Autoencoder asset pricing models

Shihao Gu , Bryan Kelly , Dacheng Xiu

Journal of Econometrics,2020

解读者：屠雪永

2021.03.04

PS:Gu, Shihao, Bryan Kelly, and Dacheng Xiu. “Empirical Asset Pricing via Machine Learning.” *The Review of Financial Studies* 33, no. 5 (2020): 51.

Outline

- Introduction
- Research design
  - The Standard autoencoder
  - The conditional autoencoder
- Empirical study
- Monte Carlo simulations
- Conclusion

1. Introduction-- Background

- Kelly, Pruitt, and Su (KPS, 2019) provide empirical evidence that characteristics appear to predict returns because they help pinpoint compensated aggregate risk exposures.
- "instrumented" PCA (IPCA) assumes that the map from P characteristics to K betas is linear

$$r_t = \beta f_t + u_t, \longrightarrow r_{i,t} = \beta(z_{i,t-1})' f_t + u_{i,t}$$

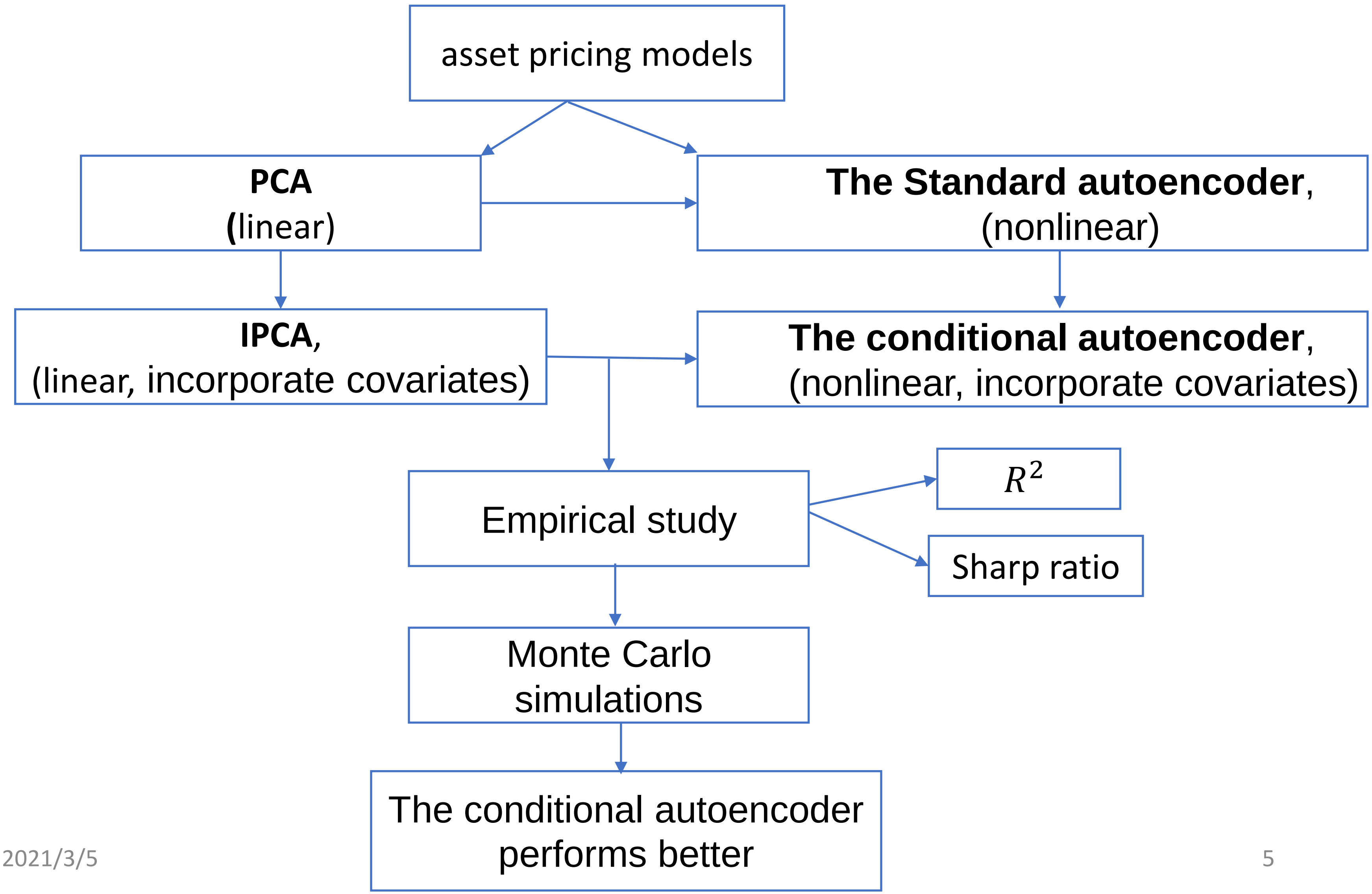
$$\beta(z_{i,t-1})' = z'_{i,t-1} \Gamma$$

The factors f_t are treated as latent,
the $\beta(Z_{\mathrm{i,t-1}})'$ is $K \times 1$ conditional factor exposure,
$Z_{\mathrm{i,t-1}}$ is an $P \times 1$ vector of asset characteristics , where $P > K$

1. Introduction-- Motivation

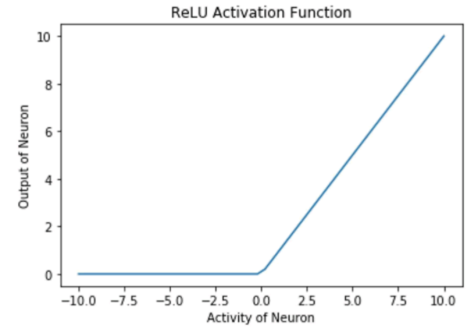
- Like Kelly, Pruitt, and Su (KPS, 2019), our model allows for latent factors and factor exposures that depend on covariates such as asset characteristics.
- Unlike the **linearity** assumption of KPS, we model factor exposures as a flexible **nonlinear** function of covariates----the conditional autoencoder(AC).
- Our machine learning framework imposes the economic restriction of no-arbitrage.

