MS&E349 Final Project Report

Bocheng Dai, Yifan Geng, James Liu, Tiankai Yan

Spring 2025

1 Introduction

In this project, we aim to implement and validate the methodology proposed in the paper *Robust Stock Index Return Predictions Using Deep Learning* by Jagannathan et al. (2023) [1]. The primary objective of this paper is to develop a conditional machine learning framework that leverages deep neural networks to forecast market index returns robustly.

Unlike traditional methods that rely on stable time-series patterns, this model addresses forecast instability by allowing for dynamic, time-varying relationships. It does so by leveraging rich cross-sectional data on asset returns and observable firm characteristics. This approach uses a period-by-period machine learning framework to estimate firm-level expected returns, effectively filtering out idiosyncratic shocks while preserving the underlying factor structure in realized returns. The estimated returns are then used in place of realized returns to generate more reliable estimates of factors and loadings.

A key innovation of the paper lies in the construction of aggregate factors. Instead of weighting firms by market capitalization, the model constructs the factors as an average weighted by stock betas. These betas are less noisy and produce more stable forecasts over short horizons.

The paper introduces innovative ideas by taking advantage of the rich information in cross-sectional data and accommodating time-varying relationships in the model. It also presents rigorous mathematical derivations and solid theoretical justification. However, the complexity of its assumptions and the multi-step modeling framework—with numerous design choices—make the results difficult to replicate. Additionally, the model demonstrates some sensitivity to assumptions validity and hyperparameter tuning, which may raise concerns about the robustness of its performance.

2 Model

We begin by outlining the model and the key assumptions underlying the proposed methodology.

The market index return, denoted by y_{t+1}, is modeled as a weighted average of individual stock returns:

$$y_{t+1} = \sum_{i=1}^{N} w_{i,t}\, x_{i,t+1}. \tag{1}$$

At the firm level, we assume that the stock realized returns $x_{i,t}$ and the book-to-market ratios $v_{i,t}$ are driven by conditional factor models:

$$x_{i,t+1} = \beta_{i,t}^{\top} f_{t+1} + u_{i,t+1} \tag{2}$$

$$v_{i,t} = \lambda_{i,t-1}^{\top} g_t + \eta_{i,t} \tag{3}$$

where f_{t+1} and g_t denote the latent factors driving stock returns and book-to-market ratios, respectively.

Additionally, the proposed framework is built upon the following key setup and assumptions:

- Setup:
  - Observable variables include the market return y_t, firm-level returns $x_{i,t}$, and book-to-market ratios $v_{i,t}$ for $t = 0, \ldots, T$.
  - The dimensions of the stock factors and BM factors, $f_{t+1} \in \mathbb{R}^{K_f}$ and $g_t \in \mathbb{R}^{K_g}$, are such that $K_f \geq K_g$.
  - The factor loadings $\beta_{i,t}$ and $\lambda_{i,t}$ are time-varying. This time variation is essential for generating forecasts that adapt to changing market conditions and are thus robust to market instabilities.
  - The idiosyncratic shocks $u_{i,t}$ and $\eta_{i,t}$ can be correlated.
- Assumptions:
  - There is temporal persistence between the risk factors; specifically, the stock factors are driven by lagged BM factors:

    $$f_{t+1} = \Phi_0 + \Phi_g\, g_t + e_{t+1} \tag{4}$$

  - Factor loadings are modeled as functions of observable firm characteristics $z_{i,t}$ (i.e. a characteristic based factor model):

    $$\beta_{i,t-1} = h_{\beta,t}(z_{i,t-1}) \tag{5}$$

- The factor loadings $\beta_{i,t-1}$ from stock returns serve as a valid insstrument for the book-to-market loadings $\lambda_{i,t-1}$ (i.e. instrumental variables are used to estimate the BM factors).

We now derive the implications of this setup for the index return. Substituting Equation (5) into Equation (1), we have:

$$\begin{aligned} y_{t+1} &= \sum_{i=1}^{N} w_{i,t}\, x_{i,t+1} \\ &= \sum_{i=1}^{N} w_{i,t}\left(\beta_{i,t}^{\top} f_{t+1} + u_{i,t+1}\right) \\ &= \sum_{i=1}^{N} w_{i,t}\, \beta_{i,t}^{\top} f_{t+1} + \sum_{i=1}^{N} w_{i,t}\, u_{i,t+1} \end{aligned}$$

Let $\tilde{\rho}_{f,t} = \sum_{i=1}^{N} w_{i,t}\, \beta_{i,t}$ and $\tilde{\epsilon}_{t+1} = \sum_{i=1}^{N} w_{i,t}\, u_{i,t+1}$. Then:

$$y_{t+1} = \tilde{\rho}_{f,t}^{\top}\, f_{t+1} + \tilde{\epsilon}_{t+1}. \tag{6}$$

