We show that macroeconomic conditions matter for asset pricing and can be summarized by a small number of economic state variables, which depend on the complete dynamics of all time series. Third, asset pricing is actually surprisingly “linear.” As long as we consider anomalies in isolation, the linear factor models provide a good approximation. However, the multidimensional challenge of asset pricing cannot be solved with linear models and requires a different set of tools.

Our results have direct practical benefits for asset pricing researchers that go beyond our empirical findings. First, we provide a new set of benchmark test assets. New asset pricing models can be tested on explaining our SDF portfolio, respectively, the portfolios sorted according to the risk exposure in our model. These test assets incorporate the information of all characteristics and macroeconomic information in a small number of assets. Explaining portfolios sorted on a single characteristic is not a high hurdle to pass. Second, we provide a set of macroeconomic time series of hidden states that encapsulate the relevant macroeconomic information for asset pricing. These time series can also be used as an input for new asset pricing models.⁴⁶ Finally, our model is directly valuable for investors and portfolio managers. The main output of our model is the risk measure β and the SDF weight ω as a function of characteristics and macroeconomic variables. Given our estimates, the user of our model can assign a risk measure and its portfolio weight to an asset even if it does not have a long time series available.

Acknowledgments

The authors thank Doron Avramov, Ravi Bansal, Daniele Bianchi (discussant), Svetlana Bryzgalova, Agostino Capponi, Xiaohong Chen, Anna Cieslak, John Cochrane, Lin William Cong, Victor DeMiguel, Jens Dick-Nielsen (discussant), Kay Giesecke, Stefano Giglio, Goutham Gopalakrishna (discussant), Robert Hodrick, Bryan Kelly (discussant), Serhiy Kozak, Martin Lettau, Anton Lines, Marcelo Medeiros (discussant), Scott Murray (discussant), Stefan Nagel, Andreas Neuhierl (discussant), Kyungwon Seo (discussant), Gustavo Schwenkler, Neil Shephard, and Guofu Zhou as well as seminar and conference participants at Yale School of Management, Stanford, University of California, Berkeley, Washington University in St. Louis, Temple University, Imperial College London, University of Zurich, University of California, Los Angeles, Bremen University, Santa Clara University, King’s College London, the

Utah Winter Finance Conference, the Georgia State University–Review of Financial Studies FinTech Conference, the New Technology in Finance Conference, the London Business School Finance Summer Symposium, the Fourth International Workshop in Financial Econometrics, Triangle Macro-Finance Workshop, German Economists Abroad Annual Meeting, the Western Mathematical Finance Conference, INFORMS, SIAM Financial Mathematics, International Conference on Computational and Financial Econometrics, Shanghai Edinburgh Fintech Conference, Annual NLP and Machine Learning in Investment Management Conference, Annual Conference on Asia-Pacific Financial Markets, European Winter Meeting of Econometric Society, AI in Asset Management Day, Winter Research Conference on Machine Learning and Business, French Association of Asset and Liability Manager Conference, Midwest Finance Association Annual Meeting, Annual Meeting of the Swiss Society for Financial Market Research, Society for Financial Econometrics Annual Conference, and China Meeting of the Econometric Society for helpful comments. They thank the China Merchants Bank for generous research support. They gratefully acknowledge the best paper awards at the Utah Winter Finance Conference and Asia-Pacific Financial Markets Conference, the second place at the Chicago Quantitative Alliance Academic Paper Competition, and the honorable mention at the AQR Capital Insight Award.

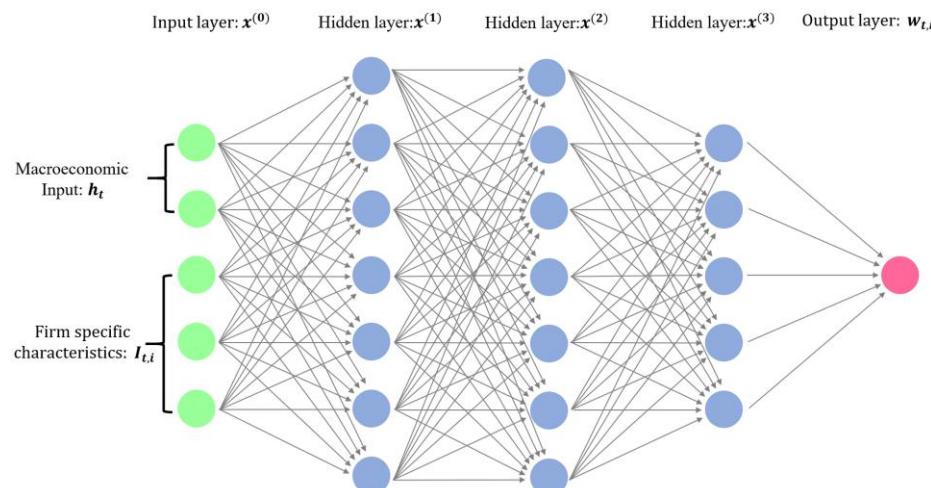
Appendix A. Estimation Method

A.1. FFN

The deep neural network considers L layers as illustrated in Figure A.1. Each hidden layer takes the output from the previous layer and transforms it into an output as

$$\begin{aligned} x^{(l)} &= \text{ReLU}\left(W^{(l-1)\top} x^{(l-1)} + w_0^{(l-1)}\right) \\ &= \text{ReLU}\left(w_0^{(l-1)} + \sum_{k=1}^{K^{(l-1)}} w_k^{(l-1)} x_k^{(l-1)}\right) \\ y &= W^{(L)\top} x^{(L)} + w_0^{(L)} \end{aligned}$$

Figure A.1. (Color online) Feedforward Network with Three Hidden Layers



with hidden layer outputs $x^{(l)} = (x_1^{(l)}, \dots, x_{K^{(l)}}^{(l)}) \in \mathbb{R}^{K^{(l)}}$ and parameters $W^{(l)} = (w_1^{(l)}, \dots, w_{K^{(l)}}^{(l)}) \in \mathbb{R}^{K^{(l)} \times K^{(l-1)}}$ for $l = 0, \dots, L - 1$ and $W^{(L)} \in \mathbb{R}^{K^{(L)}}$.

A.2. RNN

The LSTM is composed of a cell (the memory part of the LSTM unit) and three “regulators,” called gates, of the flow of information inside the LSTM unit: an input gate, a forget gate, and an output gate. Intuitively, the cell is responsible for keeping track of the dependencies between the elements in the input sequence. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell, and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit.

We take $x_t = I_t$ as the input sequence of macroeconomic information, and the output is the state processes h_t . At each step, a new memory cell \tilde{c}_t is created with current input x_t and previous hidden state h_{t-1} :

$$\tilde{c}_t = \tanh(W_h^{(c)} h_{t-1} + W_x^{(c)} x_t + w_0^{(c)}).$$

The input and forget gates control the memory cell, whereas the output gate controls the amount of information stored in the hidden state:

$$\begin{aligned} \text{input}_t &= \sigma(W_h^{(i)} h_{t-1} + W_x^{(i)} x_t + w_0^{(i)}) \\ \text{forget}_t &= \sigma(W_h^{(f)} h_{t-1} + W_x^{(f)} x_t + w_0^{(f)}) \\ \text{out}_t &= \sigma(W_h^{(o)} h_{t-1} + W_x^{(o)} x_t + w_0^{(o)}). \end{aligned}$$

The sigmoid function σ is an element-wise nonlinear transformation. Denoting the element-wise product by \circ , the final memory cell and hidden state are given by

$$c_t = \text{forget}_t \circ c_{t-1} + \text{input}_t \circ \tilde{c}_t, \quad h_t = \text{out}_t \circ \tanh(c_t).$$

We use the state processes h_t instead of the macroeconomic variables I_t as input to our SDF network.