1. Introduction-- Framework



1. Introduction-- Contribution

- 1. Using machine learning techniques to analyze the cross section of risk and return in financial markets.
- 2. We introduce a new conditional autoencoder model for individual stock returns which, like IPCA, allows covariates to help guide dimension reduction.
- 3. Our model delivers out-of-sample pricing errors that are far smaller (and generally insignificant) compared to other leading factor models.



2. Methodology

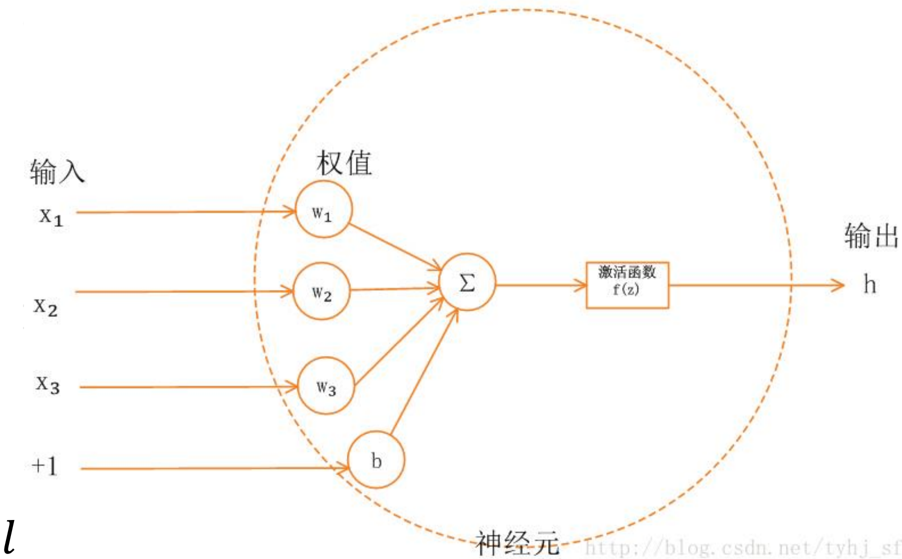
- 2.1. Standard autoencoder
- An autoencoder is a special neural network in which the outputs attempt to approximate the input variables --an unsupervised learning

Hidden layers

$$r^{(l)} = g\left(b^{(l-1)} + W^{(l-1)} r^{(l-1)}\right)$$

Output layers

$$G(r, b, W) = b^{(L)} + W^{(L)} r^{(L)}$$



$r_k^{(l)}$: define the output of neuron k in layer l

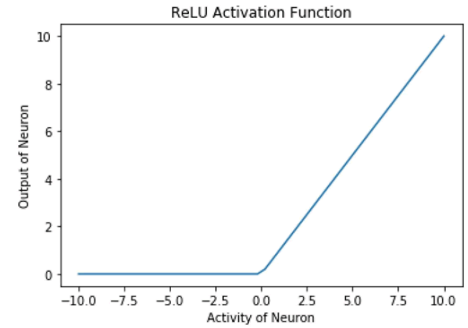
$r(l) = (r_1^{(l)}, \ldots, r_{k(l)}^{(l)})'$: the vector of all outputs for this layer

$g(\cdot)$: a nonlinear activation function

$g(y) = \max(y, 0)$:Relu

$W^{(l-1)}$ is a $K^{(l)} \times K^{(l-1)}$ matrix of weight parameters,

$b^{(l-1)}$ is a $K^{(l)} \times 1$ vector of so-called bias parameters.



2. Methodology

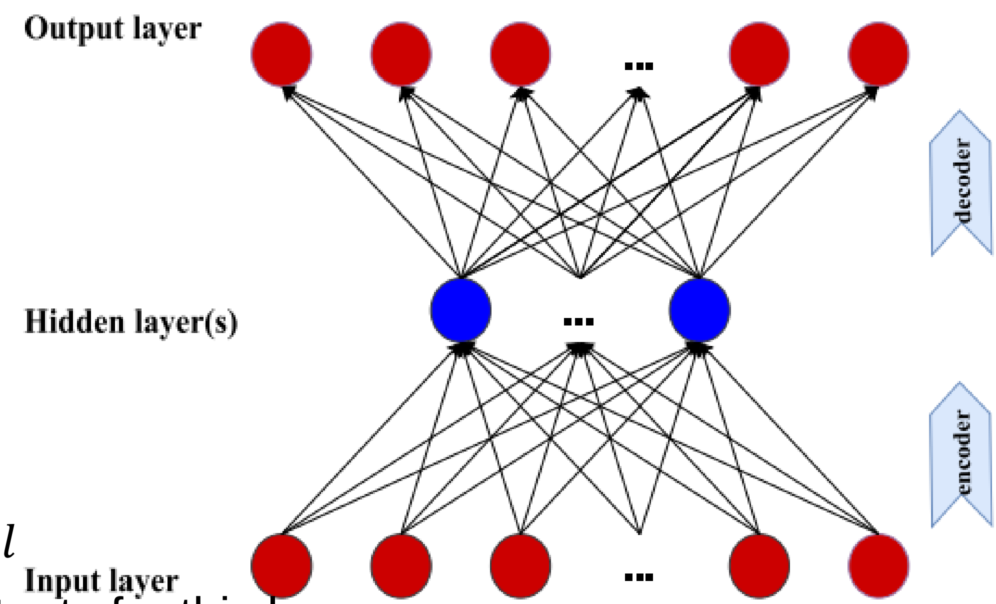
- 2.1. Standard autoencoder
- An autoencoder is a special neural network in which the outputs attempt to approximate the input variables --an unsupervised learning