Substituting Equation (4) into Equation (6) gives:

$$\begin{aligned} y_{t+1} &= \tilde{\rho}_{f,t}^{\top}\, f_{t+1} + \tilde{\epsilon}_{t+1} \\ &= \tilde{\rho}_{f,t}^{\top}\left(\Phi_0 + \Phi_g\, g_t + e_{t+1}\right) + \tilde{\epsilon}_{t+1} \\ &= \tilde{\rho}_{f,t}^{\top}\, \Phi_0 + \tilde{\rho}_{f,t}^{\top}\, \Phi_g\, g_t + \tilde{\rho}_{f,t}^{\top}\, e_{t+1} + \tilde{\epsilon}_{t+1} \end{aligned}$$

Let $\rho_{0,t} = \tilde{\rho}_{f,t}^{\top}\, \Phi_0$, $\rho_{g,t}^{\top} = \tilde{\rho}_{f,t}^{\top}\, \Phi_g$, and $\epsilon_{t+1} = \tilde{\rho}_{f,t}^{\top}\, e_{t+1} + \tilde{\epsilon}_{t+1}$. Then:

$$y_{t+1} = \rho_{0,t} + \rho_{g,t}^{\top}\, g_t + \epsilon_{t+1}. \tag{7}$$

We further assume that $\rho_{0,t}$ and $\rho_{g,t}$ are either constant or slowly moving over time.

In summary, the final model is

$$y_{t+1} = \rho_{0,t} + \rho_{g,t}^{\top}\, g_t + \epsilon_{t+1} \tag{8}$$

$$x_{i,t} = \beta_{i,t-1}^{\top} f_t + u_{i,t} \tag{9}$$

$$v_{i,t} = \lambda_{i,t-1}^{\top} g_t + \eta_{i,t} \tag{10}$$

3 Algorithm

Given the model structure and assumptions introduced in the previous section, we now discuss the algorithm in detail and explain the intuition behind its design.

3.1 The Full Algorithm

1. Construct $\hat{m}_t(\cdot)$ using cross-sectional neural network regression of stock returns $x_{i,t}$ onto firm characteristics $z_{i,t}$ at each period $t = 1, \ldots, T$, and then use $\hat{m}_t(\cdot)$ to estimate expected stock returns $\hat{x}_{i,t}$.

$$\hat{m}_t(\cdot) = \arg \min_{m \in \mathrm{DNN}} \sum_{i=1}^{N} \left(x_{i,t} - m(z_{i,t-1})\right)^2, \quad t = 1, \ldots, T \tag{11}$$

$$\hat{x}_{i,t} = \hat{m}_t(z_{i,t-1}) \tag{12}$$

2. Apply local principal component analysis (local PCA) on the estimated expected returns $\hat{x}_t \in \mathbb{R}^N$ to estimate the stock betas $\hat{\beta}_t \in \mathbb{R}^{N \times K_f}$:

$$S_t = \frac{1}{T} \sum_{s=1}^{T} K_{s,t}\, \hat{x}_s\, \hat{x}_s^\top \tag{13}$$

where $K_{s,t}$ is a time-dependent weight that is chosen to be close to 0 when s and t are far apart.

$$\hat{\beta}_{t-1} = \sqrt{N} Q_{K_f}, \tag{14}$$

where Q_{K_f} is the $N \times K_f$ eigenvector matrix of S_t, corresponding to the top K_f eigenvalues.

3. Let $\hat{\lambda}_{t-1} \in \mathbb{R}^{N \times K_g}$ denote the first K_g columns of $\hat{\beta}_{t-1}$, and then estimate BM factors $\hat{g}_t$ by:

$$\hat{g}_t = \frac{1}{N} \sum_{i=1}^{N} \hat{\lambda}_{t-1}\, v_{i,t} \tag{15}$$

We then forecast the market index return by:

$$\hat{y}_{T+1|T} = \hat{\rho}_0 + \hat{\rho}_g^\top\, \hat{g}_T, \tag{16}$$

where $\hat{\rho}_0$ and $\hat{\rho}_g$ are estimated using the time-series regression:

$$y_{t+1} = \hat{\rho}_0 + \hat{\rho}_g^\top\, \hat{g}_T + u_{i,t+1} \tag{17}$$

4. Finally, construct the forecast confidence interval for the expected index return as:

$$\left[\hat{y}_{T+1|T} - z_\tau\, \mathrm{SE}(\hat{y}_{T+1|T}), \hat{y}_{T+1|T} + z_\tau\, \mathrm{SE}(\hat{y}_{T+1|T})\right], \tag{18}$$

where z_τ is the $1 - \tau$ critical value for the standard normal distribution.

3.2 Intuitions of the Algorithm

1. **Constructing firm-level expected returns using the cross-sectional neural network regression**

   Since the firm-level realized returns can be very noisy, the paper proposes to work with the firm-level conditional expected returns given the characteristics, which are free of idiosyncratic noise but preserve the factor structure:

   $$\mathbb{E}_t(x_{i,t} \mid z_{i,t-1}) = h_{\beta,t}(z_{i,t-1})^\top f_t \tag{19}$$

   This formulation reflects the idea that a firm's expected excess return at time t, conditional on its lagged characteristics $z_{i,t-1}$, is driven entirely by latent factors. The function $h_{\beta,t}(z_{i,t-1})$ plays the role of a time-varying factor loading, linking firm characteristics to the factors f_t.

