Complexity in Factor Pricing Models

Antoine Didisheim — *Uni. Melbourne*
Shikun (Barry) Ke — *Yale*
Bryan Kelly — *Yale*
Semyon Malamud — *EPFL*

AFA 2024

The "Virtue of Complexity" in Asset Pricing

Building the "Case" for Financial ML

- Finance lit: Rapid advances in return prediction/portfolio choice using ML
- Large empirical gains over simple models
- Little theoretical understanding of why (and healthy skepticism)

"Virtue of Complexity in Return Prediction" (Kelly, Malamud, Zhou, forthcoming JF)

- Main theoretical result: Out-of-sample univariate timing strategy performance generally *increasing* in model complexity (# of parameters). Bigger models are better. Verified in data.

The "Virtue of Complexity" in Asset Pricing

Building the "Case" for Financial ML

- Finance lit: Rapid advances in return prediction/portfolio choice using ML
- Large empirical gains over simple models
- Little theoretical understanding of why (and healthy skepticism)

"Virtue of Complexity in Return Prediction" (Kelly, Malamud, Zhou, forthcoming JF)

- Main theoretical result: Out-of-sample univariate timing strategy performance generally *increasing* in model complexity (# of parameters). Bigger models are better. Verified in data.

This Paper: ML in Cross-sectional Asset Pricing

- Main theoretical result: SDF performance generally *increasing* in model complexity
  - Higher portfolio Sharpe ratio
  - Smaller pricing errors
- Prior evidence of empirical gains from ML are *what we should expect*
- **Direct empirical support for theory**

Complexity in the Cross Section: A Brief History

SDF representable as managed portfolios: $M^{\star}_{t+1} = 1 - \sum_{i=1}^{n} w(X_t)' R_{i,t+1}$, s.t. $E_t[M^{\star}_{t+1} R_{i,t+1}] = 0 \ \forall i$

- Cross-sectional asset pricing is about $w_t = w(X_t)$
- Fundamental challenge in cross-sectional asset pricing: w must be estimated
  - This is a high-dimensional (*complex*) problem
  - We know: In-sample tangency portfolio behaves horribly out-of-sample

Complexity in the Cross Section: A Brief History

SDF representable as managed portfolios: $M^\star_{t+1} = 1 - \sum_{i=1}^{n} w(X_t)' R_{i,t+1}$, s.t. $E_t[M^\star_{t+1} R_{i,t+1}] = 0 \ \forall i$

- Cross-sectional asset pricing is about $w_t = w(X_t)$
- Fundamental challenge in cross-sectional asset pricing: w must be estimated
  - This is a high-dimensional (*complex*) problem
  - We know: In-sample tangency portfolio behaves horribly out-of-sample
- Standard solution: Restrict w's functional form
  - E.g., Fama-French: $w_{i,t} = b_0 + b_1 \text{Size}_{i,t} + b_2 \text{Value}_{i,t}$ (Brandt et al. 2007 generalize)
  - Reduces parameters, implies factor model: $M_{t+1} = 1 - b_0 MKT - b_1 SMB - b_2 HML$
  - "Shrinking the cross-section" Kozak et al. (2020) — use a few PCs of anomaly factors

Complexity in the Cross Section: Machine Learning Perspective

SDF representable as $M^{\star}_{t+1} = 1 - \sum_{i=1}^{n} w(X_t)' R_{i,t+1}$, s.t. $E_t[M^{\star}_{t+1} R_{i,t+1}] = 0 \ \forall i$

Rather than restricting $w(X_t)$....

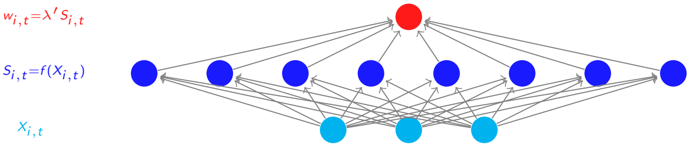
- ...expand parameterization, saturate with conditioning information

Complexity in the Cross Section: Machine Learning Perspective

SDF representable as $M^{\star}_{t+1} = 1 - \sum_{i=1}^{n} w(X_t)' R_{i,t+1}$, s.t. $E_t[M^{\star}_{t+1} R_{i,t+1}] = 0 \ \forall i$

Rather than restricting $w(X_t)$....

