# Deep Learning in Asset Pricing

Luyang Chen † Markus Pelger ‡ Jason Zhu ‡

†Institute for Computational & Mathematical Engineering, Stanford University
‡Department of Management Science & Engineering, Stanford University

March 7, 2019
Doctoral Seminar

# Hype: Machine Learning in Investment

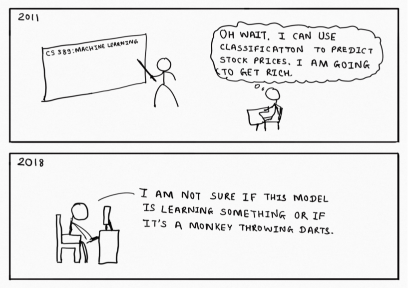
**The U.S. Stock Market Belongs to Bots**

Bloomberg, July 15, 2017

Same reporter 3 weeks later

**The Quant Fund Robot Takeover Has Been Postponed**

Bloomberg, Aug 9, 2017




- Efficient markets: Asset returns dominated by unforecastable news

$\Rightarrow$ Financial return data has very low signal-to noise ratio

$\Rightarrow$ This paper: Including financial constraints (no-arbitrage) in learning algorithm significantly improves signal

# Motivation: Asset Pricing

## The Challenge of Asset Pricing

- One of the most important questions in finance:

  **Why are asset prices different for different assets?**

- **No-Arbitrage Pricing Theory**: **Stochastic discount factor SDF** (also called pricing kernel or equivalent martingale measure) explains differences in risk and asset prices
- Fundamental question: What is the SDF?
- Challenges:
  - SDF should depend on all available economic information: Very large set of variables
  - Functional form of SDF unknown and likely complex
  - SDF needs to capture time-variation in economic conditions
  - Risk premium in stock returns has a low signal-to-noise ratio

# This paper

## Goals of this paper:

General non-linear asset pricing model and optimal portfolio design

$\Rightarrow$ Deep-neural networks applied to all U.S. equity data and large sets of macroeconomic and firm-specific information.

## Why is it important?

1. Stochastic discount factor (SDF) generates tradeable portfolio with highest risk-adjusted return (Sharpe-ratio=expected excess return/standard deviation)
2. Arbitrage opportunities
   - Find underpriced assets and earn "alpha"
3. Risk management
   - Understand which information and how it drives the SDF
   - Manage risk exposure of financial assets

# Contribution of this paper

## Contribution

- This Paper: Estimate the SDF with deep neural networks
- Crucial innovation: Include no-arbitrage condition in the neural network algorithm and combine four neural networks in a novel way
- Key elements of estimator:
  1. Non-linearity: Feed-forward network captures non-linearities
  2. Time-variation: Recurrent (LSTM) network finds a small set of economic state processes
  3. Pricing all assets: Generative adversarial network identifies the states and portfolios with most unexplained pricing information
  4. Dimension reduction: Regularization through no-arbitrage condition
  5. Signal-to-noise ratio: No-arbitrage conditions increase the signal to noise-ratio

$\Rightarrow$ General model that includes all existing models as a special case

# Contribution of this paper

## Empirical Contributions

- Empirically outperforms all benchmark models.
- Optimal portfolio has out-of-sample annual Sharpe ratio of 2.1.
- Non-linearities and interaction between firm information matters.
- Most relevant firm characteristics are price trends, profitability, and capital structure variables.

# Literature (Partial List)

- Deep-learning for predicting asset prices
  - Feng, Polson and Xu (2019)
  - Gu, Kelly and Xiu (2018)
  - Feng, Polson and Xu (2018)
  - Messmer (2017)
  
  ⇒ Predicting future asset returns with feed forward network
  - Gu, Kelly and Xiu (2019)
  - Heaton, Polson and Witte (2017)
  
  ⇒ Fitting asset returns with an autoencoder
- Linear or kernel methods for asset pricing of large data sets
  - Lettau and Pelger (2018): Risk-premium PCA
  - Feng, Giglio and Xu (2017): Risk-premium lasso
  - Freyberger, Neuhierl and Weber (2017): Group lasso
  - Kelly, Pruitt and Su (2018): Instrumented PCA