   The realized excess return $x_{i,t}$ can be decomposed as:

   $$x_{i,t} = \mathbb{E}_t(x_{i,t} \mid z_{i,t-1}) + \varepsilon_{i,t} \tag{20}$$

   where $\varepsilon_{i,t}$ is a firm-specific error term representing idiosyncratic shocks. Since this noise is independent across firms and has mean zero given characteristics, averaging over a large cross-section allows us to recover the conditional expectation.

   To estimate this expectation, the algorithm fits a feedforward neural network $m_t(\cdot)$ at each time t using a cross-sectional regression as shown in the equation (11), where DNN denotes the class of neural network functions. The fitted value $\hat{x}_{i,t} = m_t(z_{i,t-1})$ serves as an estimate of the conditional expected return for firm i.

   This estimation strategy leverages the large cross-sectional dimension N: even though only a single time period is used, the neural network can accurately learn the mapping from characteristics to expected returns. As the sample size grows, the approximation error vanishes:

   $$\hat{x}_{i,t} \xrightarrow{p} h_{\beta,t}(z_{i,t-1})^\top f_t, \quad \text{as } N \to \infty \tag{21}$$

   In other words, as the cross-sectional sample size N grows large, the estimated returns $\hat{x}_{i,t}$ converge in probability to the systematic component $h_{\beta,t}(z_{i,t-1})^\top f_t$, effectively filtering out idiosyncratic noise while preserving the underlying factor structure.

2. **Using local PCA to estimate factors and loadings**

   After obtaining the denoised firm-level expected returns $\hat{x}_{i,t}$, the next step is to estimate the time-varying factor loadings $\beta_{i,t}$. Since the expected returns retain the same factor structure as realized returns, we assume:

   $$\hat{x}_{i,t} \approx \beta_{i,t-1}^\top f_t. \tag{22}$$

This implies that the matrix of expected returns is approximately low-rank. Thus, principal component analysis (PCA) can be used to uncover the underlying factor structure.

However, standard PCA treats all time periods equally, which fails to capture the time variation in factor loadings. To address this, the paper adopts *local PCA*, where observations closer in time to the current period t are assigned higher weights. Specifically, it computes a weighted covariance matrix as shown in equation (13), where $\hat{x}_s = (\hat{x}_{1,s}, \dots, \hat{x}_{N,s})^\top$, and $K_{s,t}$ is a time-dependent kernel weight.

The kernel weights are defined as:

$$K_{s,t} = \frac{1}{h} K\left(\frac{s-t}{Th}\right) A_t^{-1}, \quad A_t = \frac{1}{Th} \sum_{l=1}^{T} K\left(\frac{l-t}{Th}\right) \tag{23}$$

where h is the bandwidth parameter and $K(\cdot)$ is a symmetric kernel function. This paper uses the two-sided quartic kernel:

$$K(x) = \frac{15}{16}(1-x^2)^2, \quad -1 \leq x \leq 1. \tag{24}$$

Because the kernel function downweights distant observations, only values of $\hat{x}_s$ with $s \approx t$ meaningfully contribute to S_t. For nearby s, we can approximate the expected returns by:

$$\hat{x}_s = \beta_{t-1}^\top f_s + o_P(1) \tag{25}$$

so that:

$$S_t = \beta_{t-1} S_{F,t} \beta_{t-1}^\top + o_P(1) \tag{26}$$

where $S_{F,t} = \frac{1}{T}\sum_{s=1}^{T} f_s f_s^\top K_{s,t}$ is the weighted factor covariance matrix. Hence, the top K eigenvectors of S_t span the same space as β_{t-1}, up to rotation.

Importantly, the paper shows that this procedure yields a *consistent estimator* of the factor loadings. Specifically, for some rotation matrix H,

$$\left\| \hat{\beta}_{i,t-1} - \beta_{i,t-1} H \right\| = o_P(1) \tag{27}$$

where the approximation error vanishes as the number of firms $N \to \infty$, regardless of the time-series length T. This means the local PCA approach remains valid even with relatively short time series, provided there is a sufficiently large cross-section.

3. **Constructing the BM-factors from the estimated stock-betas**

   The next step in the algorithm is to construct the BM-factors g_t, which are designed to capture the common variation in firm valuation ratios such as book-to-market. The key innovation of the paper lies in constructing these factors without directly estimating the loadings $\lambda_{i,t-1}$ in the valuation model:

$$v_{i,t} = \lambda_{i,t-1}^{\top} g_t + \eta_{i,t} \tag{28}$$

   where $v_{i,t}$ represents a valuation-related signal, and $\eta_{i,t}$ is an idiosyncratic error.