Hidden layers

$$r^{(l)} = g\left(b^{(l-1)} + W^{(l-1)} r^{(l-1)}\right)$$

Output layers

$$G(r, b, W) = b^{(L)} + W^{(L)} r^{(L)}$$



$r_k^{(l)}$: define the output of neuron k in layer l

$r(l) = (r_1^{(l)}, \ldots, r_{k(l)}^{(l)})'$: the vector of all outputs for this layer

$g(\cdot)$: a nonlinear activation function

$g(y) = \max(y, 0)$:Relu

$W^{(l-1)}$ is a $K^{(l)} \times K^{(l-1)}$ matrix of weight parameters,

$b^{(l-1)}$ is a $K^{(l)} \times 1$ vector of so-called bias parameters.

2. Methodology

- 2.1.1. Static linear factor models as a special case
- the most commonly studied models of asset returns assume a **linear** latent factor specification with **static** loadings

$$r_t = \beta f_t + u_t, \xrightarrow{\text{matrix}} R = \beta F + U.$$

- Estimated with PCA $\longrightarrow$ using SVD

$$\bar{R} = \widehat{P} \Lambda \widehat{Q} + \widehat{U},$$

- The one-layer, linear autoencoder with K neurons

$$r_t = b^{(1)} + W^{(1)}(b^{(0)} + W^{(0)} r_t) + u_t,$$

$$\min_{b,W} \left\| R - \left(b^{(1)} \iota' + W^{(1)}(b^{(0)} \iota' + W^{(0)} R)\right) \right\|_F^2$$

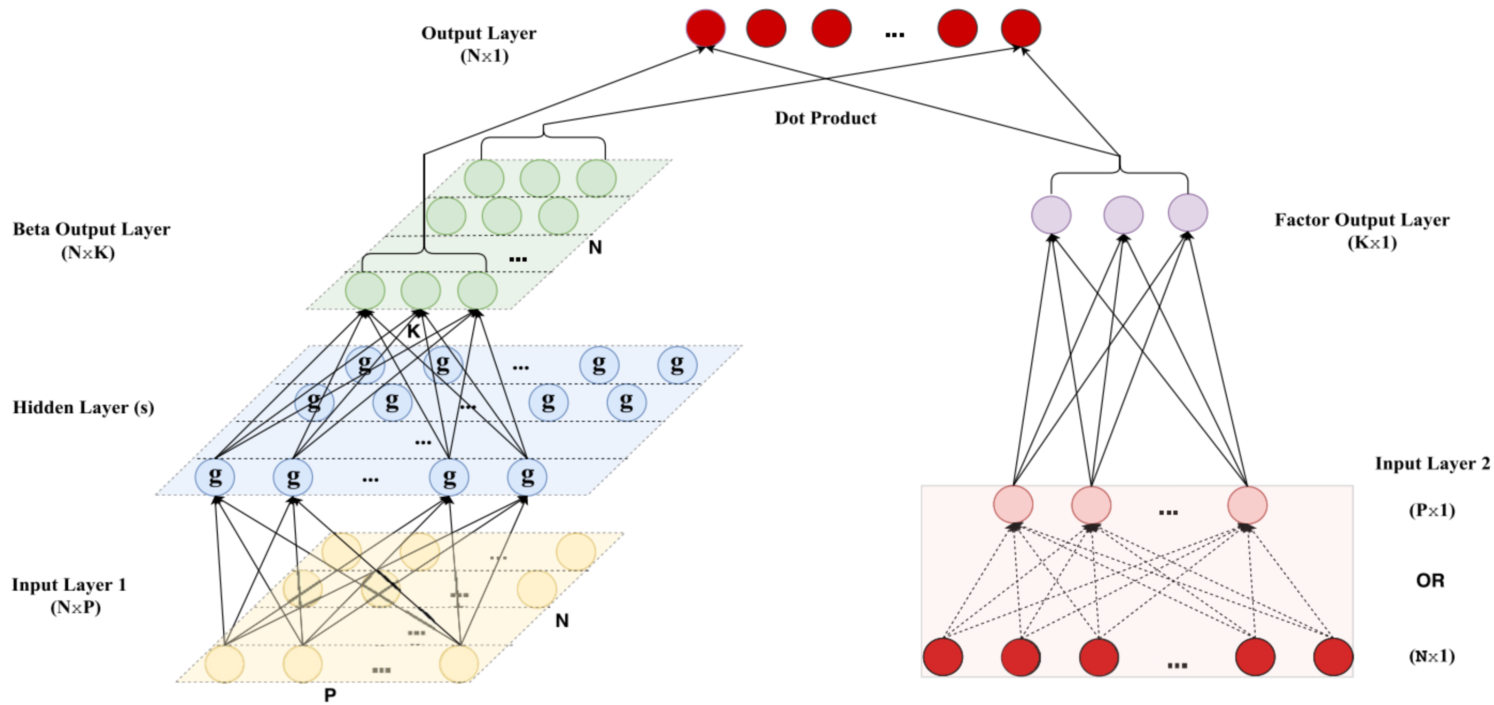
- The linear autoencoder is equivalent to PCA since they have the **same factor loading matrix $\widehat{p}$**

$$\widehat{W}^{(1)} = \widehat{P} A, \quad \widehat{W}^{(0)} = (\widehat{W}^{(1)\prime} \widehat{W}^{(1)})^{-1} \widehat{W}^{(1)\prime}, \quad \widehat{b}^{(1)} = \bar{r} - \widehat{W}^{(1)} \widehat{b}^{(0)} - \widehat{W}^{(1)} \widehat{W}^{(0)} \bar{r}, \quad \widehat{b}^{(0)} = a,$$