# The Model

## No-arbitrage pricing

- $R^e_{i,t+1}$ = excess return (return minus risk-free rate) at time $t+1$ for asset $i = 1, ..., N$
- Fundamental no-arbitrage condition:
  for all $t = 1, ..., T$ and $i = 1, ..., N$

$$\mathbb{E}_t[M_{t+1} R^e_{i,t+1}] = 0$$

  - $\mathbb{E}_t[.]$ expected value conditioned on information set at time $t$
  - $M_{t+1}$ stochastic discount factor SDF at time $t+1$.
- Conditional moments imply infinitely many unconditional moments

$$\mathbb{E}[M_{t+1} R^e_{t+1,i} I_t] = 0$$

  for any $\mathcal{F}_t$-measurable variable $I_t$

# The Model

## No-arbitrage pricing

- Without loss of generality SDF is projection on the return space

$$M_{t+1} = 1 - \sum_{i=1}^{N} w_{i,t} R^e_{i,t+1}$$

$\Rightarrow$ Optimal portfolio $\sum_{i=1}^{N} w_{i,t} R^e_{i,t+1}$ has highest conditional Sharpe-ratio

- Portfolio weights $w_{i,t}$ are a general function of macro-economic information $I_t$ and firm-specific characteristics $I_{i,t}$:

$$w_{i,t} = w(I_t, I_{i,t}),$$

$\Rightarrow$ Need non-linear estimator with many explanatory variables!

$\Rightarrow$ Use a feed forward network to estimate $w_{i,t}$

# The Model

## Equivalent factor model representation

- No-arbitrage condition is equivalent to

$$\mathbb{E}_t[R^e_{i,t+1}] = \frac{\text{cov}_t(R^e_{i,t+1}, F_{t+1})}{\text{var}_t(F_{t+1})} \cdot \mathbb{E}_t[F_{t+1}] = \beta_{i,t}\mathbb{E}_t[F_{t+1}]$$

with factor $F_t = 1 - M_t$.

$\Rightarrow$ Without loss of generality we have a factor representation

$$R^e_{t+1} = \beta_t F_{t+1} + \epsilon_{t+1}.$$

# The Model

## Objects of Interest

We use different approaches to estimate:

- The SDF factor $F_t$
- The risk loadings $\beta_t$
- The unexplained residual $\hat{e}_t = (I_N - \beta_{t-1}(\beta_{t-1}^\top \beta_{t-1})^{-1}\beta_{t-1}^\top) R_t^e$

## Asset Pricing Performance Measure

- Sharpe ratio of SDF factor: $SR = \frac{\hat{\mathbb{E}}[F_t]}{\sqrt{\widehat{Var}(F_t)}}$
- Explained variation: $EV = 1 - \frac{\left(\frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t}(\hat{\epsilon}_{i,t+1})^2\right)}{\left(\frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t}(R_{i,t+1}^e)^2\right)}$
- cross-sectional mean $R^2$: XS-$R^2$ $= 1 - \frac{\frac{1}{N}\sum_{i=1}^{N}\frac{T_i}{T}\left(\frac{1}{T_i}\sum_{t\in T_i}\hat{\epsilon}_{i,t+1}\right)^2}{\frac{1}{N}\sum_{i=1}^{N}\frac{T_i}{T}\left(\frac{1}{T_i}\sum_{t\in T_i}\hat{R}_{i,t+1}\right)^2}$

# Loss Function

## Objective Function for Estimation

- Estimate SDF portfolio weights $w(.)$ to minimize the no-arbitrage moment conditions
- For a set of conditioning variables $\hat{I}_t$ the loss function is

$$L(\hat{I}_t) = \frac{1}{N}\sum_{i=1}^{N}\frac{T_i}{T}\left(\frac{1}{T_i}\sum_{t=1}^{T_i} M_{t+1}R^e_{i,t+1}\hat{I}_t\right)^2.$$

- Allows unbalanced panel.
- How can we choose the conditioning variables $\hat{I}_t = f(I_t, I_{i,t})$ as general functions of the macroeconomic and firm-specific information?

$\Rightarrow$ Generative Adversarial Network (GAN) chooses $\hat{I}_t$!

# Generative Adversarial Network (GAN)

## Determining Moment Conditions

- Two networks play zero-sum game:
  1. one network creates the SDF $M_{t+1}$
  2. other network creates the conditioning variables $\hat{I}_t$
- Iteratively update the two networks:
  1. for a given $\hat{I}_t$ the SDF network minimizes the loss
  2. for a given SDF the conditional networks finds $\hat{I}_t$ with the largest loss (most mispricing)

$\Rightarrow$ Intuition: find the economic states and assets with the most pricing information

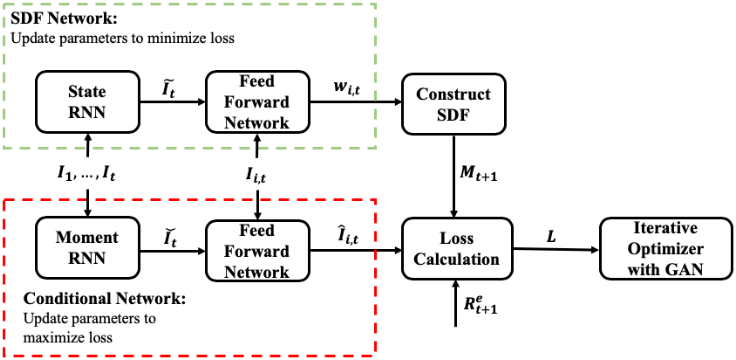
# Recurrent Neural Network (RNN)