   Instead of estimating $\lambda_{i,t-1}$, the paper uses the stock-return factor loadings $\beta_{i,t-1}$, obtained via local PCA, as instruments. Since we assume that $K_f \geq K_g$, we can use a sub-vector of $\hat{\beta}_{i,t-1}$ consisting of its first K_g elements to construct a vector of instrumental variables:

$$\hat{\lambda}_{i,t-1}^{IV} := (\hat{\beta}_{i,t-1,1}, ..., \hat{\beta}_{i,t-1,K_g})^{\top} \tag{29}$$

   which satisfy two key IV conditions:

   - **Relevance**: $\hat{\lambda}_{i,t-1}^{IV}$ is correlated with $\lambda_{i,t-1}$, since both reflect systematic exposure to firm characteristics;
   - **Exogeneity**: $\hat{\lambda}_{i,t-1}^{IV}$ is uncorrelated with $\eta_{i,t}$, assuming firm characteristics are exogenous.

   The BM-factor is then constructed using a two-stage instrumental variables estimator:

$$\hat{g}_t = \left(\sum_{i=1}^{N} \hat{\lambda}_{i,t-1}^{IV} \left(\hat{\lambda}_{i,t-1}^{IV} \right)^{\top} \right)^{-1} \sum_{i=1}^{N} \hat{\lambda}_{i,t-1}^{IV} v_{i,t} = \frac{1}{N} \sum_{i=1}^{N} \hat{\lambda}_{i,t-1}^{IV} v_{i,t} \tag{30}$$

   The final equality holds because the instrument vectors $\hat{\lambda}_{i,t-1}^{IV}$ are orthonormal (eigenvectors), satisfying:

$$\sum_{i=1}^{N} \hat{\lambda}_{i,t-1}^{IV} \left(\hat{\lambda}_{i,t-1}^{IV} \right)^{\top} = N I_{K_g} \Rightarrow \left(\sum_{i=1}^{N} \hat{\lambda}_{i,t-1}^{IV} \left(\hat{\lambda}_{i,t-1}^{IV} \right)^{\top} \right)^{-1} = \frac{1}{N} I_{K_g} \tag{31}$$

   To see why this estimator is consistent, we substitute the valuation model into the right-hand side:

$$\hat{g}_t = \left(\frac{1}{N} \sum_i \tilde{\lambda}^{IV}_{i,t-1} \tilde{\lambda}^{IV\top}_{i,t-1} \right)^{-1} \left(\frac{1}{N} \sum_i \tilde{\lambda}^{IV}_{i,t-1} (\lambda^\top_{i,t-1} g_t + \eta_{i,t}) \right) \tag{32}$$

Expanding this gives:

$$\hat{g}_t = \underbrace{\left(\frac{1}{N} \sum_i \tilde{\lambda}^{IV}_{i,t-1} \tilde{\lambda}^{IV\top}_{i,t-1} \right)^{-1} \left(\frac{1}{N} \sum_i \tilde{\lambda}^{IV}_{i,t-1} \lambda^\top_{i,t-1} \right)}_{=:H_g} g_t + \underbrace{\left(\frac{1}{N} \sum_i \tilde{\lambda}^{IV}_{i,t-1} \tilde{\lambda}^{IV\top}_{i,t-1} \right)^{-1} \left(\frac{1}{N} \sum_i \tilde{\lambda}^{IV}_{i,t-1} \eta_{i,t} \right)}_{=:\tilde{\eta}_t} \tag{33}$$

That is,

$$\hat{g}_t = H_g g_t + \tilde{\eta}_t. \tag{34}$$

Now under the following assumptions:

- **Instrument strength**: H_g is full rank (relevance);
- **Idiosyncratic shock independence**: $\mathbb{E}[\tilde{\lambda}^{IV}_{i,t-1} \eta_{i,t}] = 0$, and $\mathrm{Var}(\eta_{i,t}) < \infty$;
- **Cross-sectional independence**: As $N \to \infty$, the sample averages converge in probability (LLN).

we obtain:

- $\tilde{\eta}_t \xrightarrow{P} 0$, by LLN, since it is an average of zero-mean i.i.d. terms;
- $H_g \xrightarrow{P}$ nonsingular matrix, so invertibility is preserved.

Therefore,

$$\hat{g}_t = H_g g_t + o_P(1) \tag{35}$$

which shows that $\hat{g}_t$ is a consistent estimator of the true BM-factor g_t, up to rotation.

This is a key strength of the method: consistency does not require a large time dimension T, but only a large cross-section N. The estimator handles idiosyncratic noise through projection and averaging, solving the error-in-variables problem and allowing valid factor construction from noisy valuation signals.

4. **Economic Interpretation of the $\hat{g}_t$-factor**

   Traditional financial research often constructs aggregate firm-level variables—such as book-to-market (BM) ratios—by taking weighted averages. A common weighting scheme is to use firm-level market capitalization. For instance, an aggregate BM measure can be constructed as:

$$\tilde{g}_{t,abm} = \sum_{i=1}^{N} \tilde{w}_{it,mk} v_{i,t}, \quad w_{it,mk} \text{ is firm } i\text{'s market capitalization.} \tag{36}$$

   where

$$\tilde{w}_{it,mk} = \frac{w_{it,mk}}{\sum_{j=1}^{N} w_{jt,mk}} \tag{37}$$

   However, our CML method constructs $\hat{g}_t$ differently. Instead of market-cap weights, we implicitly use firm-level stock betas—extracted via local PCA—as the weights in forming the BM-factor. In other words, the difference between $\hat{g}_t$ and traditional market-aggregated variables lies in the choice of weighting: stock betas versus market cap.