2. Methodology

- 2.2. Extending the autoencoder model to include covariates

$$r_{i,t} = \beta'_{i,t-1} f_t + u_{i,t}.$$



(beta) (factor)

$$x_t = (Z'_{t-1} Z_{t-1})^{-1} Z_{t-1} r_t.$$

$$z_{i,t-1}^{(0)} = z_{i,t-1},$$

$$z_{i,t-1}^{(l)} = g\left(b^{(l-1)} + W^{(l-1)} z_{i,t-1}^{(l-1)}\right), \quad l = 1, \ldots, L_\beta,$$

$$\beta_{i,t-1} = b^{(L_\beta)} + W^{(L_\beta)} z_{i,t-1}^{(L_\beta)}.$$

$$r_t^{(0)} = r_t,$$

$$r_t^{(l)} = \tilde{g}\left(\tilde{b}^{(l-1)} + \tilde{W}^{(l-1)} r_t^{(l-1)}\right), \quad l = 1, \ldots, L_f,$$

$$f_t = \tilde{b}^{(L_f)} + \tilde{W}^{(L_f)} r_t^{(L_f)}.$$

2. Methodology

- 2.2.1. Conditional linear factor models as a special case
- IPCA solves the optimization problem:

$$\min_{\Gamma,F} \sum_{t=1}^{T} \sum_{i=1}^{N} \left\| r_{i,t} - z'_{i,t-1}\Gamma' f_t \right\|^2 = \min_{\Gamma,F} \sum_{t=1}^{T} \left\| r_t - Z_{t-1}\Gamma' f_t \right\|^2 . \qquad (17)$$

- the estimation objective of the conditional autoencoder is:

$$\beta'_{i,t} = Z_{t-1} W'_0 \qquad f_t = W_1 x_t$$

$$\min_{W_0, W_1} \sum_{t=1}^{T} \left\| r_t - Z_{t-1} W'_0 W_1 x_t \right\|^2 . \qquad (18)$$

- The solution to (18) is equivalent to the solution of (17) if $Z'_t Z_t = \Sigma$ for a constant matrix Σ
- In the general case where $Z'_t Z_t$ is non-constant, the two estimators are similar but no longer equivalent

2. Methodology

- 2.3. Regularized autoencoder learning
- 2.3.1. Training, validation, and testing
- 2.3.2. Regularization techniques

$$\mathcal{L}(\theta;\cdot)=\frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N}\left\|r_{i,t}-\beta'_{i,t-1}f_t\right\|^2+\phi(\theta;\cdot),$$

$$\phi(\theta;\lambda)=\lambda\sum_{j}|\theta_j|.$$

θ summarizes the weight parameters in the loading and factor networks

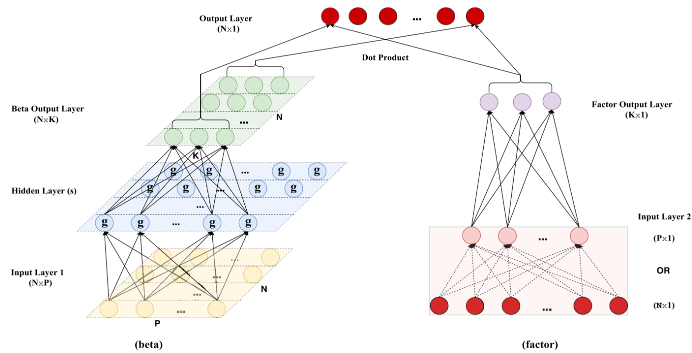
  - Early stopping:optimization is terminated when the validation sample errors begin to increase
  - use multiple random seeds, and predicted by averaging estimates

- 2.3.3. Optimization algorithms
  - stochastic gradient descent (SGD)---Adam

3. An empirical study of US equity

- 3.1. Data
- The same dataset studied in Gu et al. (2019)
- Sample period:1957.3-2016.12
- Sample data:94 characteristics, monthly individual stock returns from the CRSP, for all firms listed in NYSE, AMEX, and NASDAQ.
- The risk-free rate :the Treasury bill rate
- Training sample:18 years (1957–1974) ;validation sample:12 years of (1975–1986);testing sample:30 years (1987–2016)

3. An empirical study of US equity



- 3.2. Models comparison set
- Conditional autoencoder (CA) :
- CA_0, uses a single linear layer in both the beta and factor networks.
- CA_1, CA_2 and CA_3 add a first, second and third hidden layer in the beta network.
- CA_0 through CA_3 all maintain a one-layer linear specification on the factor side of the model, the number of neurons range from 1 to 6
- FF :1 to 6 factors. the excess market return, SMB, HML, and UMD, sequentially. The five-factor model is the market, SMB, HML, CMA, and RMW, and the six-factor model again appends UMD.

3. An empirical study of US equity

- 3.3. Statistical performance evaluation

$$R^2_{\text{total}} = 1 - \frac{\sum_{(i,t)\in OOS}(r_{i,t} - \widehat{\beta}'_{i,t-1}\widehat{f}_t)^2}{\sum_{(i,t)\in OOS} r_{i,t}^2}$$