A.3. Implementation

For training deep neural networks, the vanilla stochastic gradient descent method has proven to be not an efficient method.

A better approach is to use optimization methods that introduce an adaptive learning rate.⁴⁷ We use Adam, which is an algorithm for gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower order moments to continuously adjust the learning rate. It is more likely to escape saddle points and, hence, is more accurate and also providing faster convergence.

Regularization is crucial and prevents the model from overfitting on the training sample. Although l_1/l_2 regularization might also be used in training other neural networks, dropout is preferable and generally results in better performances. The term “dropout” refers to dropping out units in a neural network. By dropping out a unit, we mean temporarily removing it from the network along with all its incoming and outgoing connections with a certain probability. Dropout can be shown to be a form of ridge regularization and is only applied during the training.⁴⁸ When doing out-of-sample testing, we keep all the units and their connections.

In summary, the hyperparameter selection works as follows: (1) First, for each possible combination of hyperparameters (384 models), we fit the GAN model. (2) Second, we select the four best combinations of hyperparameters on the validation data set. (3) Third, for each of the four combinations, we fit nine models with the same hyperparameters but different initialization. (4) Finally, we select the ensemble model with the best performance on the validation data set. Table A.1 reports the tuning parameters of the best performing model. The feedforward network estimating the SDF weights has two hidden layers (HL), each of which has 64 nodes (HU). There are four hidden states (SMV) that summarize the macroeconomic dynamics in the LSTM network. The conditional adversarial network generates eight moments (CHU) in a zero-layer (CHL) network. The macroeconomic dynamics for the conditional moments are summarized in 32

hidden states (CSMV). This conditional network essentially applies a nonlinear transformation to the characteristics and the hidden macroeconomic states and then combines them linearly. The resulting moments can, for example, capture the pricing errors of long-short portfolios based on characteristic information or portfolios that only pay off under certain macroeconomic conditions. The FFN for the forecasting approach uses the optimal hyperparameters selected by Gu et al. (2020), which is a three-layer neural network with [32, 16, 8] hidden units, dropout retaining probability of 0.95, and a learning rate of 0.001.⁴⁹

Our GAN network is inspired by machine learning GANs as proposed in Goodfellow et al. (2014) but implemented differently and specifically designed for our problem. The machine learning GAN is usually implemented with deeper neural networks and a simultaneous optimization, that is, each network is only doing some optimization steps without completing the optimization before iterating with the other network. This is often necessary as the machine learning GANs are applied to huge data sets for which it would be computationally too expensive to solve each optimization completely. In contrast, we run each optimization step until convergence. For our benchmark model, this means in the first step we find an SDF to price all 10,000 stocks without instrumenting them. Then, we completely solve the adversarial problem to generate an eight-dimensional vector of instruments. In the third step, we completely estimate the SDF that can price all 80,000 instrumented stock returns. It turns out that this procedure is very stable and already converges after the first three steps in our empirical analysis as shown in Online Figure IA.14. In contrast, a conventional machine learning GAN typically creates smaller sets of adversarial data for computational reasons, which then also leads to more iterations to exhaust the information set.

Table A.1. Selection of Hyperparameters for GAN in the Empirical Analysis

Notation	Hyperparameters	Candidates	Optimal
HL	Number of layers in SDF network	2, 3, or 4	2
HU	Number of hidden units in SDF network	64	64
SMV	Number of hidden states in SDF network	4 or 8	4
CSMV	Number of hidden states in conditional network	16 or 32	32
CHL	Number of layers in conditional network	0 or 1	0
CHU	Number of hidden units in conditional network	4, 8, 16, or 32	8
LR	Initial learning rate	0.001, 0.0005, 0.0002, or 0.0001	0.001
DR	Dropout	0.95	0.95

Notes. This table shows the optimal tuning parameters of the benchmark GAN. The optimal SDF network has two layers each with 64 nodes and uses four hidden macroeconomic state processes. The optimal adversarial network is a generalized linear model creating eight instruments and uses 32 hidden macroeconomic states. The dropout parameter denotes the probability of keeping a node.

Appendix B. List of Firm-Specific Characteristics

Table B.1. Firm Characteristics by Category

		Past returns				Value
(1)	r2_1	Short-term momentum	(26)	A2ME	Assets to market cap	
(2)	r12_2	Momentum	(27)	BEME	Book to market ratio	
(3)	r12_7	Intermediate momentum	(28)	C	Ratio of cash and short-term investments to total assets	
(4)	r36_13	Long-term momentum	(29)	CF	Free cash flow to book value	
(5)	ST_Rev	Short-term reversal	(30)	CF2P	Cashflow to price	
(6)	LT_Rev	Long-term reversal	(31)	D2P	Dividend yield	
		Investment	(32)	E2P	Earnings to price	
(7)	Investment	Investment	(33)	Q	Tobin's Q	
(8)	NOA	Net operating assets	(34)	S2P	Sales to price	
(9)	DPI2A	Change in property, plants, and equipment	(35)	Lev	Leverage	
(10)	NI	Net share issues			Trading frictions	
		Profitability	(36)	AT	Total assets	
(11)	PROF	Profitability	(37)	Beta	CAPM beta	
(12)	ATO	Net sales over lagged net operating assets	(38)	IdioVol	Idiosyncratic volatility	
(13)	CTO	Capital turnover	(39)	LME	Size	
(14)	FC2Y	Fixed costs to sales	(40)	LTurnover	Turnover	
(15)	OP	Operating profitability	(41)	MktBeta	Market Beta	
(16)	PM	Profit margin	(42)	Rel2High	Closeness to past year high	
(17)	RNA	Return on net operating assets	(43)	Resid_Var	Residual variance	
(18)	ROA	Return on assets	(44)	Spread	Bid-ask spread	
(19)	ROE	Return on equity	(45)	SUV	Standard unexplained volume	
(20)	SGA2S	Selling, general and administrative expenses to sales	(46)	Variance	Variance	
(21)	D2A	Capital intensity				
		Intangibles				
(22)	AC	Accrual				
(23)	OA	Operating accruals				
(24)	OL	Operating leverage				
(25)	PCM	Price to cost margin				

Notes. This table shows the 46 firm-specific characteristics sorted into six categories. More details on the construction are in the online appendix.