   There is a key economic rationale for this choice. Market capitalization reflects realized returns and contains considerable idiosyncratic noise, which does not necessarily provide predictive information for future market returns. As a result, market-cap-based aggregations, such as $\tilde{g}_{t,abm}$, tend to be noisy and less effective as predictors.

   By contrast, the stock betas used in our method are constructed from firm characteristics and designed to capture the expected components of return, which are much cleaner and more stable. These betas represent systematic exposures and serve as economically meaningful weights. Therefore, the BM-factor $\hat{g}_t$ derived via our IV approach offers a less noisy, more predictive summary of cross-sectional information than simple market-cap-weighted averages, while still retaining forecasting power for aggregate market returns.

5. **Why Traditional BM-Factor Estimation is Not Robust to Idiosyncratic Noise When T is Small**

   Traditional methods like PCA or PLS estimate BM-betas and BM-factors by directly fitting the factor model over time from equation (28), where $v_{i,t}$ is the book-to-market ratio of firm i at time t, λ_i is a fixed factor loading (assumed constant over time), g_t is the time-t BM-factor, and $\eta_{i,t}$ is the idiosyncratic error term.

   To estimate λ_i from this model, one typically applies ordinary least squares (OLS) over T time periods. The estimator is:

$$\tilde{\lambda}_i = \left(\sum_{t=1}^{T} g_t g_t^\top \right)^{-1} \left(\sum_{t=1}^{T} g_t v_{i,t} \right) \tag{38}$$

Substituting the model (28) into this expression, we obtain:

$$\tilde{\lambda}_i = \lambda_i + \left(\sum_{t=1}^{T} g_t g_t^\top \right)^{-1} \left(\sum_{t=1}^{T} g_t \eta_{i,t} \right) \tag{39}$$

Under the normalization assumption that $\sum_t g_t g_t^\top \approx TI$, the estimation error becomes:

$$\tilde{\lambda}_i - \lambda_i \approx \frac{1}{T} \sum_{t=1}^{T} g_t \eta_{i,t} \tag{40}$$

This reveals a key limitation of the traditional approach: the estimation error in $\tilde{\lambda}_i$ is driven by the average of the idiosyncratic noise $\eta_{i,t}$ over T periods. As such, the error scales with $\mathcal{O}_P\left(\sqrt{\frac{\mathrm{Var}(\eta_{i,t})}{T}}\right)$, and only diminishes when T is large.

Furthermore, this estimation error carries over to the estimation of g_t, because $\tilde{\lambda}_i$ is used in a second-step regression to recover the factor:

$$\hat{g}_t = \text{function of } \tilde{\lambda}_i.$$

The induced bias in g_t due to the error-in-variables (EIV) problem can be approximated by:

$$\frac{1}{N} \sum_i \eta_{i,t} \tilde{\eta}_i \approx \frac{1}{T} \cdot \frac{1}{N} \sum_i g_t \cdot \mathrm{Var}(\eta_{i,t}) \tag{41}$$

where $\tilde{\eta}_i$ is the error in the estimated λ_i.

This result shows that when T is small, the EIV problem leads to substantial noise being transferred into the estimated g_t. Therefore, traditional methods that rely purely on time-series fitting of $v_{i,t}$ suffer from non-robustness when T is limited or when $\eta_{i,t}$ is volatile.

By contrast, our CML approach avoids this issue by constructing λ_i from the cross-sectional structure of predicted returns $x_{i,t}$ via deep learning, which reduces dependence on long T and mitigates the impact of idiosyncratic volatility on factor estimation.

4 Contribution

In this section, we discuss the key contributions of this paper in comparison to two commonly used machine learning approaches.

4.1 "Naive" CML approach

The naive conditional machine learning approach simply applies the cross-sectional DNN to the last period:

$$\hat{m}_T(z) = \arg\min_{m\in\text{DNN}} \sum_{i=1}^{N} \left(x_{i,T} - m(z_{i,T-1})\right)^2,$$

and then use $\hat{m}_T(\cdot)$ and $z_{i,T}$ to forecast $\hat{y}_{T+1}^{\text{Naive}}$:

$$\hat{y}_{T+1}^{\text{Naive}} = \sum_{i=1}^{N} w_i \hat{m}_T(z_{i,T}).$$

Given the stock-return factor model:

$$x_{i,t} = h_{\beta,t}(z_{i,t-1})^\top f_t + u_{i,t},$$

where the factor loading function $\hat{h}_{\beta,t}(z)$ and the latent factor realization $\hat{f}_t$ are both estimated by the DNN and contained in $\hat{m}_t(z)$. The cross-sectional DNN on the last period removes the idiosyncratic error $u_{i,T}$, so