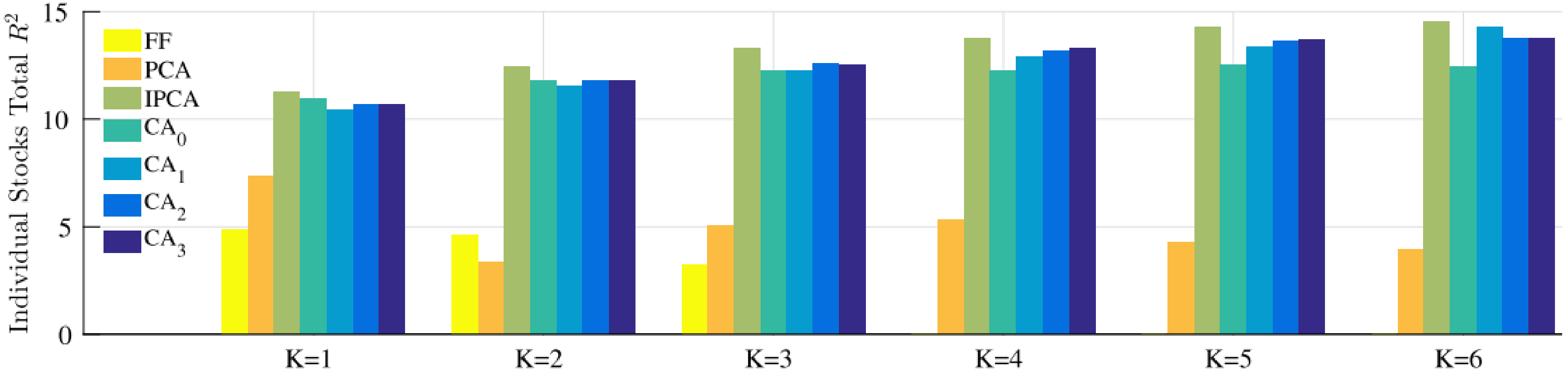
- quantifies the explanatory power of contemporaneous factor realizations, and assesses the model's description of individual stock riskiness

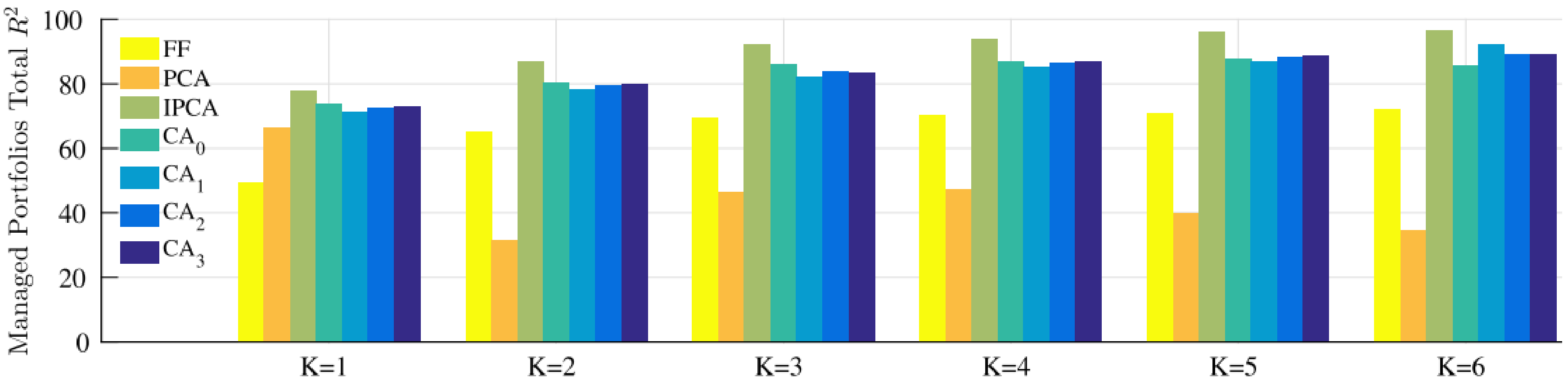
$$R^2_{\text{pred}} = 1 - \frac{\sum_{(i,t)\in OOS}(r_{i,t} - \widehat{\beta}'_{i,t-1}\widehat{\lambda}_{t-1})^2}{\sum_{(i,t)\in OOS} r_{i,t}^2},$$

- where $\hat{\lambda}_{t-1}$ is the prevailing sample average of $\hat{f}$ up to month $t-1$.
- quantifies the predictions of future individual excess stock returns. assesses a model's ability to explain panel variation in risk compensation.