Appendix C. Asset Pricing Results for Sorted Portfolios

Table C.1. Explained Variation and Pricing Errors for Decile Sorted Portfolios

	EN	FFN	GAN	EN	FFN	GAN	EN	FFN	GAN	EN	FFN	GAN
Short-term reversal												
Decile				Explained variation				Explained variation				Alpha
1	0.84	0.74	0.77	−0.18	−0.21	−0.13	0.04	−0.06	0.33	0.37	0.39	0.11
2	0.86	0.81	0.82	0.00	−0.05	0.00	0.12	0.10	0.52	0.25	0.18	−0.01
3	0.80	0.82	0.84	0.13	0.04	0.06	0.19	0.25	0.66	0.14	0.05	−0.06
4	0.69	0.80	0.82	0.16	0.03	0.03	0.28	0.34	0.73	0.15	0.08	−0.02
5	0.58	0.68	0.71	0.13	−0.03	−0.04	0.37	0.46	0.80	0.19	0.09	0.02
6	0.43	0.66	0.75	0.22	0.05	0.01	0.45	0.58	0.78	0.02	−0.03	−0.09
7	0.23	0.64	0.77	0.20	0.03	−0.02	0.62	0.69	0.68	0.01	0.01	−0.05
8	−0.07	0.49	0.67	0.23	0.03	−0.05	0.58	0.71	0.64	−0.03	−0.04	−0.09
9	−0.25	0.29	0.58	0.30	0.09	−0.01	0.55	0.70	0.58	0.08	0.04	−0.03
10	−0.24	−0.04	0.35	0.47	0.38	0.18	0.51	0.53	0.53	0.24	0.29	0.19
Explained variation				Cross-sectional R ²				Explained variation				Cross-sectional R ²
All	0.43	0.58	0.70	0.45	0.79	0.94	0.26	0.27	0.54	0.66	0.71	0.93
Book-to-market												
Decile				Explained variation				Explained variation				Alpha
1	0.38	0.66	0.70	0.03	−0.12	−0.08	0.80	0.75	0.79	0.09	−0.00	0.10
2	0.48	0.73	0.78	0.10	−0.05	−0.04	0.89	0.89	0.90	−0.11	−0.09	−0.06
												Size

Table C.1. (Continued)

	EN	FFN	GAN	EN	FFN	GAN	EN	FFN	GAN	EN	FFN	GAN
3	0.71	0.84	0.86	0.07	-0.03	-0.01	0.91	0.80	0.91	-0.07	0.02	-0.02
4	0.76	0.88	0.89	0.00	-0.07	-0.07	0.90	0.77	0.91	-0.05	0.04	-0.01
5	0.82	0.87	0.88	0.05	0.02	0.01	0.90	0.78	0.91	0.01	0.10	0.04
6	0.77	0.82	0.88	0.06	0.04	0.02	0.88	0.80	0.91	0.03	0.09	0.02
7	0.81	0.81	0.87	0.03	0.08	0.03	0.84	0.81	0.89	0.04	0.05	-0.01
8	0.71	0.59	0.78	0.03	0.12	0.06	0.84	0.85	0.88	0.06	0.03	-0.02
9	0.80	0.72	0.80	-0.02	0.11	0.07	0.77	0.81	0.82	0.06	-0.01	-0.04
10	0.68	0.73	0.79	-0.05	-0.00	0.00	0.32	0.28	0.49	-0.04	-0.15	-0.10
	Explained variation			Cross-sectional R^2			Explained variation			Cross-sectional R^2		
All	0.70	0.75	0.82	0.97	0.94	0.98	0.83	0.78	0.86	0.96	0.95	0.97

Note. This table shows the out-of-sample explained variation and pricing errors for decile-sorted portfolios based on short-term reversal (ST_REV), momentum (r12_2), book-to-market ratio (BEME), and size (LME).

Table C.2. Explained Variation and Pricing Errors for Double-Sorted Portfolios Based on Short-Term Reversal/Momentum and Size/Book-to-Market Ratio

Short-term reversal and momentum						Size and book-to-market ratio							
ST_REV / LME	r12_2 / BEME	EN	FFN	GAN	EN	FFN	GAN	EN	FFN	GAN	EN	FFN	GAN
		Explained variation			Alpha			Explained variation			Alpha		
1	1	0.35	0.32	0.62	0.16	0.13	0.08	0.55	0.47	0.63	-0.01	-0.00	-0.06
1	2	0.55	0.48	0.72	-0.02	-0.04	-0.05	0.66	0.62	0.74	0.01	0.00	-0.04
1	3	0.66	0.61	0.74	-0.06	-0.07	-0.05	0.74	0.70	0.76	0.04	0.01	0.01
1	4	0.74	0.62	0.67	-0.06	-0.05	-0.02	0.77	0.69	0.75	0.01	-0.02	0.01
1	5	0.69	0.58	0.58	-0.10	-0.06	-0.03	0.70	0.66	0.76	-0.01	-0.03	0.02
2	1	0.17	0.16	0.53	0.22	0.19	0.11	0.58	0.20	0.68	0.01	0.11	-0.02
2	2	0.32	0.39	0.67	0.18	0.11	0.08	0.68	0.48	0.81	0.02	0.07	-0.01
2	3	0.59	0.61	0.71	0.08	0.03	0.01	0.82	0.74	0.86	0.04	0.06	0.03
2	4	0.72	0.74	0.59	0.00	-0.03	-0.02	0.81	0.75	0.85	-0.03	-0.00	-0.01
2	5	0.56	0.61	0.54	0.08	0.05	0.06	0.77	0.79	0.85	-0.04	0.00	0.02
3	1	-0.02	-0.01	0.48	0.18	0.16	0.01	0.53	0.25	0.73	0.08	0.12	0.02
3	2	0.13	0.33	0.65	0.12	0.02	-0.03	0.70	0.59	0.85	0.10	0.11	0.05
3	3	0.41	0.62	0.66	0.13	0.02	-0.00	0.86	0.82	0.90	0.06	0.08	0.05
3	4	0.46	0.60	0.48	0.03	-0.06	-0.07	0.86	0.82	0.88	0.01	0.05	0.02
3	5	0.39	0.53	0.42	0.08	-0.01	-0.02	0.79	0.76	0.81	-0.04	0.02	0.01
4	1	-0.24	-0.27	0.31	0.26	0.24	0.06	0.53	0.50	0.79	0.12	0.09	0.01
4	2	-0.24	0.15	0.58	0.14	0.05	-0.04	0.74	0.78	0.85	0.07	0.04	-0.00
4	3	0.02	0.51	0.68	0.11	-0.02	-0.06	0.80	0.84	0.83	0.05	0.02	0.00
4	4	0.19	0.53	0.51	0.11	-0.01	-0.04	0.83	0.81	0.85	0.02	0.03	0.01
4	5	0.17	0.47	0.51	0.14	0.02	-0.01	0.73	0.77	0.79	-0.05	-0.02	-0.01
5	1	-0.58	-0.88	0.08	0.13	0.17	-0.08	0.28	0.29	0.44	0.01	-0.09	-0.06
5	2	-0.41	-0.12	0.42	0.14	0.06	-0.06	0.54	0.53	0.58	0.00	-0.08	-0.05
5	3	-0.28	0.23	0.53	0.16	0.03	-0.03	0.51	0.56	0.57	-0.01	-0.04	-0.04
5	4	-0.06	0.31	0.44	0.12	-0.00	-0.05	0.54	0.60	0.67	-0.01	-0.00	-0.02
5	5	-0.01	0.27	0.36	0.29	0.16	0.11	0.37	0.52	0.56	-0.04	-0.03	-0.03
	Explained variation			Cross-sectional R^2			Explained variation			Cross-sectional R^2			
All		0.20	0.26	0.53	0.50	0.77	0.92	0.67	0.60	0.76	0.94	0.91	0.98

Note. This table shows the out-of-sample explained variation and pricing errors for double sorted portfolios based on short-term reversal (ST_REV) and momentum (r12_2), respectively, size (LME) and book-to-market ratio (BEME).

