**Master in Financial Engineering**

**FIN-407 Machine Learning in Finance**

**Spring Semester 2025**

---

# Factoring Complexity: Random Features and Transformers in Asset Pricing

---

Gabriele Calandrino
Francesco Mutti
Martim Prada
Federico Sabbatani Schiuma

# Contents

**Abstract**

This project investigates the benefits of complexity in portfolio construction. We propose two distinct approaches based on the stochastic discount factor framework: a ridge regression model and a transformer-based deep learning model. The first leverages stock characteristics expansion via nonlinear feature maps while maintaining linear estimation of risk prices, whereas the transformer model captures nonlinear dependencies and cross-sectional interactions via self-attention mechanisms. Empirical findings highlight that increasing model complexity can yield substantial improvements in risk-adjusted returns, in both absolute terms and relative traditional factor models.

**Keywords:** Stochastic discount factor, ridge regression, factor models, deep learning, transformer, optimal portfolio

# 1 Introduction

The goal of this project is to investigate and analyze different machine learning methods applied to asset pricing. In particular, we aim to build optimal portfolios with a monthly frequency by leveraging models grounded in stochastic discount factor (SDF) theory.
In the SDF framework, asset prices are determined by their risk-adjusted expected payoffs. We know from the theory of SDF that the optimal portfolio weight of a single stock is given by

$$w_t = f(X_t)\lambda,$$

where $X_t \in \mathbb{R}^{N_t \times D}$ encodes the values of $D$ characteristics for all $N_t$ stocks at month $t$, $f(X_t)$ is the function that translates characteristics into signals and $\lambda$ is the shadow price of risk, which converts a signal into a position.

The literature [1] provides several interpretations of this framework, which vary a lot in complexity. Nevertheless, the general structure across models remains consistent: signals are generated using updated features $X_t$, and then are transformed into weights via an estimated $\lambda$, typically obtained over a rolling window of past data. Consequently, portfolio returns take the form

$$R^P_{t+1} = R^T_{t+1} w_t = R^T_{t+1} f(X_t)\hat{\lambda}.$$

Our goal is to test and compare two different approaches in this context, with different features mapping functions and different ways to estimate the optimal $\lambda$. Specifically, we develop first a ridge regression model using random features to increase the dimensionality of $f(X_t)$ while maintaining a relatively simple, linear estimation of $\lambda$. Secondly, we present a transformer-based model that keeps the characteristics in their original form but uses a deep learning architecture to estimate a more complex, nonlinear function for the portfolio weights.

While these models differ in architecture and learning strategy, they are fundamentally connected by a common goal: to approximate the SDF using data driven methods that balance model flexibility and estimation stability. We explore this trade-off in depth in the next sections, beginning with the theoretical foundations that justify each approach within the context of SDF-based asset pricing.

# 2 Methodology

## 2.1 Theoretical foundation

A conditional stochastic discount factor can be represented as

$$M_{t+1} = 1 - w(X_t)^\top R_{t+1} \tag{1}$$

where $X_t \in \mathbb{R}^{N_t \times D}$ is the matrix of stocks characteristics at time $t$. To explore the "virtue of complexity", instead of restricting the function $w(X_t)$ to few factors − as traditionally done by APT models − we aim to expand it and approximate it, for example with a neural network, as

$$w(X_t) \approx S_t(X_t)\lambda \tag{2}$$

where $S_t(X_t) \in \mathbb{R}^{N_t \times P}$ is matrix of random features, $P$ is an arbitrary parameter and $\lambda \in \mathbb{R}^P$ is a vector to be estimated. Plugging (2) in (1) gives

$$M_{t+1} \approx 1 - \lambda^\top S_t(X_t)^\top R_{t+1} = 1 - \lambda^\top F_{t+1} \tag{3}$$

where $F_{t+1} \in \mathbb{R}^P$ are the random factors (note that if $S_t(X_t) = X_t$ and $P = D$ we get the "traditional" factors returns). To estimate the vector $\lambda$, we use the Maximum Sharpe Ratio Regression (MSSR) as in [2]:

$$\hat{\lambda} = \arg\min_\lambda \left( \frac{1}{T}\sum_{t=1}^{T}(1 - \lambda^\top F_t)^2 + z\|\lambda\|^2 \right) = \left( zI + \frac{1}{T}\sum_{t=1}^{T} F_t F_t^\top \right)^{-1} \left( \frac{1}{T}\sum_{t=1}^{T} F_t \right), \tag{4}$$