3. An empirical study of US equity

- 3.3. Statistical performance evaluation-- total R^2

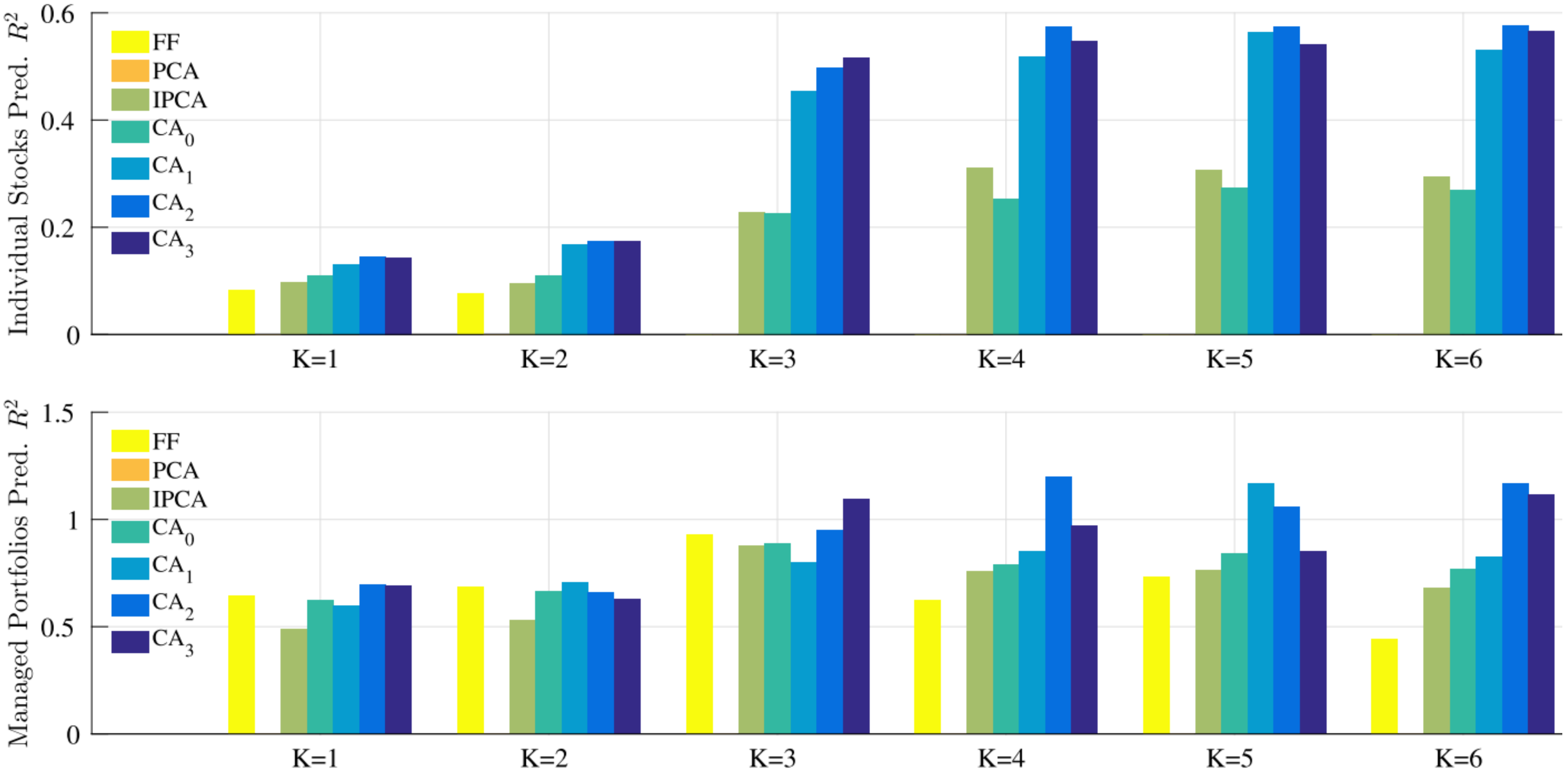




➢ IPCA provides the best fit , total R^2 at the portfolio-level tends to be far higher

3. An empirical study of US equity

- 3.3. Statistical performance evaluation– pred. R^2



➢ CA_1, CA_2, and CA_3 dramatically outperform the FF, PCA and IPCA models

3. An empirical study of US equity

- 3.4. Economic performance evaluation

Table 3
Out-of-sample sharpe ratios of long–short portfolios.

Equal-weight	*K*					
	1	2	3	4	5	6
FF	−0.66	−0.85	−0.40	−0.30	0.36	−0.21
PCA	0.28	0.09	0.13	−0.08	−0.12	0.15
IPCA	0.20	0.19	1.26	2.16	2.31	2.25
CA_0	0.23	0.32	1.34	1.87	2.10	2.18
CA_1	0.30	0.39	2.12	2.63	2.67	2.60
CA_2	0.30	0.38	2.16	2.64	2.68	2.63
CA_3	0.31	0.38	2.19	2.57	2.57	2.59
Value-weight	*K*					
	1	2	3	4	5	6
FF	−0.82	−1.13	−0.69	−0.60	0.18	−0.53
PCA	0.12	−0.18	0.05	−0.10	−0.30	−0.08
IPCA	−0.15	−0.07	0.59	0.81	1.05	0.96
CA_0	−0.11	−0.03	0.41	0.81	0.83	0.88
CA_1	−0.03	0.11	0.91	1.30	1.48	1.40
CA_2	−0.03	0.08	0.92	1.39	1.45	1.53
CA_3	−0.02	0.08	1.09	1.41	1.34	1.51

- CA_2 outperforms others. Following the nonlinear conditional autoencoders, the best model is IPCA

3. An empirical study of US equity

- 3.4. Economic performance evaluation

- To evaluate the multi-factor mean–variance efficiency

Table 4
Out-of-sample factor tangency portfolio sharpe ratios.

	K					
	1	2	3	4	5	6
FF	0.51	0.41	0.53	0.71	0.71	0.82
PCA	0.35	0.23	0.25	0.38	0.48	0.55
IPCA	0.39	0.44	1.81	3.14	3.71	3.72
CA_0	0.42	0.48	1.47	1.76	1.94	1.97
CA_1	0.56	0.91	3.18	3.82	3.63	4.58
CA_2	0.54	0.75	3.56	4.26	4.72	2.77
CA_3	0.54	0.77	3.94	4.75	4.94	4.37