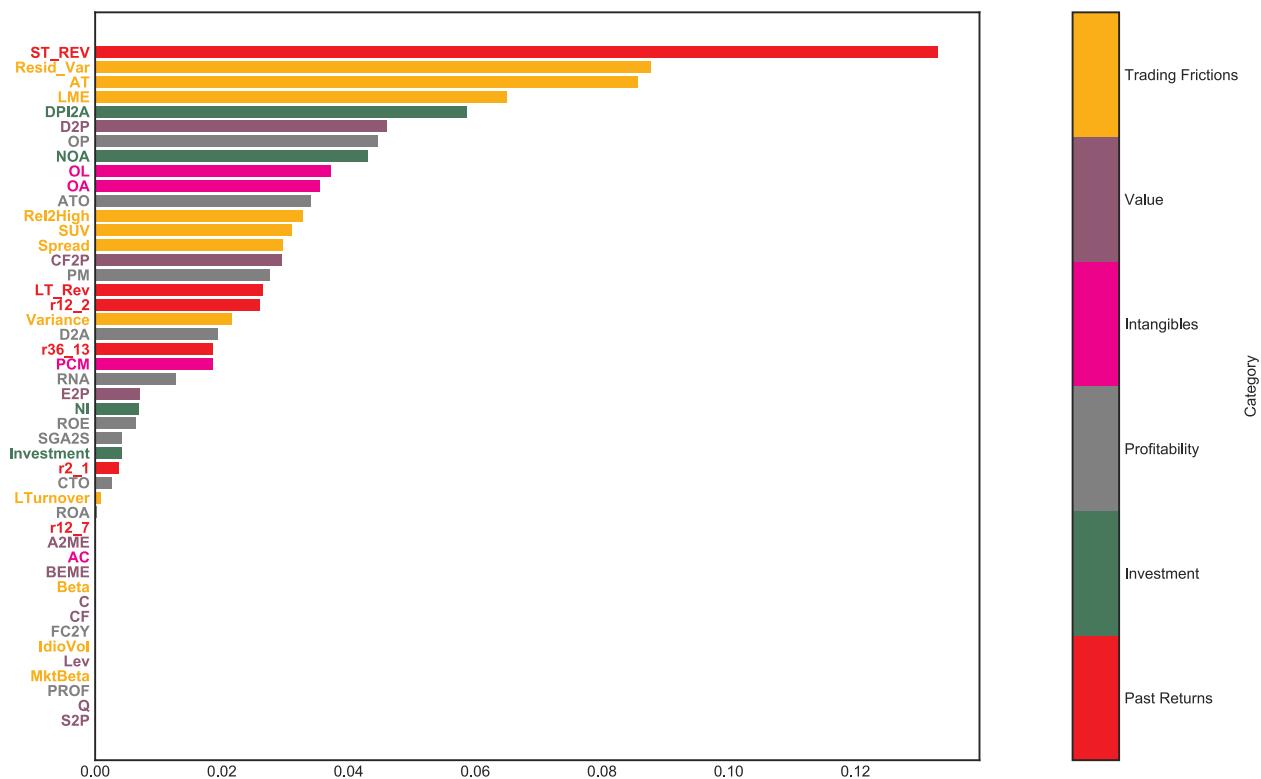
Appendix D. Variable Importance

Figure D.1. (Color online) Macroeconomic Variable Importance for GAN SDF



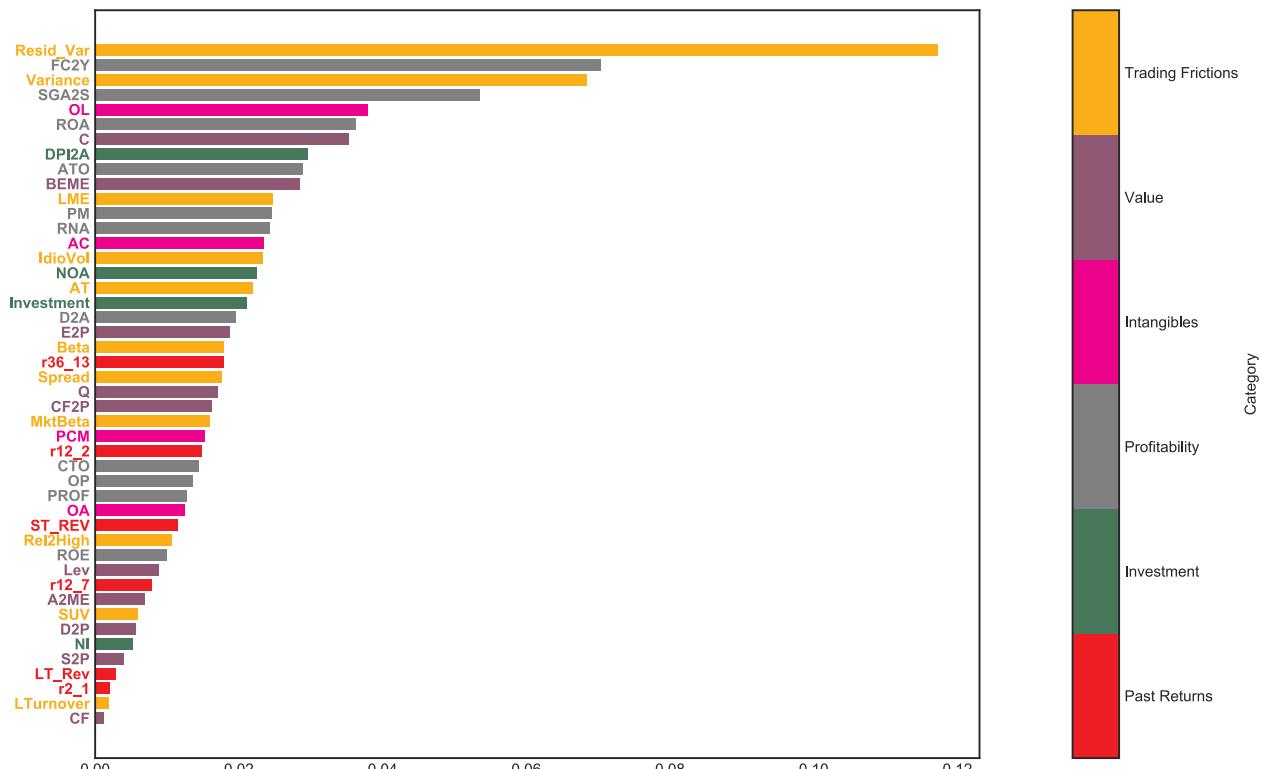
Notes. This figure shows the GAN variable importance ranking of the 178 macroeconomic variables in terms of average absolute gradient on the test data. The values are normalized to sum up to one.