- ...expand parameterization, saturate with conditioning information
- Approximate w with neural network: $\hat{w}(X_{i,t}, \lambda) \approx \lambda' S_{i,t}$ with a linear family
- $P \times 1$ vector $S_{i,t}$ is known nonlinear function of original predictors $X_{i,t}$

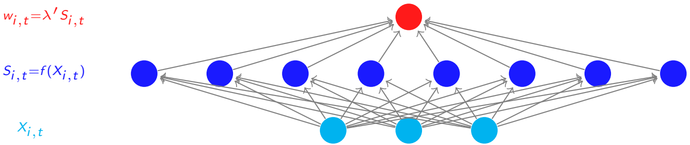


Complexity in the Cross Section: Machine Learning Perspective

SDF representable as $M^{\star}_{t+1} = 1 - \sum_{i=1}^{n} w(X_t)' R_{i,t+1}$, s.t. $E_t[M^{\star}_{t+1} R_{i,t+1}] = 0 \ \forall i$

Rather than restricting $w(X_t)$....

- ...expand parameterization, saturate with conditioning information
- Approximate w with neural network: $\hat{w}(X_{i,t}, \lambda) \approx \lambda' S_{i,t}$ with a linear family
- $P \times 1$ vector $S_{i,t}$ is known nonlinear function of original predictors $X_{i,t}$



- Implies that empirical SDF is a high-dimensional factor model with factors F_{t+1} :

$$
\begin{aligned}
M^{\star}_{t+1} \approx M_{t+1} &= 1 - \lambda' S_t' R_{t+1} \\
&= 1 - \sum_i (\lambda' S_{i,t} R_{i,t+1}) = 1 - \lambda' \underbrace{\sum_i S_{i,t} R_{i,t+1}}_{= F_{t+1} \in \mathbb{R}^{P \times 1}} = 1 - \lambda' F_{t+1}
\end{aligned}
\tag{1}
$$

Complexity in the Cross Section: Machine Learning Perspective

True SDF: $M^{\star}_{t+1} = 1 - w(X_t)' R_{t+1}$ Empirical Model: $M_{t+1} = 1 - \underbrace{\lambda' F_{t+1}}_{P \text{ params}}$

The Objective:

- Maximize out-of-sample Sharpe ratio (equivalently, minimize out-of-sample pricing errors) of SDF

Complexity in the Cross Section: Machine Learning Perspective

True SDF: $M^{\star}_{t+1} = 1 - w(X_t)'R_{t+1}$ Empirical Model: $M_{t+1} = 1 - \underbrace{\lambda' F_{t+1}}_{P \text{ params}}$

The Objective:

- Maximize out-of-sample Sharpe ratio (equivalently, minimize out-of-sample pricing errors) of SDF

The Choice:

- Fix T data points. Decide on "complexity" (number of factors P) to use in approximating model

Complexity in the Cross Section: Machine Learning Perspective

True SDF: $M^{\star}_{t+1} = 1 - w(X_t)' R_{t+1}$ Empirical Model: $M_{t+1} = 1 - \underbrace{\lambda' F_{t+1}}_{P \text{ params}}$

The Objective:

- Maximize out-of-sample Sharpe ratio (equivalently, minimize out-of-sample pricing errors) of SDF

The Choice:

- Fix T data points. Decide on "complexity" (number of factors P) to use in approximating model

The Tradeoff:

- Simple SDF ($P << T$) has low variance (thanks to parsimony) but is a poor approximator of w
- Complex SDF ($P > T$) is good approximator but may behave poorly (and requires shrinkage)

Complexity in the Cross Section: Machine Learning Perspective

True SDF: $M^{\star}_{t+1} = 1 - w(X_t)' R_{t+1}$ Empirical Model: $M_{t+1} = 1 - \underbrace{\lambda' F_{t+1}}_{P \text{ params}}$

The Objective:

- Maximize out-of-sample Sharpe ratio (equivalently, minimize out-of-sample pricing errors) of SDF

The Choice:

- Fix T data points. Decide on "complexity" (number of factors P) to use in approximating model

The Tradeoff:

- Simple SDF ($P << T$) has low variance (thanks to parsimony) but is a poor approximator of w
- Complex SDF ($P > T$) is good approximator but may behave poorly (and requires shrinkage)

The Central Research Question:

- Which P should the researcher opt for? Does the benefit of more factors justify their cost?