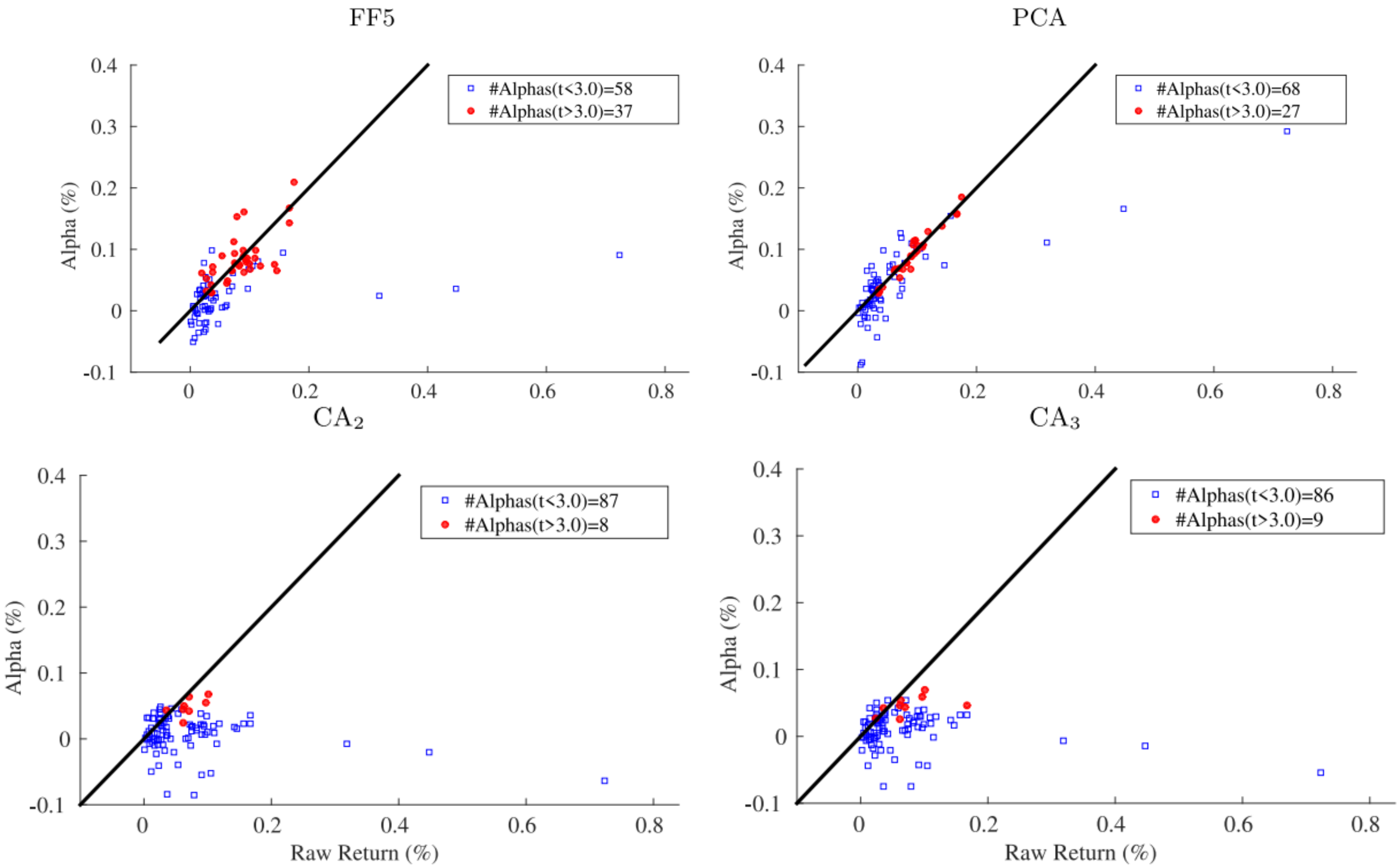
➢ The most dominant overall model on this dimension is CA_3 with five factors

3. An empirical study of US equity

- 3.5. Risk premia vs. mispricing

Out-of-sample pricing errors (alphas) :

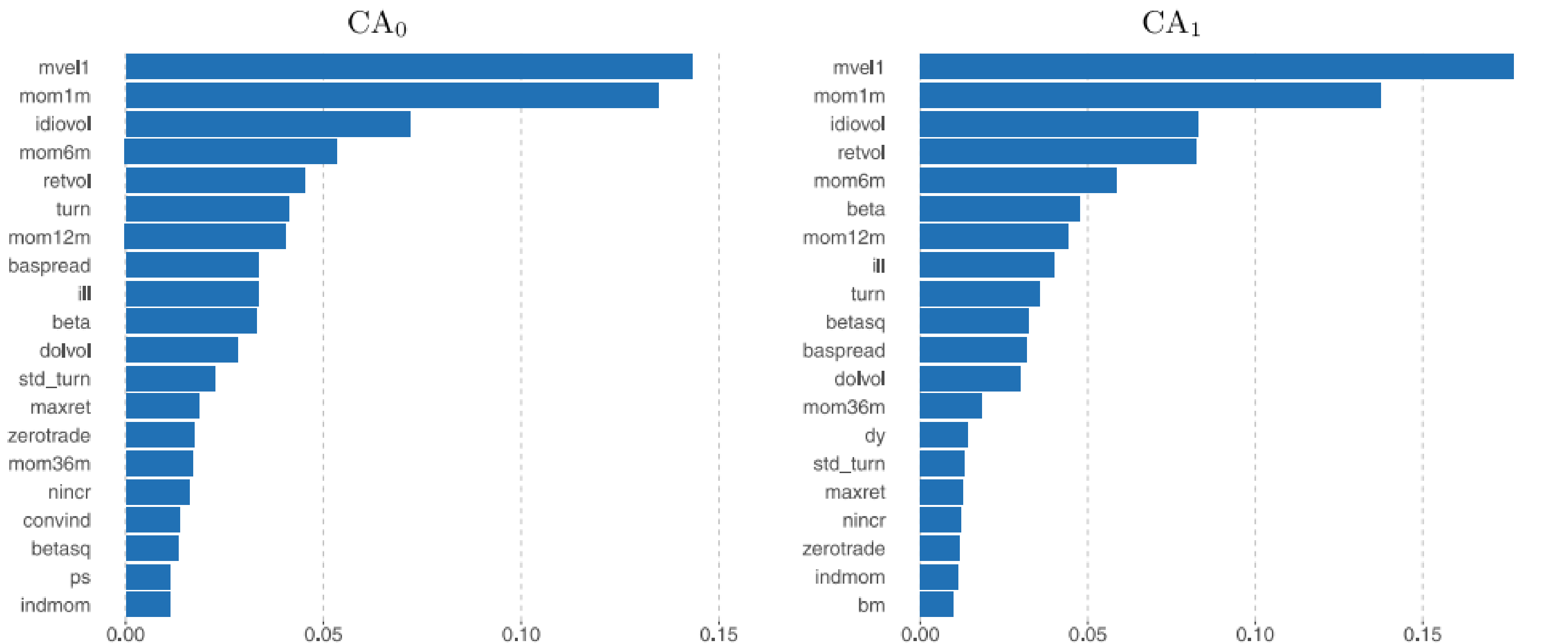
$$\alpha_i := \mathrm{E}(u_{i,t}) = \mathrm{E}(r_{i,t}) - \mathrm{E}(\beta'_{i,t-1} f_t).$$



➢ For FF5, 37 of the 95 managed portfolios have alpha t-statistics in excess of 3.0. For CA2, that number drops to 8 out of 95.