$$\hat{m}_T(z) \xrightarrow{p} h_{\beta,T}(z)^\top f_T.$$

When we plug in the updated firm characteristic inputs $z_{i,T}$ into the model $\hat{m}_T(z)$ to obtain the estimated $\hat{x}_{i,T+1}$, we get

$$\hat{x}_{i,T+1} = \hat{m}_T(z_{i,T}) \xrightarrow{p} h_{\beta,T}(z_{i,T})^\top f_T,$$

and

$$\hat{y}_{T+1}^{\text{Naive}} = \sum_{i=1}^{N} w_i \hat{m}_T(z_{i,T}) \xrightarrow{p} \sum_{i=1}^{N} w_i \, h_{\beta,T}(z_{i,T})^\top f_T.$$

However, the true out-of-sample return y_{T+1} should be

$$y_{T+1} = \sum_{i=1}^{N} w_i \, x_{i,T+1} \approx \sum_{i=1}^{N} w_i \, \beta_{i,T}^\top f_{T+1}.$$

We see that there's a substantial gap between the predicted and true out-of-sample return due to the difference in f_T and f_{T+1}, which is generally not assumed to remain the same across time.

4.2 Pooled ML Approach

The pooled ML approach is to get a single model $\hat{m}(\cdot)$ using data pooled over all time periods and cross-sections:

$$\hat{m}(z) = \arg\min_{m\in\text{ML}} \sum_{t=1}^{T}\sum_{i=1}^{N} \left(x_{i,t} - m(z_{i,t-1})\right)^2,$$

and then use $\hat{m}(\cdot)$ and $z_{i,T}$ to predict $\hat{y}_{T+1}^{\text{Pooled}}$:

$$\hat{y}_{T+1}^{\text{Pooled}} = \sum_{i=1}^{N} w_i \hat{m}(z_{i,T}).$$

According to Equation (8) and the assumption that $\rho_{0,T}$ and $\rho_{g,T}$ are constants, we have

$$y_{T+1} = y_{T+1|T} + \epsilon_{T+1} = \rho_0 + \rho_g^\top g_T + \epsilon_{T+1}$$

Let $\mathcal{F}_{z,T}$ denote the filtration (information set) generated by the firm characteristics $z_{i,t}$ up to time T. This is a strictly smaller information set than $\mathcal{F}_T$ as it excludes the realized factors g_t observed up to time T. Thus the conditional expected return $y_{T+1|T}$ can be decomposed as

$$\begin{aligned} y_{T+1|T} &= \mathbb{E}\left[y_{T+1} \mid \mathcal{F}_T\right] \\ &= \mathbb{E}\left[\rho_0 + \rho_g^\top g_T \mid \mathcal{F}_T\right] \\ &= \rho_0 + \rho_g^\top g_T \quad (\text{since } g_T \text{ is } \mathcal{F}_T \text{ measurable, then } \mathbb{E}\left[g_T \mid \mathcal{F}_T\right] = g_T) \\ &= \mathbb{E}\left[y_{T+1} \mid \mathcal{F}_{z,T}\right] + \rho_g^\top \left(g_T - \mathbb{E}\left[g_T \mid \mathcal{F}_{z,T}\right]\right). \end{aligned}$$

It has been shown that the pooled machine learning predictor captures the unconditional expected return (the expected return given only firm characteristics information $z_{i,t}$ up to time T), but not the factor realization:

$$\hat{y}_{T+1}^{\text{Pooled}} \xrightarrow{p} \mathbb{E}\left[y_{T+1} \mid \mathcal{F}_{z,T}\right].$$

Thus,

$$\begin{aligned} y_{T+1} &= y_{T+1|T} + \epsilon_{T+1} \\ &= \mathbb{E}\left[y_{T+1} \mid \mathcal{F}_{z,T}\right] + \rho_g^\top \left(g_T - \mathbb{E}\left[g_T \mid \mathcal{F}_{z,T}\right]\right) + \epsilon_{T+1} \\ &= \hat{y}_{T+1}^{\text{Pooled}} + \rho_g^\top \left(g_T - \mathbb{E}\left[g_T \mid \mathcal{F}_{z,T}\right]\right) + \epsilon_{T+1}, \end{aligned}$$

which implies that the forecast error of $\hat{y}_{T+1}^{\text{Pooled}}$:

$$y_{T+1} - \hat{y}_{T+1}^{\text{Pooled}} = \rho_g^\top \left(g_T - \mathbb{E}\left[g_T \mid \mathcal{F}_{z,T}\right]\right) + \epsilon_{T+1}.$$

In contrast, the proposed approach in this paper forecasts the conditional expected return:

$$\hat{y}_{T+1|T} \xrightarrow{p} y_{T+1|T},$$

and the forecast error of $\hat{y}_{T+1|T}$:

$$y_{T+1} - \hat{y}_{T+1|T} = \epsilon_{T+1}.$$

Therefore, $\hat{y}_{T+1|T}$ has strictly **less forecast error variance** than $\hat{y}_{T+1}^{\text{Pooled}}$ asymptotically (though not necessarily lower forecast error for every realization). The conditional ML approach dominates the pooled ML by capturing the factor realization.