Figure D.2. (Color online) Characteristic Importance for EN



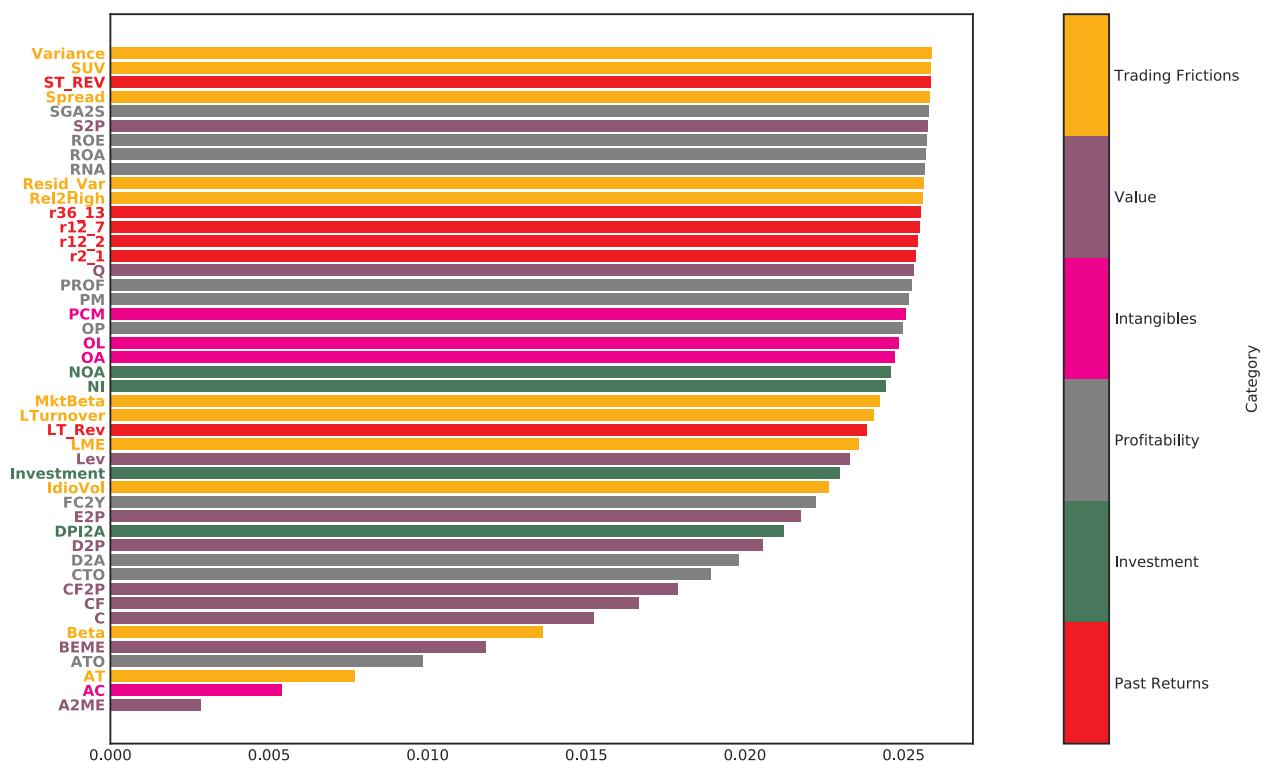
Notes. The figure shows the elastic net variable importance ranking of the 46 firm-specific characteristics in terms of average absolute gradient on the test data. The values are normalized to sum up to one.

Figure D.3. (Color online) Characteristic Importance for LS



Notes. The figure shows the LS variable importance ranking of the 46 firm-specific characteristics in terms of average absolute gradient on the test data. The values are normalized to sum up to one.

Figure D.4. (Color online) Characteristic Importance for Conditioning Function g for GAN



Notes. The figure shows the GAN variable importance ranking for the conditioning function g of the 46 firm-specific characteristics in terms of average absolute gradient on the test data. The ranking is the average of the absolute gradients for the nine ensemble fits. The values are normalized to sum up to one.

Appendix E. SDF Structure

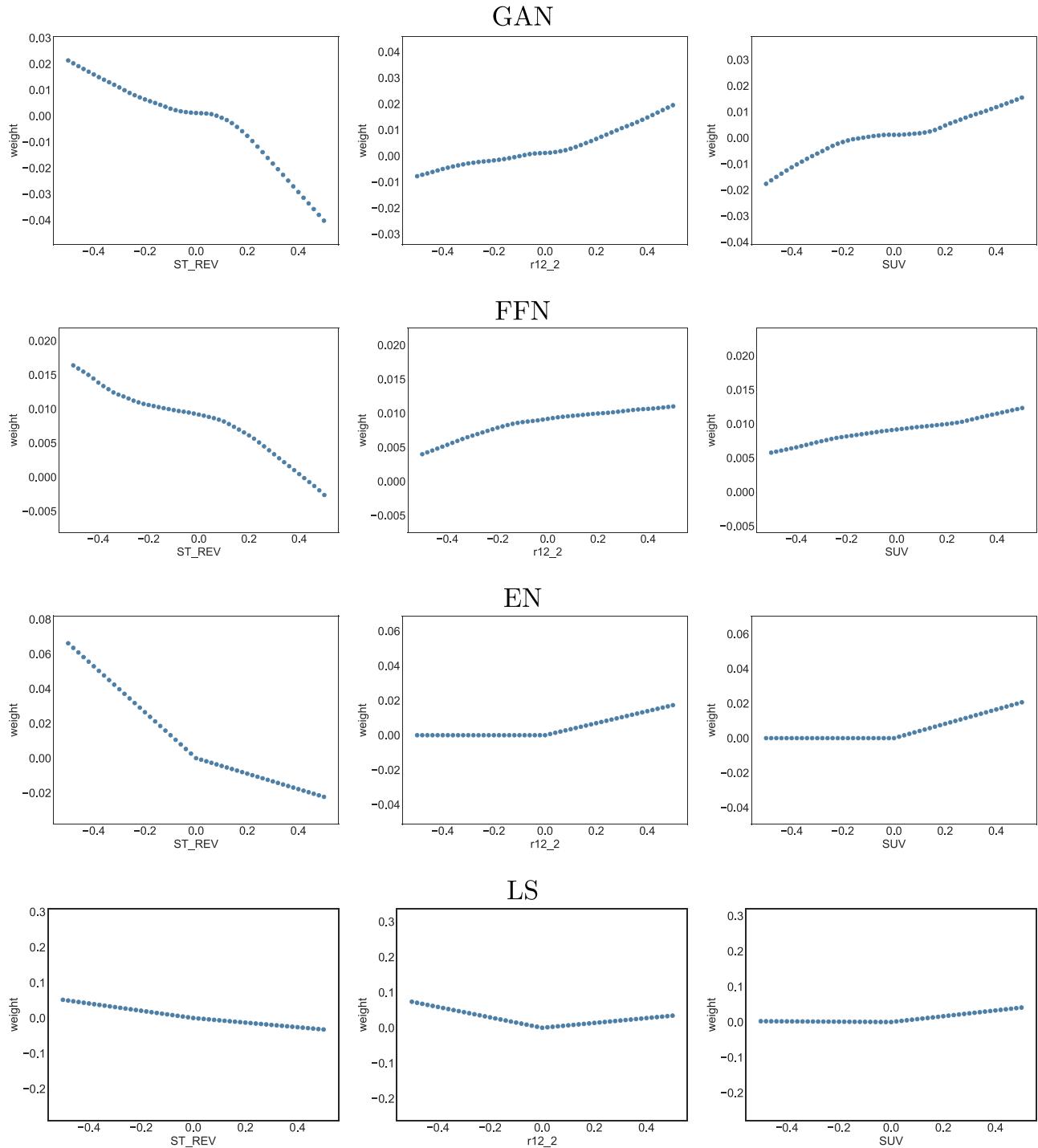
Table E.1. Correlation of GAN SDF with Fama–French Five Factors and Time Series Regression

	Mkt-RF	SMB	HML	RMW	CMA	Intercept
Regression coefficients	0.00 (0.02)	0.00 (0.02)	-0.04 (0.03)	0.08*** (0.03)	0.04 (0.04)	0.76*** (0.06)
Correlations	-0.10	-0.09	0.01	0.17	0.05	—

Notes. This table shows out-of-sample correlations and the time series regression of the GAN SDF on the Fama–French five factors. Standard errors are in parentheses. The regression intercept is the monthly time series pricing error of the SDF portfolio.