Complexity in the Cross Section: Machine Learning Perspective

True SDF: $M^{\star}_{t+1} = 1 - w(X_t)' R_{t+1}$ Empirical Model: $M_{t+1} = 1 - \underbrace{\lambda' F_{t+1}}_{P \text{ params}}$

The Objective:

- Maximize out-of-sample Sharpe ratio (equivalently, minimize out-of-sample pricing errors) of SDF

The Choice:

- Fix T data points. Decide on "complexity" (number of factors P) to use in approximating model

The Tradeoff:

- Simple SDF ($P << T$) has low variance (thanks to parsimony) but is a poor approximator of w
- Complex SDF ($P > T$) is good approximator but may behave poorly (and requires shrinkage)

The Central Research Question:

- Which P should the researcher opt for? Does the benefit of more factors justify their cost?

Answer:

- Use the largest factor model (largest P) that you can compute Illustration

Theory Environment

Model

- n assets with returns R_{t+1}
- Empirical SDF $M_{t+1} = 1 - \lambda' S_t' R_{t+1}$
  - Think of S_t as "generated features" in neural net with input X_t
  - $P \times 1$ vector of instruments, S_t (i.e., P factors F_{t+1})
- (Ridge-penalized) objective

Max Sharpe Ratio		Min Pricing Error (HJ-distance)
$\min_\lambda E[(1 - \lambda' S_t' R_{t+1})^2] + z\lambda'\lambda$	or	$\min_\lambda E[MF]' E[FF']^{-1} E[MF] + z\lambda'\lambda$

Solution:

$$\hat{\lambda}(z) = \left(zI + \frac{1}{T}\sum_t F_t F_t'\right)^{-1} \frac{1}{T}\sum_t F_t$$

Theory Environment

Model

- n assets with returns R_{t+1}
- Empirical SDF $M_{t+1} = 1 - \lambda' S_t' R_{t+1}$
  - Think of S_t as "generated features" in neural net with input X_t
  - $P \times 1$ vector of instruments, S_t (i.e., P factors F_{t+1})
- (Ridge-penalized) objective

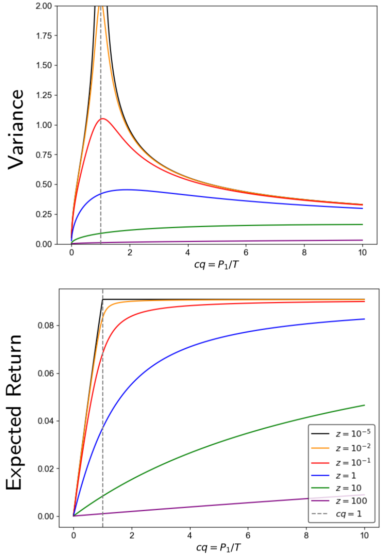
Max Sharpe Ratio		Min Pricing Error (HJ-distance)
$\min_\lambda E[(1-\lambda' S_t' R_{t+1})^2] + z\lambda'\lambda$	or	$\min_\lambda E[MF]' E[FF']^{-1} E[MF] + z\lambda'\lambda$

Solution:

$$\hat{\lambda}(z) = \left(zI + \frac{1}{T}\sum_t F_t F_t'\right)^{-1} \frac{1}{T}\sum_t F_t$$

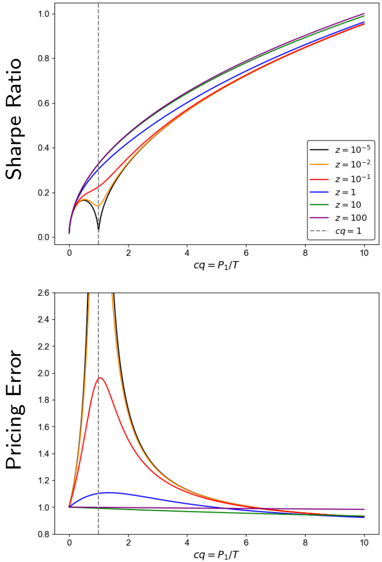
- Goal: Characterize **out-of-sample** behaviors, contrast **simple** (small P) models vs. **complex** models
- Tools: Joint limits as numbers of observations and parameters are large, $T, P \to \infty$, RMT

Complexity and the SDF



1. SDF variance
   - As $c \to 1$, λ variance blows up
   - When $c > 1$, variance *drops* with model complexity! Why?
   - Many λ's exactly fit training data, ridge selects one with a small variance
2. SDF expected returns
   - Low for $c \approx 0$ due to poor approximation of the true model
   - Monotonically increases with model complexity