## Transforming Macroeconomic Time-Series

- **Problems** with economic time-series data
  - Time-series data is often non-stationary $\Rightarrow$ transformation necessary
  - Business cycles can affect pricing $\Rightarrow$ assuming Markovian structure of the pricing kernel not sufficient
  - Redundant information $\Rightarrow$ large number of predictors prove to negatively impact model performance
- **Solution**: Recurrent Neural Network (RNN) with Long-Short-Term Memory (LSTM) cells
- Transform all macroeconomic time-series into a low dimensional vector of stationary state variables

# Neural Networks

## Model Architecture



# Data

## Data

- 50 years of monthly observations: 01/1967 - 12/2016.
- Monthly stock returns for all U.S. securities from CRSP (around 31,000 stocks)
  Use only stocks with with all firm characteristics (around 10,000 stocks)
- 46 firm-specific characteristics for each stock and every month (usual suspects) $\Rightarrow I_{i,t}$
  normalized to cross-sectional quantiles
- 178 macroeconomic variables
  (124 from FRED, 46 cross-sectional median time-series for characteristics, 8 from Goyal-Welch) $\Rightarrow I_t$
- T-bill rates from Kenneth-French website
- Training/validation/test split is 20y/5y/25y

# Benchmark models

## Benchmark models

1. LS & EN - Linear factor models:
   The optimal portfolio weights $w_t = I_t\theta$ is linear in characteristics. We minimize loss function
   $$\frac{1}{2}\left\|\frac{1}{T}\tilde{R}^{K\top}1 - \frac{1}{T}\tilde{R}^{K\top}\tilde{R}^{K}\theta\right\|_2^2 + \lambda_1\|\theta\|_1 + \frac{1}{2}\lambda_2\|\theta\|_2^2.$$
   $\tilde{R}^K_{t+1} = I_t^\top R^e_{t+1}$ are $K$ portfolios weighted by characteristics $I_t$.
2. FFN - Deep learning return forecasting (Gu et al. (2018)):
   - Predict conditional expected returns $\mathbb{E}_t[R_{i,t+1}]$
   - Empirical loss function for prediction
   $$\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left(R_{i,t+1} - g(I_t, I_{i,t})\right)^2$$
   - Use only simple feedforward network for forecasting

# Results - Cross Section of Individual Stock Returns

Table: Performance of Different SDF Models

| | SR | | | EV | | | Cross-Sectional $R^2$ | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | Train | Valid | Test | Train | Valid | Test | Train | Valid | Test |
| LS | 1.35 | 0.80 | 0.45 | 0.09 | 0.04 | 0.03 | 0.03 | 0.04 | 0.02 |
| EN | 1.01 | 0.95 | 0.47 | 0.15 | 0.07 | 0.06 | 0.04 | 0.07 | 0.04 |
| FFN | 0.30 | 0.28 | 0.36 | 0.16 | 0.07 | 0.06 | 0.01 | 0.05 | 0.05 |
| GAN | 3.26 | 0.97 | 0.60 | 0.21 | 0.10 | 0.08 | 0.01 | 0.05 | 0.05 |

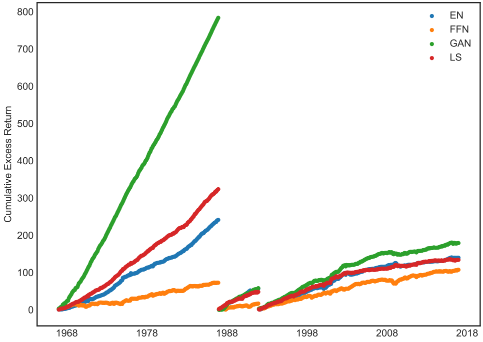
# Results - Cross Section of Individual Stock Returns

Table: SDF Factor Portfolio Performance

| | SR | | | Max Loss | | | Max Drawdown | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | Train | Valid | Test | Train | Valid | Test | Train | Valid | Test |
| FF-3 | 0.27 | -0.09 | 0.19 | -2.45 | -2.85 | -4.31 | 7 | 10 | 10 |
| FF-5 | 0.48 | 0.40 | 0.22 | -2.62 | -2.33 | -4.90 | 4 | 3 | 7 |
| LS | 1.35 | 0.80 | 0.45 | -1.82 | -1.50 | -3.67 | 2 | 2 | 7 |
| EN | 1.01 | 0.95 | 0.47 | -3.22 | -2.21 | -5.99 | 2 | 3 | 6 |
| FFN | 0.30 | 0.28 | 0.36 | -3.88 | -4.93 | -4.07 | 7 | 4 | 5 |
| GAN | 3.26 | 0.97 | 0.60 | -0.09 | -1.01 | -4.48 | 1 | 2 | 3 |