* , **, and *** correspond to 10%, 5%, and 1% significance levels, respectively.

Figure E.1. (Color online) SDF Weight ω as a Function of Characteristics for Different Models



Notes. This figure shows the SDF weight ω as a one-dimensional function of covariates keeping the other covariates at their mean level. The covariates are short-term reversal (ST_REV), momentum (r12_2), and standard unexplained volume (SUV).

Endnotes

¹ These models include Gu et al. (2020), Messmer (2017), and Kelly et al. (2019).

² Other related work includes Sadhwani et al. (2021), who estimate mortgage prepayments, delinquencies, and foreclosures with deep neural networks; Moritz and Zimmerman (2016), who apply tree-based models to portfolio sorting; and Heaton et al. (2017), who automate portfolio selection with a deep neural network. Horel and Giesecke (2020) propose a significance test in neural networks and apply it to house price valuation. Rossi (2018) uses boosted regression trees to form conditional mean-variance efficient portfolios based on the market portfolio and the risk-free asset. Our approach also yields the conditional mean-variance efficient portfolio but based on all stocks.

³ The intuition behind their and our approach can be best understood when considering the linear special cases. Our approach can be viewed as a conditional, nonlinear generalization of Kozak et al. (2020) with the additional elements of finding the macroeconomic states and identifying the most robust conditioning instruments. Fundamentally, our object of interest is the pricing kernel. Kelly et al. (2019) obtain a multifactor factor model that maximizes the explained variation. The linear special case applies PCA to a set of characteristic-based factors to obtain a linear lower dimensional factor model, whereas their more general autoencoder obtains the loadings to characteristic-based factors that can depend nonlinearly on the characteristics. We show in Online Appendix IA.B how our SDF framework and their conditional multifactor framework can be combined to obtain an even better asset pricing model.

⁴ We shared our data and estimated models with Avramov et al. (2022). In their comparison study, Avramov et al. (2022) also include a portfolio derived from our GAN model. However, they do not consider our SDF portfolio based on ω , but use the SDF loadings β to construct a long-short portfolio based on prediction quantiles. Similarly, they use extreme quantiles of a forecasting approach with neural networks to construct long-short portfolios, which is again different from our SDF framework. Thus, they study different portfolios.

⁵ We show that the special case of a linear formulation of our model is essentially a version of their model, and we include it as the linear benchmark case in our analysis.

⁶ Lewis and Syrgkanis (2018) provide conditions under which the identified set is characterized as a solution to a zero-sum game between a modeler and an adversary. They also show that, under some conditions, the training can be done using stochastic gradient descent, which is the basis for training deep neural networks. Furthermore, they connect the equilibrium of an empirical game with the population game in terms of the Rademacher complexity of the hypothesis space.

⁷ Cong et al. (2021) provide an overview of sequence models including LSTM for return prediction.

⁸ See Back (2010) for more details. As we work with excess returns, we have an additional degree of freedom. Following Cochrane (2003), we use the preceding normalized relationship between the SDF and the mean-variance efficient portfolio. We consider the SDF based on the projection on the asset space.

⁹ Any portfolio on the globally efficient frontier achieves the maximum Sharpe ratio. These portfolio weights represent one possible efficient portfolio.

¹⁰ Denote the conditional residual covariance matrix by $\Sigma_t^\epsilon = \text{Var}_t(\epsilon_t)$. Then, sufficient conditions are $\|\Sigma_t^\epsilon\|_2 < \infty$ and $\frac{\beta^T \beta}{N} > 0$ for $N \rightarrow \infty$, that is, Σ_t^ϵ has bounded eigenvalues and β_t has sufficiently many nonzero elements. Importantly, the dependency in the residuals is irrelevant for our SDF estimation. For our estimator, we obtain the SDF weights $\omega_{t,i}$ and the stock loadings $\beta_{t,j}$. Neither step requires a weak dependency in the residuals. For the asset pricing analysis, we

separate the return space into the part spanned by loadings and its orthogonal complement, which does not impose assumptions on the covariance matrix of the residuals.

¹¹ No arbitrage requires the conditional moment equations implied by the law of one price and a strictly positive SDF. Our estimation is based on the conditional moments without directly enforcing the positivity of the SDF. Note that, in our empirical analysis, our estimated SDF is always positive on the in-sample data and, hence, does not require the additional positivity constraint.

¹² In more detail, Hansen and Jagannathan (1997) study the minimax estimation problem for the SDF in general (infinite dimensional) Hilbert spaces. Minimizing the pricing error is equivalent to minimizing the distance between the empirical pricing functional and an admissible true unknown pricing functional. The Riesz representation theorem provides a mapping between the pricing functional and the SDF, and hence, minimizing the worst pricing error implies convergence to an admissible SDF in a specific norm. Hence, our SDF is the solution to problems (1) and (2) in Hansen and Jagannathan (1997). The minimax framework to estimate or evaluate an SDF is also used in Bakshi and Chen (1997), Chen and Knez (1995), and Bansal et al. (1993) among others.

¹³ See Blanchet et al. (2019) for a discussion on robust estimation with an adversarial approach.

¹⁴ The number of test assets limits the number of parameters and, hence, the complexity of the SDF that is identified in each step of the iterative optimization. Our approach yields $D \cdot N$ test assets, which allows us to fit a complex structure for the SDF with few iteration steps. Obviously, instrumenting each stock with g only leads to more test assets if the resulting portfolios are not redundant. Empirically, we observe that there is a large variation in the vectors of characteristics cross-sectionally and over time. For example, if g is an indicator function for small-cap stocks, the returns of $R_{t+1,j}^\epsilon g(I_t, I_{t,j})$ provide different information for different stocks j as the other characteristics are in most cases not identical and stocks have a small market capitalization at different times t .

¹⁵ Kozak et al. (2020) consider also cross-products of the characteristics. They show that the PCA rotation of the factors improves the pricing performance. Lettau and Pelger (2020) extend this important insight to RP-PCA rotated factors. We consider PCA-based factors in Online Appendix IA.B. Our main analysis focuses on conventional long-short factors as these are the most commonly used models in the literature.

¹⁶ As before, we define as a tangency portfolio one of the portfolios on the global mean-variance efficient frontier.

¹⁷ FFN are among the simplest neural networks and treated in detail in standard machine learning textbooks, for example, Goodfellow et al. (2016).

¹⁸ Other activation functions include sigmoid, hyperbolic tangent function, and leaky ReLU. ReLU activation functions have a number of advantages including the nonsaturation of its gradient, which greatly accelerates the convergence of stochastic gradient descent compared with the sigmoid/hyperbolic functions (Krizhevsky et al. (2012) and fast calculations of expensive operations.

¹⁹ A conventional GAN network iterates this procedure until convergence. We find that our algorithm converges already after the preceding three steps, that is, the model does not improve further by repeating the adversarial game. Detailed results on the GAN iterations for the empirical analysis are in Online Figure IA.14.