where $T$ is the train set size (using a factor representation allows to estimate an unconditional SDF). It is immediate to prove that solving (4) is equivalent to solve

$$\arg\max_\lambda \left( \frac{1}{T}\sum_{t=1}^{T} \lambda^\top F_t - \frac{1}{2T}\sum_{t=1}^{T}(\lambda^\top F_t)^2 - z\|\lambda\|^2 \right) = \arg\max_\lambda \left( \frac{1}{T}\sum_{t=1}^{T} U(\lambda^\top F_t) - z\|\lambda\|^2 \right),$$

where $U(x) = x - \frac{x^2}{2}$, hence getting $\hat{\lambda}$ as the sample Markowitz portfolio of factors. The ridge penalty $z$ is required as in (4) the matrix $F_t F_t^\top$ has low rank due to the fact that $P >> T$.

In parallel, we explore an alternative approach to approximate the stochastic discount factor as defined in (4), modeling it using a transformer. This approach indeed allows not only to exploit the benefits of complexity, but also to make use of the attention mechanism.
The architecture of the model is deeply described in the next section; here we would like to stress that, following what we have just said, a natural loss function for the model is precisely the MSSR.

### 2.2 Transformer Model

The nonlinear portfolio transformer used in the experiments that follow is built according to [1], with multiple stacked blocks, each of them composed of both an attention and a feed-forward sublayer. Its architecture is illustrated in the Appendix, Figure 6.

The first sublayer applies a multi-head attention, which takes as input the matrix $X_t \in \mathbb{R}^{N_t \times D}$, where $N_t$ is the number of stocks available at time $t$ and $D = 132$ are the features (JKP Stock Characteristics). The multi-head attention yields:

$$\mathcal{A}(X_t) = \sum_{h=1}^{H} \sigma(X_t W_h X_t^\top) X_t V_h, \quad \in \mathbb{R}^{N_t \times D}.$$

Here $H$ is the number of heads, $W_h, V_h \in \mathbb{R}^{D \times D}$ are learnable parameters and $\sigma$ is a row-wise softmax operator, which maps each row (i.e., the information about each stock) to a probability distribution. It follows a residual connection that adds back the input $X_t$ to the output of the multi-head-attention:

$$\mathcal{A}^R(X_t) \;=\; \mathcal{A}(X_t) + X_t.$$

The output $\mathcal{A}^R(X_t)$ of the attention sublayer is fed to the second sublayer. The latter consists of a fully connected feed-forward network with one hidden layer of $d_{\mathcal{F}}$ neurons:

$$\mathcal{F}(Y) = \mathrm{ReLU}(YW_1 + \mathbf{1}b_1^\top)W_2 + \mathbf{1}b_2^\top, \quad Y := \mathcal{A}^R(X_t),$$

where the parameter matrix $W_1$ is of size $D \times d_f$, $b_1$ is $d_f \times 1$, $W_2$ is $d_f \times D$, and $b_2$ is $D \times 1$. Therefore, the output of the feed-forward network has the same dimension as the input, $N_t \times D$.

Again, we include a residual connection:

$$\mathcal{F}^R(Y) \;=\; \mathcal{F}(Y) + Y, \quad Y := \mathcal{A}^R(X_t).$$

This whole process represents one transformer block and is stacked $K$ times:

$$\mathcal{T}(X_t) \;=\; \mathcal{F}\big(\mathcal{A}(X_t) + X_t\big) + \mathcal{A}(X_t) + X_t$$

$$\dots$$

$$\mathcal{T}^{(k)}(X_t) \;=\; \mathcal{T}\big(\mathcal{T}^{(k-1)}(X_t)\big), \quad k = 1, \dots, K,$$

where the input to each additional transformer block is the output from the previous block.