5 Implementation

5.1 Dataset

We built our empirical analysis and prediction models on several comprehensive datasets at the firm level, including a variety of characteristics. We used two datasets: firm-level dataset originally compiled by Chen and Zimmermann (2021), and characteristics data collected and imputed by Bryzgalova, Pelger (2025.)

5.1.1 Dataset of 87 characteristics

Inspired by the paper by Chen and Zimmermann (2021), this dataset assembles 87 characteristics, capturing a spectrum of fields, including valuation, profitability, risk, liquidity, investment, or event-driven information, used to be fitting during the first step of the algorithm. All characteristics are drawn monthly from CRSP (return, prices, volumes) and Compustat(accounting data). Each characteristic adopts monthly frequency, and is rank-transformed to the [0,1] for comparability and reducing the effect of outliers. Those characteristics span January 1955 through December 2021 and correspond to US-listed common stock trading on NYSE, NASDAQ, and AMEX. This dataset is comprehensive, capturing both long-term information, short-term dynamics, and conditional/special shift, relevant for cross-sectional factor estimation. However, we noticed that some characteristics suffer from severe missing ratios. For example, the Firm Age characteristic has 62.62 of missing ratio. To handle that issue, we implemented cross-sectional linear imputation.

5.1.2 Dataset of 45 Characteristics

Aiming for a dataset which simultaneously support effective training and resilience against missing ratio, we adopt the 45-characteristic dataset [2]. This dataset narrows the universe to 45 characteristics by retaining only empirically informative characteristics and discarding idiosyncratic ones. Specifically, this data set is more careful in picking features with high missing ratios and adds new categories (such as intangibles and trading friction measures). more efficient in capturing modern factors correlated with cross-section return. This data set also incorporates a more efficient two-step latent factor imputation method that considers both cross-sectional and time series dependency.

Table 1 is the table of sections covered by the characteristics in both datasets. Notice that even though the new dataset has fewer characteristics, it covers more sections (Intangibles and Trading Fictions sections)

5.2 Experiment Setting

In this section we describe in detail our replication of the Conditional Machine Learning (CML) forecasting approach. All code is implemented in Python (Py-

Characteristic Section	**Examples**	**Relevance**
Past Returns	r2_1, r12_2, LT_Rev	Use historical return patterns to forecast.
Investment	Investment, NOA, DPI2A, NI	Measure capital allocation and financing activities.
Profitability	PROF, ATO, PM, ROA, SGA2S	Measure operating and earnings generation to indicate the corporate performance.
Intangibles	AC, OA, OL, PCM	Capture accounting-based, non-cash signals of earnings quality.
Value	BEME, A2ME, CF2P, D2P, Q	Measure the corporate value thought fundamentals, including valuation ratios and balance-sheet metrics.
Trading Frictions	Spread, IdioVol, LTurnover, Resid_Var, SUV	Quantify liquidity constraints and short-term return anomalies from market microstructure.

Table 1: Overview of Characteristic Sections

Torch, scikit-learn, statsmodels). Below we outline data preparation, model architecture, training and out-of-sample evaluation.

5.3 In-Sample Estimation (Step I–III(a))

- **Sample split:** We use data up to December 1983 (`TRAIN_END=2018`) for in-sample estimation. Out-of-sample forecasting begins in January 1984 (`OOS_START=198401`) and ends in December 1989 (`OOS_END=198912`).
- **Neural network (Step I):**
  - Architecture: a two-layer feed-forward network. Input dimension is p, one hidden layer of size $H = 32$, ReLU activation, and a linear output node.
  - Loss: mean squared error (MSE).
  - Optimizer: Adam with learning rate $\alpha = 10^{-4}$.
  - Batch size: 64; epochs per cross-section: 1 (for demo; can be increased to 2000 for full replication).
- **Local Cross-Sectional PCA (Step II):**

  For each month t, let $\hat{x}_{i,t}$ denote the fitted values from the DNN models. Following the local PCA framework, we apply an **exponential decay kernel** to assign greater weight to the expected returns $\hat{x}_{i,t}$ for months closer to the prediction target $T+1$, and progressively lower weight to those further in the past:

$$K_{s,T} = \frac{(1-\alpha)\alpha^{T-s}}{1-\alpha^T}, \quad s = 1, 2, \ldots, T, \quad \text{where } 0 < \alpha < 1.$$

We choose $\alpha = 0.5$. Using the corresponding exponential decay weights, we compute the weighted covariance matrix and perform cross-sectional PCA as follows:

$$\hat{\Sigma} = \hat{X} K \hat{X}^\top \quad \text{where } \hat{X} \in \mathbb{R}^{N \times T}.$$

- **BM-factor (Step III(a)):**

$$g_t \;=\; \frac{1}{N_t} \sum_{i=1}^{N_t} \beta_{i,t-1} \cdot B2M_{i,t}.$$

We store each month's g_t in `g_train.csv`.