Complexity and the SDF



Main theory result

- If model is mis-spec, model performance increases with complexity
  - Approximation benefits dominate costs of heavy parameterization
  - Complexity is a virtue
- Other theory results

Empirical Analysis

- Analyze empirical analogs to theoretical comparative statics
- Study conventional setting with conventional data
  - Monthly return of US stocks from CRSP 1963–2021
  - Conditioning info ($X_{i,t}$): 130 stock characteristics from Jensen, Kelly, and Pedersen (2022)
- Out-of-sample performance metrics are:
  - SDF Sharpe ratio
  - Mean squared pricing errors (nonlinear factors as test assets)

Empirical Analysis

Random Fourier Features

- Empirical model: $M_{t+1} = 1 - \lambda' S_t' R_{t+1}$
- Need framework to smoothly transition from low to high complexity

Empirical Analysis

Random Fourier Features

- Empirical model: $M_{t+1} = 1 - \lambda' S_t' R_{t+1}$
- Need framework to smoothly transition from low to high complexity
- Adopt ML method known as "random Fourier features" (RFF)
  - Let $X_{i,t}$ be 130×1 predictors. RFF converts $X_{i,t}$ into

$$S_{\ell,i,t} = [\sin(\gamma_\ell' X_{i,t}), \cos(\gamma_\ell' X_{i,t})], \quad \gamma_\ell \sim iidN(0, \gamma I)$$

  - $S_{\ell,i,t}$: Random lin-combo of $X_{i,t}$ fed through non-linear activation
- For fixed inputs can create an arbitrarily large (or small) feature set
  - Low-dim model (say $P = 1$) draw a single random weight
  - High-dim model (say $P = 10{,}000$) draw many weights

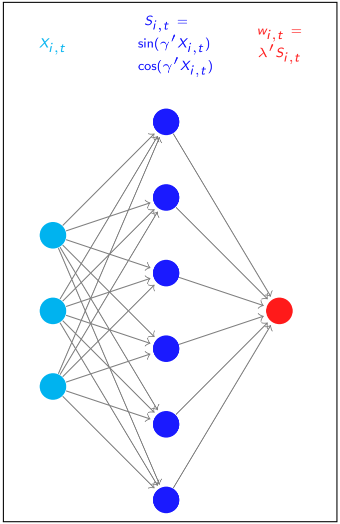
Empirical Analysis

Random Fourier Features

- Empirical model: $M_{t+1} = 1 - \lambda' S_t' R_{t+1}$
- Need framework to smoothly transition from low to high complexity
- Adopt ML method known as "random Fourier features" (RFF)
  - Let $X_{i,t}$ be 130×1 predictors. RFF converts $X_{i,t}$ into

$$S_{\ell,i,t} = [\sin(\gamma_\ell' X_{i,t}), \cos(\gamma_\ell' X_{i,t})], \quad \gamma_\ell \sim iidN(0, \gamma I)$$

  - $S_{\ell,i,t}$: Random lin-combo of $X_{i,t}$ fed through non-linear activation
- For fixed inputs can create an arbitrarily large (or small) feature set
  - Low-dim model (say $P = 1$) draw a single random weight
  - High-dim model (say $P = 10{,}000$) draw many weights
- In fact, RFF is a two-layer neural network with fixed weights (γ) in the first layer and optimized weights (λ) in the second layer

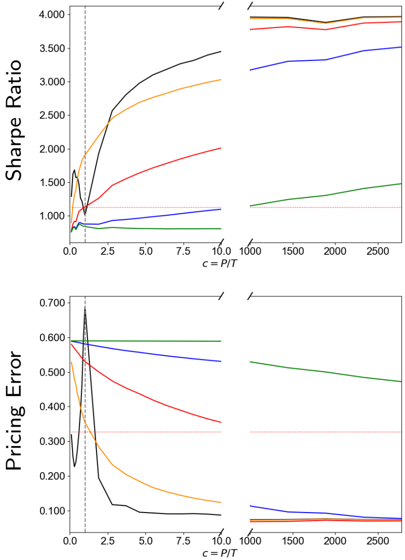


Empirical Analysis

Training and Testing

- We estimate out-of-sample SDF with:
  1. Thirty-year rolling training window ($T = 360$)
  2. Various shrinkage levels, $\log_{10}(z) = -12, ..., 3$
  3. Various complexity levels $P = 10^2, ..., 10^6$
- For each level of complexity $c = P/T$, we plot
  1. Out-of-sample Sharpe ratio of the kernels and
  2. Pricing errors on 10^6 "complex" factors: $F_{t+1} = S_t' R_{t+1}$
- Also report Sharpe ratio and pricing errors of FF6 to benchmark our results