²⁰ We allow for potentially different macroeconomic states for the SDF and the conditional network as the unconditional moment conditions that identify the SDF can depend on different states than the SDF weights.

²¹ An ensemble over nine models produces very robust and stable result, and there is no effect of averaging over more models. The results are available upon request.

²² We have used different criteria functions, including the error in minimizing the moment conditions, to select the hyperparameters. The results are virtually identical and available upon request.

²³ The elastic net includes lasso and ridge regularization as a special case. We select the tuning parameters of the elastic net optimally on the validation data.

²⁴ There are five differences to their paper. First, they use a modified ridge penalty based on a Bayesian prior. Second, they also include product terms of the characteristics. Third, their second moment matrix uses demeaned returns, that is, the two approaches choose different mean-variance efficient portfolios on the globally efficient frontier. Fourth, we allow for different linear weights on the long and short legs of the characteristic based factors. Fifth, they advocate to first apply PCA to the characteristic managed factors before solving the mean-variance optimization with elastic net penalty. Lettau and Pelger (2020) generalize the robust SDF recovery to the RP-PCA space. Bryzgalova et al. (2019) also include additional mean shrinkage in the robust SDF recovery and propose decision trees as an alternative to PCA. Bryzgalova et al. (2019) also show that mean-variance optimization with regularization can be interpreted as an adversarial approach with parameter uncertainty. We use conventional univariate factors as a benchmark as those are the most commonly used linear models in the literature. PCA-based methods are deferred to Online Appendix IA.B.

²⁵ The tangency portfolio weights are obtained on the training data set and used on the validation and test data set.

²⁶ We weight the estimated means by their rate of convergence to account for the differences in precision.

²⁷ Note, that $XS-R^2 = 1 - \sum_{i=1}^N \hat{\alpha}_i^2$.

²⁸ Pelger and Xiong (2021a) provide the theoretical arguments and show empirically in a linear setup why “proximate” factors that only capture the extreme factor weights correctly have similar time series properties as the population factors, but their portfolio weights are not the correct loadings.

²⁹ We use the characteristics that Freyberger et al. (2020) use in the 2017 version of their paper.

³⁰ Using stocks with missing characteristic information requires data imputation based on model assumptions. Gu et al. (2021) replace a missing characteristic with the cross-sectional median of that characteristic during that month. However, this approach introduces an additional source of error and ignores the dependency structure in the characteristic space and, thus, creates artificial time-series fluctuation in the characteristics, which we want to avoid. Hence, our main analysis follows the same approach as in Freyberger et al. (2020) and Kelly et al. (2019) and uses only stocks that have all firm characteristics available in a given month, which has the additional benefit of removing predominantly stocks with a very small market capitalization. Note that we do not require for one stock to have the characteristics to exist throughout its entire time series. We simply only include the returns at time t for stock i if, for this point in time, it has all characteristics; that is, stock i does not necessarily have a complete time series, which is allowed in our approach. In the online appendix, we show that the main findings continue to hold when we impute the missing characteristic values.

³¹ Kelly et al. (2019) and Kozak et al. (2020) construct factors in this way.

³² In the first version of this paper, we used the conventional long-short factors. However, our empirical results suggest that the long and short legs have different weights in the SDF, and this additional flexibility improves the performance of the linear model. These findings are also in line with Lettau and Pelger (2020), who extract linear factors from the extreme deciles of single-sorted portfolios and show that they are not spanned by long-short factors that put equal weight on the extreme deciles of each characteristic.

³³ Note that the sign of g is not identified, and we could multiply g by -1 and obtain the same output.

³⁴ The results are not affected by normalizing the SDF weights to have $\|\omega\|_1 = 1$. The explained variation and pricing results are based on a cross-sectional projection at each time step, which is independent of any scaling.

³⁵ Pesaran and Timmermann (1996), among others, show the time variation in risk premia.

³⁶ Our estimated loadings are only proportional to the SDF β_t . At each time t , we scale the loadings such that the SDF portfolio has a loading of one. This is equivalent to obtaining the correctly scaled β_t without explicitly calculating the second conditional moment of the SDF.

³⁷ The online appendix collects the results for additional characteristic sorts with similar findings. Here, we report only the results for value-weighted portfolios. The results for equally weighted portfolios are similar. The results for the unregularized linear model are the worst and available upon request.

³⁸ The results for PCA-based macroeconomic factors are available upon request. We also want to clarify that, for other applications, PCA-based factors based on macroeconomic time series might actually capture the relevant information.

³⁹ Each ensemble fit returns a four-dimensional vector of the state processes. However, it is not meaningful to average these vectors as the first state process in one fit does not need to correspond to the same process in another fit. It is only meaningful to report model averages of scalar output variables.

⁴⁰ NBER-based recession indicators for the United States from the peak through the trough are taken from <https://fred.stlouisfed.org/series/USRECM>.

⁴¹ As the characteristics are normalized to quantiles, their mean is equal to their median value.

⁴² As the linear model with regularization removes variables, it is possible that the SDF weights for one characteristic conditioned on different quantiles of the second characteristic collapse to one line.

⁴³ The online appendix collects the detailed results. Max drawdown is defined as the maximum number of consecutive months with negative returns. The maximum one-month loss is normalized by the standard deviation of the asset.

⁴⁴ Gu et al. (2020) report high out-of-sample Sharpe ratios for long-short portfolios based on the extreme quantiles of returns predicted by FFN. The online appendix compares the Sharpe ratios for different extreme quantiles for equally and value-weighted long-short portfolios with FFN. We can replicate the high out-of-sample Sharpe ratios when using extreme deciles of 10% or less and equally weighted portfolios. However, for value-weighted portfolios, the Sharpe ratio drops by around 50%. This is a clear indication that the performance of these portfolios heavily depends on small stocks.

⁴⁵ In their comparison study, Avramov et al. (2022) also include a portfolio derived from GAN. However, they do not consider our SDF portfolio based on ω , but use the SDF loadings β to construct a long-short portfolio based on prediction quantiles.

⁴⁶ The data are available on <https://mpelger.people.stanford.edu/research>.

⁴⁷ See, for example, Ruder (2016) and Kingma and Ba (2014). We use the leading algorithm Adam. Other adaptive gradient descent methods include Adagrad or Adadelta.

⁴⁸ See, for example, Srivastava et al. (2014) for the better performance results for dropout and Wager et al. (2013) for the connection with ridge.

⁴⁹ We estimate our models on two GPU clusters in which each cluster has two Intel Xeon E5-2698 v3 CPUs, 1 TB memory, and 8 Nvidia Titan V GPUs. We use TensorFlow with Python 3.6 for the model fitting. A complete estimation of the GAN model with hyperparameter tuning takes around three days. We confirm that our estimation results are

robust to using a larger hyperparameter space. As a full hyperparameter search on a larger hyperparameter space can easily take weeks or months even on our fast GPU cluster, we selectively test further hyperparameters.