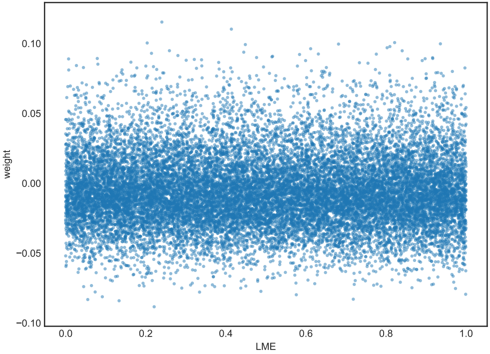
# Results - Cross Section of Individual Stock Returns

Figure: Cumulated Normalized SDF Portfolio



# Results - Size Effect

Figure: GAN SDF Weight $\omega$ and Size (LME)




$\Rightarrow$ SDF portfolio is not predominantly investing in small stocks.

# Results - Sharpe Ratio for Forecasting Approach

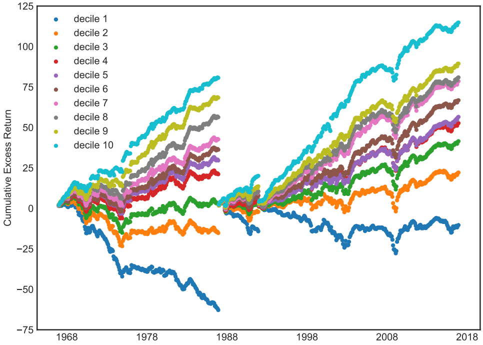
Table: Sharpe Ratio of Long-Short Portfolios with FFN

| Quantile | SR (Train) | SR (Valid) | SR (Test) |
|---|---|---|---|
| | (i) Equally-Weighted | | |
| 1% | 1.08 | 0.75 | 0.65 |
| 5% | 1.26 | 1.15 | 0.70 |
| 10% | 1.11 | 1.22 | 0.65 |
| 25% | 1.03 | 1.20 | 0.56 |
| 50% | 0.96 | 1.16 | 0.54 |
| | (ii) Value-Weighted | | |
| 1% | 0.77 | 0.55 | 0.41 |
| 5% | 0.79 | 0.77 | 0.39 |
| 10% | 0.59 | 0.46 | 0.32 |
| 25% | 0.46 | 0.09 | 0.19 |
| 50% | 0.42 | 0.23 | 0.18 |

$\Rightarrow$ Long-short portfolio is based on extreme quantiles.

# Results - Predictive Performance

Figure: Cumulative Excess Return of Decile Sorted Portfolios by GAN




⇒ Risk loading predicts future stock returns.

# Results - Decile Sorted Portfolios

Table: Explained Variation and Pricing Errors for Short-Term Reversal Sorted Portfolios

| ST_REV | Explained Variation (EV) | | | Cross-Sectional $R^2$ | | |
|---|---|---|---|---|---|---|
| Decile | Elastic Net | FFN | GAN | Elastic Net | FFN | GAN |
| 1 | 0.91 | 0.92 | 0.91 | 0.96 | 0.96 | 0.96 |
| 2 | 0.95 | 0.96 | 0.95 | 0.89 | 0.94 | 0.96 |
| 3 | 0.94 | 0.96 | 0.95 | 0.94 | 0.95 | 0.96 |
| 4 | 0.93 | 0.93 | 0.93 | 0.96 | 0.95 | 0.94 |
| 5 | 0.91 | 0.92 | 0.91 | 1.00 | 0.99 | 0.96 |
| 6 | 0.85 | 0.88 | 0.92 | 0.96 | 0.99 | 0.99 |
| 7 | 0.69 | 0.78 | 0.88 | 0.84 | 0.93 | 1.00 |
| 8 | 0.48 | 0.61 | 0.81 | 0.63 | 0.80 | 0.99 |
| 9 | 0.19 | 0.32 | 0.64 | 0.25 | 0.43 | 0.91 |
| 10 | -0.03 | -0.11 | 0.29 | -0.05 | -0.47 | 0.68 |
| Overall | 0.70 | 0.72 | 0.81 | 0.87 | 0.89 | 0.95 |

Explained variation and pricing errors for decile-sorted portfolios based on Short-Term Reversal (ST_REV).