Out-of-sample SDF Performance



Main Empirical Result

- OOS behavior of ML-based SDF closely matches theory
- High complexity models
  - Improve over simple models by a factor of 3 or more
  - Dominate popular benchmarks like FF6
  - Dominate low-rank rotation of complex factors ◂ PCA
- ◂ Mktcap groups

Conclusions

- We provide new, rigorous theoretical insight into the behavior of ML models/portfolios
- Contrary to conventional wisdom: Higher complexity improves model performance

Virtue of Complexity: Performance of ML portfolios can be improved by pushing model parameterization far beyond the number of training observations

Conclusions

- We provide new, rigorous theoretical insight into the behavior of ML models/portfolios
- Contrary to conventional wisdom: Higher complexity improves model performance

Virtue of Complexity: Performance of ML portfolios can be improved by pushing model parameterization far beyond the number of training observations

In canonical empirical problem—pricing the cross section of returns—we find

- OOS Sharpe rise by factor of 4 relative to FF6 model, pricing errors reduced by a factor of 3

Conclusions

- We provide new, rigorous theoretical insight into the behavior of ML models/portfolios
- Contrary to conventional wisdom: Higher complexity improves model performance

Virtue of Complexity: Performance of ML portfolios can be improved by pushing model parameterization far beyond the number of training observations

In canonical empirical problem—pricing the cross section of returns—we find

- OOS Sharpe rise by factor of 4 relative to FF6 model, pricing errors reduced by a factor of 3

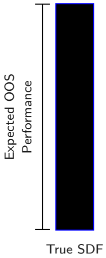
To empirical AP researchers, we recommend

i. including all plausibly relevant predictors
ii. using rich non-linear models rather than simple linear specifications

- Doing so confers prediction/portfolio benefits, even when training data is scarce and particularly when accompanied by shrinkage

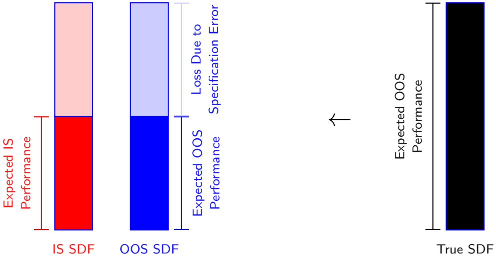
Appendix

Complexity in the Cross Section: Machine Learning Perspective



Complexity in the Cross Section: Machine Learning Perspective

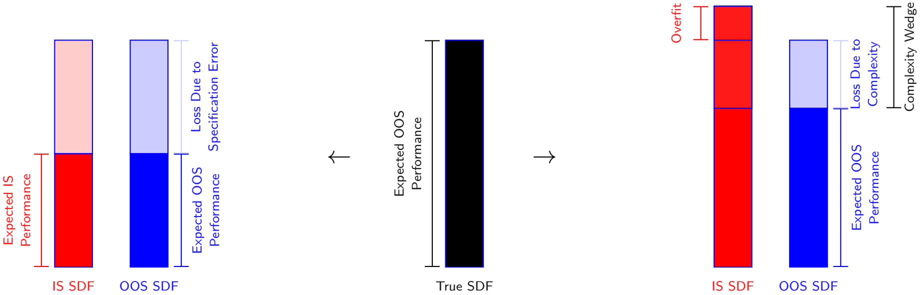
Traditional Approach



- Restrict specification so $P/T \approx 0$
- Aligns IS and OOS performance
- May get lucky with spec, but can't be lucky on average
- Like shrinking *before seeing data*

Complexity in the Cross Section: Machine Learning Perspective

Traditional Approach | Machine Learning Approach



- Restrict specification so $P/T \approx 0$
- Aligns IS and OOS performance
- May get lucky with spec, but can't be lucky on average
- Like shrinking *before seeing data*

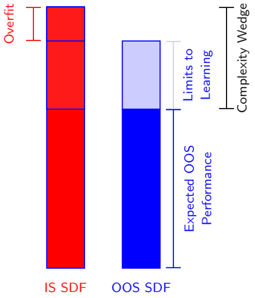
- $P/T \to \infty$ eliminates specification error
- IS overfit *improves* OOS performance
- Loss due to limits on learning (breakdown of LLN, high variance)
- Mitigate with shrinkage *after seeing data*
- Back

Complexity and the SDF: Other Theoretical Results

1. "Complexity wedge" $= \text{IS Performance} - \text{Expected OOS Performance}$

$$= \underbrace{\text{IS} - \text{True}}_{\text{"Overfit"}} + \underbrace{\text{True} - \text{OOS}}_{\text{"Limits to Learning"}}$$

- Quantifiable based on training data
- Can infer performance of true SDF and how far you are from it, but cannot recover it!