References

- Arjovsky M, Chintala S, Leon B (2017) Wasserstein GAN. *Proc. 34th Internat. Conf. Machine Learn.*
- Avramov D, Cheng S, Metzker L (2022) Machine learning versus economic restrictions: Evidence from stock return predictability. *Management Sci.*, ePub ahead of print June 29, <https://doi.org/10.1287/mnsc.2022.4449>.
- Back K (2010) *Asset Pricing and Portfolio Choice* (Oxford University Press, Oxford, UK).
- Bakshi GS, Chen Z (1997) The spirit of capitalism and stock market prices. *Amer. Econom. Rev.* 86(1):133–157.
- Bansal R, Viswanathan S (1993) No arbitrage and arbitrage pricing: A new approach. *J. Finance* 48(4):1231–1262.
- Bansal R, Hsieh DA, Viswanathan S (1993) A new approach to international arbitrage pricing. *J. Finance* 48(5):1719–1747.
- Bianchi D, Tamoni A, Buchner M (2021) Bond risk premiums with machine learning. *Rev. Financial Stud.* 34(2):1046–1089.
- Blanchet J, Kang Y, Murthy K (2019) Robust Wasserstein profile inference and applications to machine learning. *J. Appl. Probab.* 56(3):830–857.
- Bryzgalova S, Pelger M, Zhu J (2019) Forest through the trees: Building cross-sections of stock returns. Working paper, Stanford University, Stanford, CA.
- Chamberlain G (1987) Asymptotic efficiency in estimation with conditional moment restrictions. *J. Econometrics* 34(3):305–334.
- Chamberlain G, Rothschild M (1983) Arbitrage, factor structure, and mean-variance analysis on large asset markets. *Econometrica* 51(5):1281–1304.
- Chen X, Ludvigson S (2009) Land of addicts? An empirical investigation of habit-based asset pricing models. *J. Appl. Econometrics* 24(7):1057–1093.
- Chen Z, Knez PJ (1995) Measurement of market integration and arbitrage. *Rev. Financial Stud.* 8(2):287–325.
- Cochrane J (2003) *Asset Pricing* (Princeton University Press, Princeton, NJ).
- Cong LW, Tang K, Wang J, Zhang Y (2020) AlphaPortfolio for investment and economically interpretable AI. Working paper, Cornell University, Ithaca, NY.
- Cong LW, Tang K, Wang J, Zhang Y (2021) Deep sequence modeling: Development and applications in asset pricing. *J. Financial Data Sci.* 3(1):28–42.
- Fama EF, French KR (1992) The cross-section of expected stock returns. *J. Finance* 47(2):427–465.
- Fama EF, French KR (1993) Common risk factors in the returns on stocks and bonds. *J. Financial Econom.* 33(1):3–56.
- Fama EF, French KR (2015) A five-factor asset pricing model. *J. Financial Econom.* 116(1):1–22.
- Feng G, He J, Polson NG (2018) Deep learning for predicting asset returns. Working paper, University of Chicago, Chicago.
- Feng G, Polson NG, Xu J (2019) Deep learning in characteristics-sorted factor models. Working paper, University of Chicago, Chicago.
- Freyberger J, Neuhierl A, Weber M (2020) Dissecting characteristics nonparametrically. *Rev. Financial Stud.* 33(5):2326–2377.
- Goodfellow I, Bengio Y, Courville A (2016) *Deep Learning* (MIT Press).
- Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. *Adv. Neural Inform. Processing Systems* 27:2672–2680.
- Gu S, Kelly BT, Xiu D (2020) Empirical asset pricing via machine learning. *Rev. Financial Stud.* 33(5):2223–2273.
- Gu S, Kelly B, Xiu D (2021) Autoencoder asset pricing models. *J. Econometrics* 222(1):429–450.
- Guijarro-Ordonez J, Pelger M, Zanotti G (2019) Deep learning statistical arbitrage. Working paper, Stanford University, Stanford, CA.
- Hansen LP (1982) Large sample properties of generalized method of moments estimators. *Econometrica* 50(4):1029–1054.
- Hansen LP, Jagannathan R (1997) Assessing specification errors in stochastic discount factor models. *J. Finance* 52(2):557–590.
- Heaton J, Polson N, Witte JH (2017) Deep learning for finance: Deep portfolios. *Appl. Stochastic Models Bus. Indust.* 33(1):3–12.
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput.* 9(8):1735–1780.
- Horel E, Giesecke K (2020) Towards explainable AI: Significance tests for neural networks. *J. Machine Learn. Res.* 21(1):9291–9319.
- Kelly B, Pruitt S, Su Y (2019) Characteristics are covariances: A unified model of risk and return. *J. Financial Econom.* 134(3):501–524.
- Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. Preprint, submitted December 22, <https://arxiv.org/abs/1412.6980>.
- Kozak S, Nagel S, Santosh S (2020) Shrinking the cross section. *J. Financial Econom.* 135(2):271–292.
- Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. *Proc. Adv. Neural Inform. Processing Systems* (Curran Associates Inc., Red Hook, NY), 25.
- Lettau M, Pelger M (2020) Factors that fit the time-series and cross-section of stock returns. *Rev. Financial Stud.* 33(5):2274–2325.
- Lewis G, Syrigkanis V (2018) Adversarial generalized method of moments. Working paper, Stanford University, Stanford, CA.
- Ludvigson S, Ng S (2007) The empirical risk return relation: A factor analysis approach. *J. Financial Econom.* 83(1):171–222.
- Ludvigson S, Ng S (2009) Macro factors in bond risk premia. *Rev. Financial Stud.* 22(12):5027–5067.
- Martin I, Nagel S (2022) Market efficiency in the age of big data. *J. Financial Econom.* 145(1):154–177.
- McCracken MW, Ng S (2016) FRED-MD: A monthly database for macroeconomic research. *J. Bus. Econom. Statist.* 34(4):574–589.
- Messmer M (2017) Deep learning and the cross-section of expected returns. Working paper, University of St. Gallen, St. Gallen, Switzerland.
- Moritz B, Zimmerman T (2016) Tree-based conditional portfolio sorts: The relation between past and future stock returns. Working paper, University of Cologne, Cologne, Germany.
- Nagel S, Singleton KJ (2011) Estimation and evaluation of conditional asset pricing models. *J. Finance* 66(3):873–909.
- Pelger M (2020) Understanding systematic risk: A high-frequency approach. *J. Finance* 75(4):2179–2220.
- Pelger M, Xiong R (2021a) Interpretable sparse proximate factors for large dimensions. *J. Bus. Econom. Statist.* 40(4):1642–1664.
- Pelger M, Xiong R (2021b) State-varying factor models of large dimensions. *J. Bus. Econom. Statist.* 40(3):1315–1333.
- Pesaran HM, Timmermann A (1996) Predictability of stock returns: Robustness and economic significance. *J. Finance* 50(4):1201–1228.
- Ross SA (1976) The arbitrage theory of capital asset pricing. *J. Econom. Theory* 13:341–360.
- Rossi AG (2018) Predicting stock market returns with machine learning. Working paper, Georgetown University, Washington, DC.
- Ruder S (2016) An overview of gradient descent optimization algorithms. Preprint, submitted September 15, <https://arxiv.org/abs/1609.04747>.
- Sadhwani A, Sirignano J, Giesecke K (2021) Deep learning for mortgage risk. *J. Financial Econom.* 19(2):313–368.
- Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: A simple way to prevent neural networks from overfitting. *J. Machine Learn. Res.* 15(56):1929–1958.
- Wager S, Wang S, Liang PS (2013) Dropout training as adaptive regularization. *Adv. Neural Inform. Processing Systems* 26:351–359.
- Welch I, Goyal A (2007) A comprehensive look at the empirical performance of equity premium prediction. *Rev. Financial Stud.* 21(4):1455–1508.