3. An empirical study of US equity

- 3.6. Characteristics importance-- the reduction in total R^2



- ➢ The first is a price trend category, which includes short-term reversal (mom1m), stock momentum (mom12m) .
- ➢ The second category includes liquidity variables, such as turnover and turnover volatility (turn, std_turn), dollar volume (dolvol).
- ➢ Risk measures constitute the third influential group, including total and idiosyncratic return volatility (retvol, idiovol), market beta (beta), and betasquared (betasq).

4. Monte Carlo simulations

- We simulate a conditional factor model for excess returns r_t, for $t = 1, 2, \ldots, T$:

$$r_{i,t} = \beta_{i,t-1} f_t + \varepsilon_{i,t}, \quad \beta_{i,t-1} = g^*(c_{i,t-1}; \theta), \quad f_t = W x_t + \eta_t$$

$$x_t \sim \mathcal{N}(0.03, 0.1^2 \times \mathbb{I}_{P_x}), \; \eta_t \sim \mathcal{N}(0, 0.01^2 \times \mathbb{I}_3) \text{ and } \varepsilon_{i,t} \sim t_5(0, 0.1^2),$$

- The panel of characteristics

$$c_{ij,t} = \frac{2}{n+1} \operatorname{rank}(\bar{c}_{ij,t}) - 1, \quad \bar{c}_{ij,t} = \rho_j \bar{c}_{ij,t-1} + \epsilon_{ij,t},$$

$$\rho_j \sim \mathcal{U}[0.9, 1], \text{ and } \epsilon_{ij,t} \sim \mathcal{N}(0, 1)$$

- Weight matrix

$$W = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \end{bmatrix}$$

- $g^*(\cdot)$ functions:
  - Linear $\quad g^*(c_{i,t}; \theta) = (1.2 \times c_{i1,t}, c_{i2,t}, 0.8 \times c_{i3,t})'.$
  - nonlinear $\quad g^*(c_{i,t}; \theta) = \left(c_{i1,t}^2, 2 \times (c_{i1,t} \times c_{i2,t}), 0.6 \times \operatorname{sgn}(c_{i3,t})\right)'.$

4. Monte Carlo simulations

Table 6
Comparison of total R^2(%)s and predictive R^2(%)s in simulations.

Model (a)	K					
Total. R^2	1	2	3	4	5	6
PCA	3.5	4.7	5.5	6.3	7.1	7.8
IPCA	18.6	32.2	40.7	41.0	41.4	$\boxed{41.7}$
CA_0	15.6	26.7	33.7	33.5	33.4	33.2
CA_1	17.6	30.3	38.1	37.7	37.3	37.1
CA_2	17.7	29.2	36.8	36.5	36.3	35.9
CA_3	17.6	25.6	30.0	29.5	26.3	23.4
Pred. R^2						
PCA	0.17	0.10	0.04	0.01	−0.01	−0.03
IPCA	2.20	2.93	3.33	3.32	3.32	$\boxed{3.32}$
CA_0	2.04	2.84	3.17	3.14	3.12	3.13
CA_1	2.11	2.93	3.27	3.29	3.26	3.26
CA_2	2.10	2.85	3.22	3.22	3.23	3.22
CA_3	2.06	2.57	2.89	2.86	2.58	2.39

- IPCA delivers the best OOS total and predictive R^2s. Because the true model is sparse and linear in the input covariates.

4. Monte Carlo simulations

Model (b)	K					
Total. R^2	1	2	3	4	5	6
PCA	3.4	5.1	6.0	6.6	7.3	7.9
IPCA	11.0	11.4	11.9	12.3	12.7	13.1
CA_0	8.5	8.2	7.9	7.6	7.4	7.2
CA_1	15.0	24.6	31.8	32.0	31.9	31.8
CA_2	15.7	23.5	30.9	31.8	30.2	28.2
CA_3	15.9	15.6	14.6	14.0	11.2	9.2
Pred. R^2						
PCA	0.15	0.19	0.15	0.12	0.10	0.09
IPCA	0.84	0.82	0.81	0.80	0.79	0.79
CA_0	0.80	0.76	0.77	0.76	0.72	0.70
CA_1	1.83	2.31	2.70	2.70	2.71	2.73
CA_2	1.95	2.24	2.73	2.80	2.69	2.53
CA_3	1.77	1.43	1.32	1.26	1.06	0.86

- CA models clearly beat IPCA, because the latter cannot capture the nonlinearity in the model.

4. Conclusion

- Firstly, our conditional autoencoder model dominates competing asset pricing models, including Fama–French models, PCA methods, and linear conditioning methods such as IPCA.
- Secondly, the pricing errors in our model (likewise measured on an out-of-sample basis) are a fraction of the magnitude of those from traditional Fama–French factor models.

5. Inspiration

- This paper enriches the asset pricing model, by combining machine learning and asset pricing model.
- This paper **simultaneously considers features and labels to predict labels,** which can be used to solve other similar problems, or designs other models to try to consider features and labels at the same time to predict for time series data.
- Change return into fundamentals for fundamental extrapolation