5.4 Forecasting (Step IV)

1. For each forecast month from Jan 2018 to Dec 2020:

   (a) Re-estimate the DNN on all data up to and including month m (expanding window).

   (b) Predict $\hat{x}_{i,m}$ for the cross-section at m, compute $\beta_{i,m-1}$ via local PCA, and form

   $$g_m \;=\; \frac{1}{N_m} \sum_i \beta_{i,m-1} \, B2M_{i,m}.$$

   (c) Forecast the index return at $m+1$ by

   $$\hat{y}_{m+1} = \hat{\rho}_0 \;+\; \hat{\rho}_g \, g_m.$$

2. We collect all out-of-sample forecasts and actuals and compute

$$\text{OOS-MSE} \;=\; \frac{1}{T} \sum (y_{t+1} - \hat{y}_{t+1})^2,$$

$$R^2_{\text{OOS}} \;=\; 1 - \frac{\sum (y - \hat{y})^2}{\sum (y - \bar{y})^2},$$

6 Empirical Results

6.1 Out-of-Sample R^2 Dynamics

Figure 1 plots two time-series of out-of-sample R^2 for our CML forecast:

- **[t:end]** R_t^2: cumulative performance from month t through the end of the sample.
- **[1:t]** R_t^2: cumulative performance from the first OOS month up to month t.

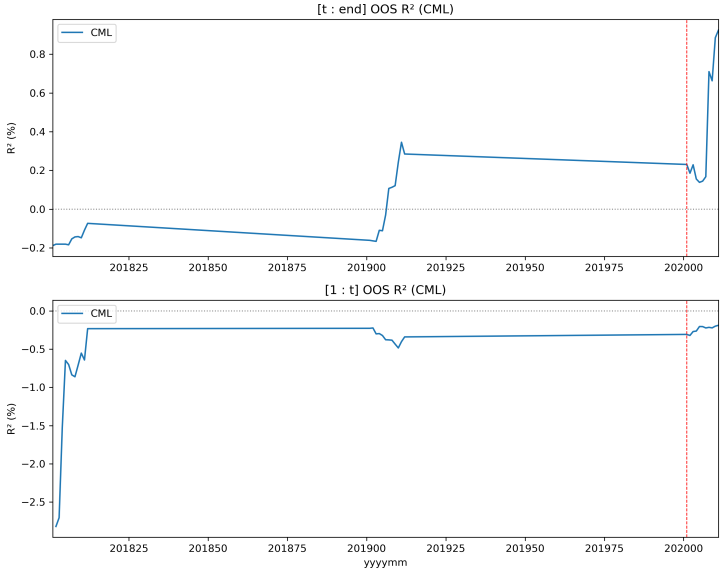

Figure 1: Out-of-sample R^2 for CML. *Top:* [t:end] series. *Bottom:* [1:t] series. The dashed vertical lines mark subperiod boundaries.

7 Analysis

Our implementation of the conditional machine learning framework indicates advantages over traditional factor models. By training deep neural networks to denoise expected returns and then applying PCA, the method effectively mitigates idiosyncratic volatility without relying on long term panels. This two-step procedure preserves the underlying factor structure, capturing short-term trends in index returns more reliably than standard PCA alone. The incorporation of the Creative Destruction Index (CDI) and the derived confidence intervals provides economically interpretable insights into the uncertainty of the forecast.

However, our findings also highlight important limitations.

1. Conditional ML framework hinges on a rich cross-sectional panel: over 10,000 stocks and 87 characteristics. The dataset suffers from severe missing ratios. Imputation and multi-layers networks make the method both

prone to overfitting and computationally intensive. These practical challenges may limit the model's scalability and real-time applicability. The smaller dataset (2,000 stocks and 45 characteristics) and model we used, on the other hand, is not large enough to capture the "cross-sectional" trend suggested in the paper, even though advantages in efficiency and better handling of missing ratio.

2. The methodology proposed in this paper relies on several key assumptions. However, as noted in the paper, the validity of these assumptions may exhibit temporal variability. That helps explain the model's strong performance during certain periods, but also its difficulty in achieving consistent performance over the long term.

3. The multi-step structure of the proposed model involved multiple design choices, such as the architecture of the DNN, the kernel function used in local PCA, and the number of principal components retained, which makes it challenging to reproduce the consistent results reported in the paper and suggests that the model is sensitive to specific design parameters.

Future work could explore hybrid architectures, merging period-by-period cross-sectional denoising with time-series models—such as LSTMs or transformer within an AutoML framework— capturing both cross-sectional and short-term temporal dependencies. Evaluating tree-based learners, like LightGBM, during the stock-level predictions may reveal strengths.

References

[1] Ravi Jagannathan, Yuan Liao, and Andreas Neuhierl. Robust stock index return predictions using deep learning. *Available at SSRN 4890466*, 2023.

[2] Markus Svetlana, Martin. Missing financial data. *Available at SSRN 4106794*, 2022.