# Results - Decile Sorted Portfolios

Table: Explained Variation and Pricing Errors for Momentum Sorted Portfolios

| r12_2 | Explained Variation (EV) | | | Cross-Sectional $R^2$ | | |
|---|---|---|---|---|---|---|
| Decile | Elastic Net | FFN | GAN | Elastic Net | FFN | GAN |
| 1 | 0.22 | 0.25 | 0.48 | 0.29 | 0.30 | 0.71 |
| 2 | 0.49 | 0.52 | 0.72 | 0.73 | 0.82 | 0.98 |
| 3 | 0.68 | 0.73 | 0.86 | 0.90 | 0.97 | 1.00 |
| 4 | 0.81 | 0.85 | 0.91 | 0.95 | 1.00 | 0.99 |
| 5 | 0.89 | 0.90 | 0.92 | 1.00 | 1.00 | 0.98 |
| 6 | 0.92 | 0.90 | 0.89 | 1.00 | 0.99 | 0.98 |
| 7 | 0.91 | 0.89 | 0.86 | 0.99 | 0.99 | 0.98 |
| 8 | 0.88 | 0.88 | 0.84 | 0.98 | 0.99 | 0.99 |
| 9 | 0.84 | 0.85 | 0.82 | 0.99 | 1.00 | 1.00 |
| 10 | 0.80 | 0.79 | 0.77 | 1.00 | 0.99 | 0.99 |
| Overall | 0.61 | 0.63 | 0.73 | 0.86 | 0.87 | 0.95 |

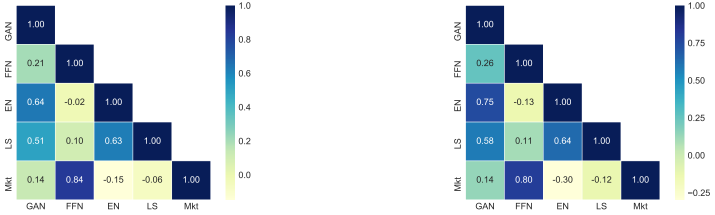
Explained variation and pricing errors for decile-sorted portfolios based on Momentum (r12_2).

# Results - ST_REV and r12_2 Double Sorted Portfolios

| ST_REV | r12_2 | Explained Variation (EV) | | | Cross-Sectional $R^2$ | | |
|---|---|---|---|---|---|---|---|
| | | Elastic Net | FFN | GAN | Elastic Net | FFN | GAN |
| 1 | 1 | 0.58 | 0.70 | 0.77 | 0.74 | 0.88 | 0.92 |
| 1 | 2 | 0.85 | 0.86 | 0.88 | 0.99 | 1.00 | 1.00 |
| 1 | 3 | 0.90 | 0.91 | 0.89 | 0.95 | 0.95 | 0.97 |
| 1 | 4 | 0.85 | 0.89 | 0.87 | 0.95 | 0.98 | 1.00 |
| 1 | 5 | 0.80 | 0.86 | 0.83 | 0.93 | 0.99 | 1.00 |
| 2 | 1 | 0.48 | 0.54 | 0.68 | 0.84 | 0.91 | 0.98 |
| 2 | 2 | 0.79 | 0.81 | 0.87 | 1.00 | 0.99 | 0.97 |
| 2 | 3 | 0.87 | 0.86 | 0.83 | 0.97 | 0.93 | 0.93 |
| 2 | 4 | 0.80 | 0.83 | 0.77 | 0.93 | 0.94 | 0.95 |
| 2 | 5 | 0.79 | 0.82 | 0.80 | 0.90 | 0.96 | 0.98 |
| 3 | 1 | 0.24 | 0.26 | 0.53 | 0.45 | 0.54 | 0.92 |
| 3 | 2 | 0.60 | 0.69 | 0.82 | 0.92 | 1.00 | 0.95 |
| 3 | 3 | 0.81 | 0.83 | 0.82 | 0.98 | 0.99 | 0.95 |
| 3 | 4 | 0.86 | 0.85 | 0.76 | 1.00 | 0.99 | 0.96 |
| 3 | 5 | 0.78 | 0.77 | 0.73 | 1.00 | 1.00 | 0.98 |
| 4 | 1 | -0.13 | -0.22 | 0.21 | -0.55 | -0.61 | 0.66 |
| 4 | 2 | 0.20 | 0.41 | 0.69 | 0.45 | 0.89 | 0.95 |
| 4 | 3 | 0.54 | 0.71 | 0.82 | 0.79 | 0.99 | 0.97 |
| 4 | 4 | 0.72 | 0.80 | 0.80 | 0.90 | 0.99 | 0.99 |
| 4 | 5 | 0.68 | 0.67 | 0.71 | 0.93 | 0.95 | 1.00 |
| 5 | 1 | -0.51 | -0.81 | -0.17 | -4.71 | -15.39 | 0.97 |
| 5 | 2 | -0.17 | -0.06 | 0.36 | -0.45 | -0.16 | 0.88 |
| 5 | 3 | 0.18 | 0.38 | 0.63 | 0.34 | 0.59 | 0.91 |
| 5 | 4 | 0.35 | 0.44 | 0.57 | 0.73 | 0.86 | 0.99 |
| 5 | 5 | 0.43 | 0.44 | 0.56 | 0.73 | 0.75 | 0.89 |
| Overall | | 0.49 | 0.52 | 0.65 | 0.83 | 0.88 | 0.96 |