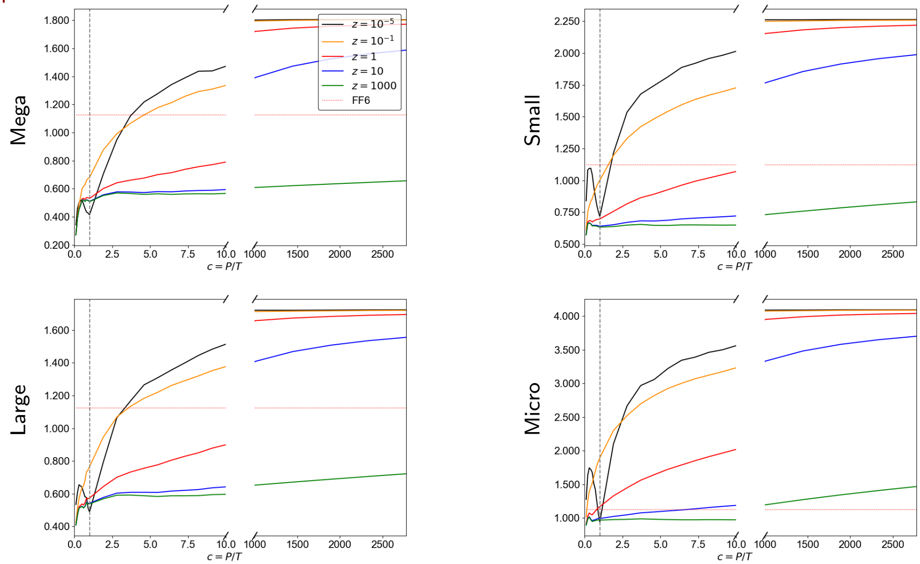


2. Show how to infer optimal shrinkage, z^*, from training data
3. There is no low-rank rotation of complex factors that preserves model performance (cf. Kozak, Nagel, and Santosh, 2020)

- Back

SDF Performance in Restricted Samples: Sharpe Ratio

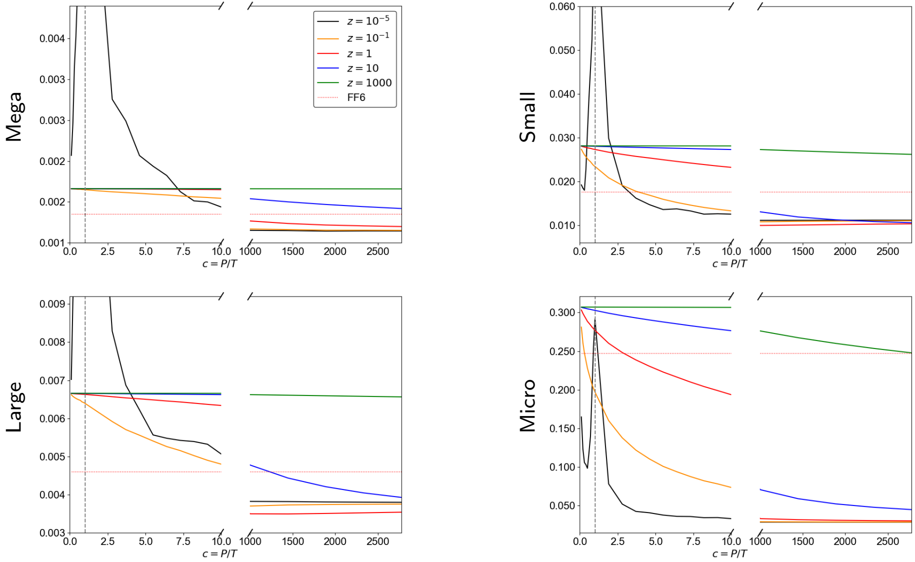
Market Capitalization Subsamples



SDF Performance in Restricted Samples: Pricing Errors

Market Capitalization Subsamples

Back



What About “Shrinking” With PCA?

Back