# Results - SDF Factors for Different Models

Figure: Correlation between SDF Factors for Different Models




(a) Whole Time Horizon

(b) Test Period

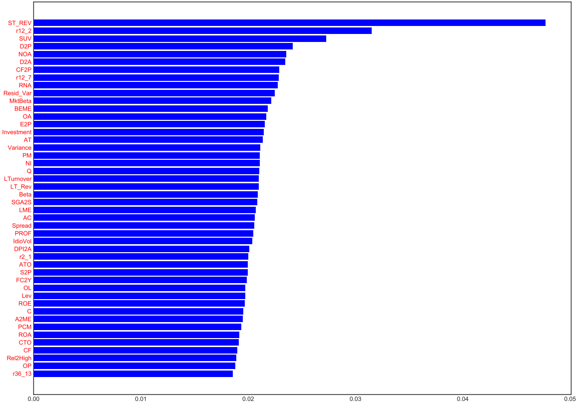
# Results - SDF Factor and Fama-French Factors

Table: GAN-SDF Factor and Fama-French 5 Factors

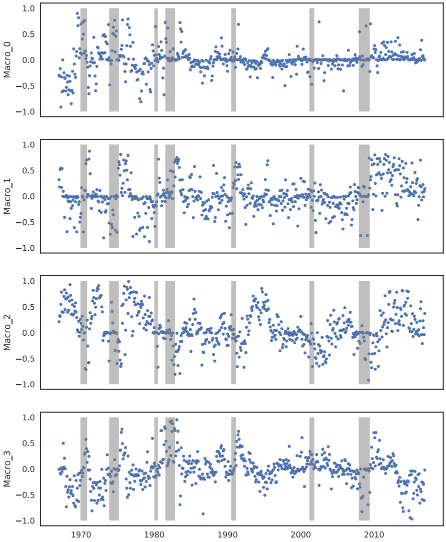
| | Mkt-RF | SMB | HML | RMW | CMA | intercept |
|---|---|---|---|---|---|---|
| Regression Coefficients | 0.07*** | 0.01 | 0.03 | 0.13*** | -0.01 | 0.38*** |
| | (0.01) | (0.02) | (0.02) | (0.02) | (0.03) | (0.04) |
| Correlation | 0.14 | -0.11 | 0.23 | 0.31 | 0.04 | - |

Out-of-sample correlation and regression of GAN SDF factor on the Fama-French 5 factors. The regression intercept is the monthly time-series pricing error of the SDF portfolio for the Fama-French model. Standard errors are in parenthesis.

# Results - Characteristic Importance by GAN



# Results - Macroeconomic Hidden State Processes



# Results - SDF Weights

## Relationship between Weights and Characteristics

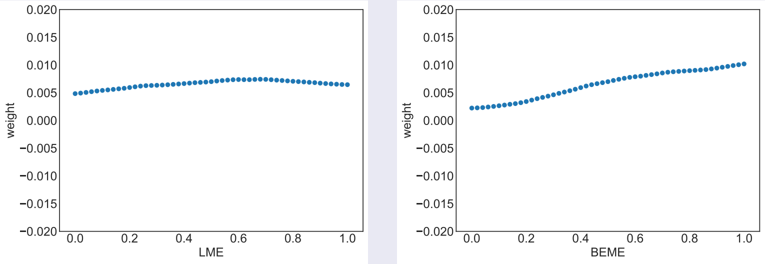



Figure: Weight as a function of Size (LME) and Book-to-Market Ratio (BEME)

⇒ Size and value have close to linear effect

# Results - SDF Weights

## Relationship between Weights and Characteristics

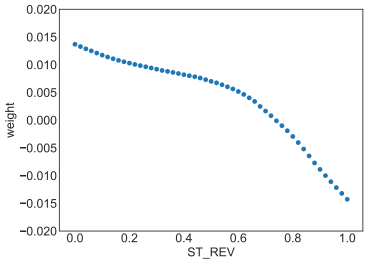



Figure: Weight as a function of Short-Term Reversal (ST_REV)

$\Rightarrow$ non-linear effect

# Results - SDF Weights

## Relationship between Weights and Characteristics

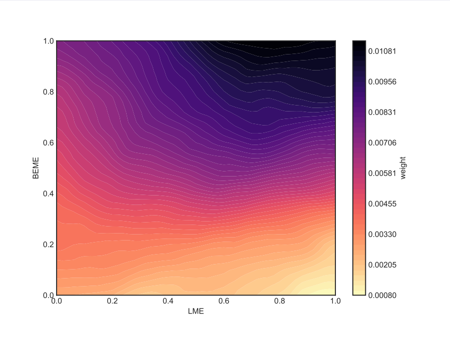



Figure: Weight as a function of Size (LME) and Book-to-Market Ratio (BEME)

$\Rightarrow$ Size and value have non-linear interaction!

# Results - SDF Weights

## Relationship between Weights and Characteristics

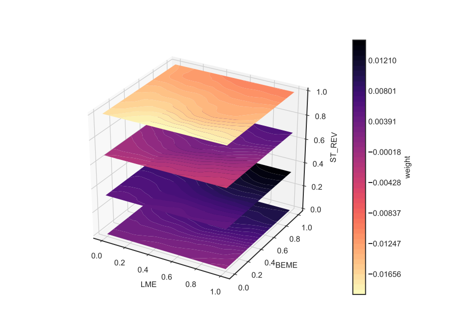



Figure: Weight as a function of Size (LME), Book-to-Market Ratio (BEME) and Short-Term Reversal (ST_REV).

$\Rightarrow$ Complex interaction between multiple variables!

# Simulation Setup

## Motivation

We illustrate with simulations that

- the no-arbitrage condition in GAN is necessary to find the SDF in a low-signal to noise setup
- the flexible form of GAN is necessary to correctly capture the interactions between characteristics
- the LSTM-RNN is necessary to correctly incorporate macroeconomic dynamics in the pricing kernel

# Simulation Setup

## Setup

- Excess returns follow a no-arbitrage model with SDF factor $F$

$$R^e_{i,t+1} = \beta_{i,t} F_{t+1} + \epsilon_{i,t+1}.$$

- The SDF factor follows $F_t \overset{i.i.d.}{\sim} \mathcal{N}(\mu_F, \sigma_F^2)$.
- The idiosyncratic component $\epsilon_{i,t} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_e^2)$.
- $N = 500$ and $T = 600$. Define training/validation/test=250,100,250.
- The SDF factor has $\sigma_F^2 = 0.1$ and $SR_F = 1$. The idiosyncratic noise variance $\sigma_e^2 = 1$.

# Simulation Setup

## Setup

We consider two different formulations for the risk loadings

1. Two characteristics:

$$\beta_{i,t} = C_{i,t}^{(1)} \cdot C_{i,t}^{(2)} \qquad \text{with} \qquad C_{i,t}^{(1)}, C_{i,t}^{(2)} \overset{i.i.d.}{\sim} \mathcal{N}(0,1).$$

2. One characteristic and one macroeconomic state process:

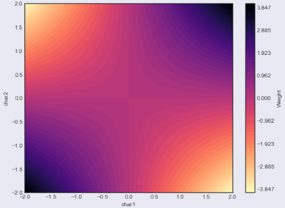
$$\beta_{i,t} = C_{i,t}^{(1)} \cdot b(h_t), \qquad h_t = sin(\pi * t/24) + \epsilon_t^h.$$

$$b(h) = \begin{cases} 1 & \text{if } h > 0 \\ -1 & \text{otherwise.} \end{cases}$$

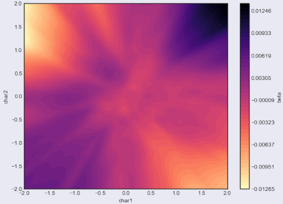
We observe only the macroeconomic time-series $Z_t = \mu_M t + h_t$. All innovations are independent and normally distributed: $C_{i,t}^{(1)} \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ and $\epsilon_t^h \overset{i.i.d.}{\sim} \mathcal{N}(0,0.25)$.

# Simulation Results - Setup I

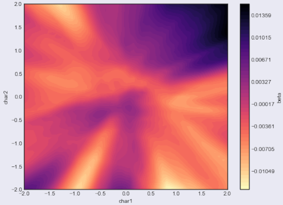
## Loadings $\beta$ with 2 characteristics



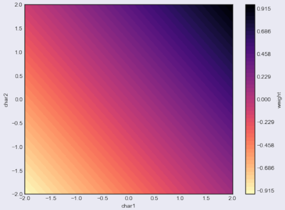

(a) Population Model




(b) GAN




(c) FFN




(d) LS

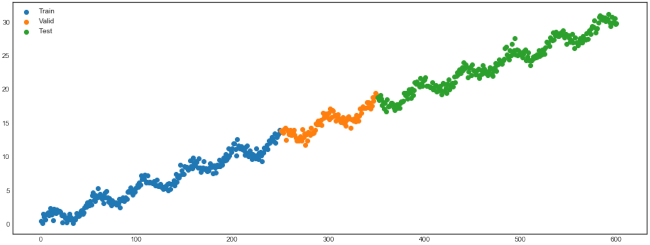
# Simulation Results - Setup I

Table: Performance of Different SDF Models

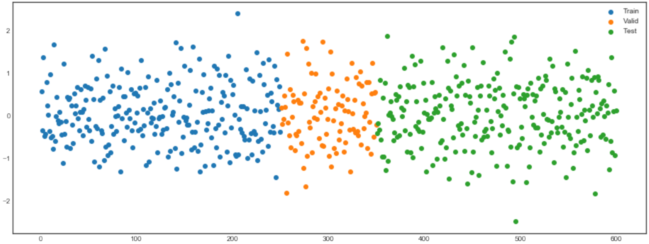
| | Sharpe Ratio | | | EV | | | Cross-sectional $R^2$ | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | Train | Valid | Test | Train | Valid | Test | Train | Valid | Test |
| Population | 0.96 | 1.09 | 0.94 | 0.16 | 0.15 | 0.17 | 0.17 | 0.15 | 0.17 |
| GAN | 0.98 | 1.11 | 0.94 | 0.12 | 0.11 | 0.13 | 0.10 | 0.09 | 0.07 |
| FFN | 0.94 | 1.04 | 0.89 | 0.05 | 0.04 | 0.05 | -0.30 | -0.09 | -0.33 |
| LS | 0.07 | -0.10 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.01 |

# Simulation Results - Setup II

## Observed Macroeconomic Variable

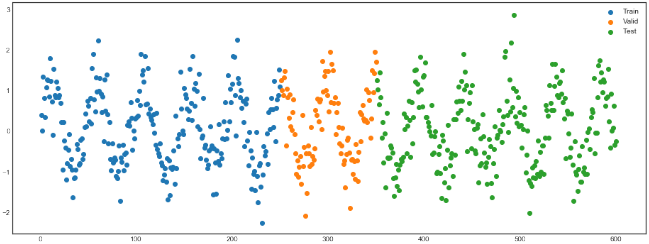



## First order difference of Macroeconomic Variable

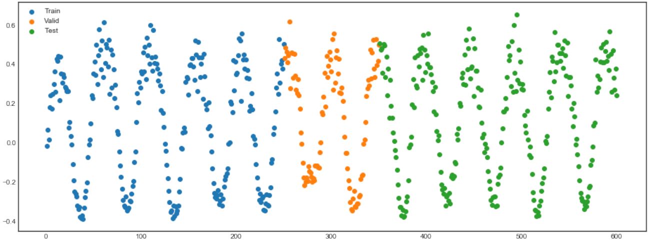


# Simulation Results - Setup II

True hidden Macroeconomic State




Fitted Macroeconomic State by LSTM



# Simulation Results - Setup II

Table: Performance of Different SDF Models

| | Sharpe Ratio | | | EV | | | Cross-sectional $R^2$ | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | Train | Valid | Test | Train | Valid | Test | Train | Valid | Test |
| Population | 0.89 | 0.92 | 0.86 | 0.18 | 0.18 | 0.17 | 0.19 | 0.20 | 0.15 |
| GAN | 0.79 | 0.77 | 0.64 | 0.18 | 0.18 | 0.17 | 0.19 | 0.20 | 0.15 |
| FFN | 0.05 | -0.05 | 0.06 | 0.02 | 0.01 | 0.02 | 0.01 | 0.01 | 0.02 |
| LS | 0.12 | -0.05 | 0.10 | 0.16 | 0.16 | 0.15 | 0.15 | 0.18 | 0.14 |

# Conclusion

## Summary

- Linear models perform well because when considering characteristics in isolation, the models are approximately linear.
- Non-linearities matter for the interaction.
- Most relevant variables are price trends and liquidity.
- Macroeconomic data has a low dimensional factor structure.
- Pricing all individual stocks leads to better pricing models on portfolios.
- SDF structure stable over time.
- Mean-variance efficient portfolio implied by pricing kernel highly profitable in a risk-adjusted sense.