The recursion result $\mathcal{T}^{(K)}(X_t)$ is a $N_t \times D$ matrix that is projected into the conditional SDF portfolio weights by a linear transformation with a final parameter vector $\lambda \in \mathbb{R}^D$:

$$w_t = \mathcal{T}^{(K)}(X_t)\lambda, \quad \in \mathbb{R}_t^N.$$

# 3 Empirical Findings

## 3.1 Data pre-processing

For the empirical analysis, we used monthly stock-level data from WRDS, specifically the 153 JKP stock characteristics for U.S. equities spanning 1963–2024. We first removed "nano" stocks, then discarded characteristics with over 34% missing values, retaining 132 relevant features. Next, we excluded stock-month entries with more than 30% missing values and eliminated rows lacking returns or identifiers. We then applied cross-sectional rank-standardization to each characteristic by month, mapping values to the range $[-0.5, +0.5]$, ensuring a median of zero. Remaining missing values were filled with the cross-sectional median (zero). This step reduces outliers and scaling issues, allowing comparability across months. The final dataset is sorted by month and stock, yielding an $N_t \times D$ characteristic matrix for each month $t$.

## 3.2 Random Features Model Performance

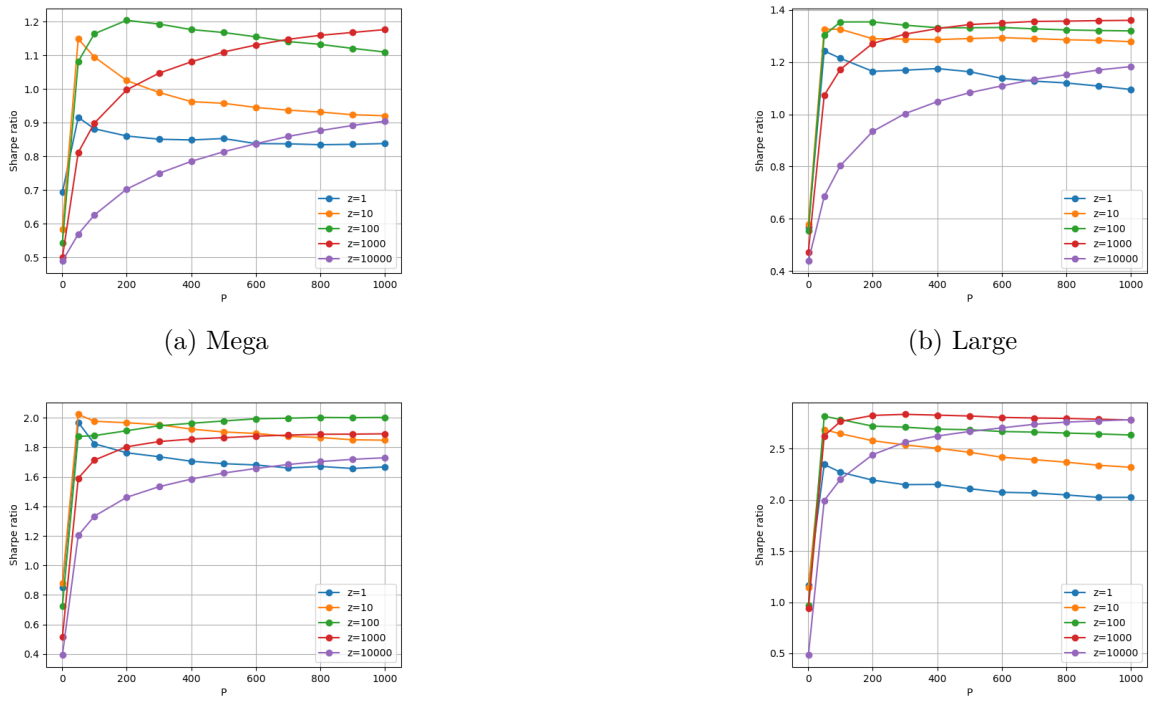
The success of the random features model lies in maximizing data complexity. To this end, we used the Random Fourier Features. We first built

$$\hat{S}_t(X_t; \gamma_g) = \text{concatenate}\Big(\cos(X_t W_t^{g\top}), \sin(X_t W_t^{g\top})\Big) \in \mathbb{R}^{N_t \times 2P},$$

where $W_t^g \sim \mathcal{N}(0, \gamma_g) \in \mathbb{R}^{P \times D}$. Next, we concatenated the results for $G$ random seeds:

$$S_t(X_t) = \text{concatenate}\Big(\hat{S}_t(X_t; \gamma_1), \dots \hat{S}_t(X_t; \gamma_g)\Big) \in \mathbb{R}^{N_t \times (2P \cdot g)}.$$

We set the number of random seeds $G$ to 10, leaving the parameter $\gamma_g$ always equal to 1, and tested the model for several values of number of random features $P$ to analyze how the Sharpe ratio of the strategy behaves. The model was trained using a rolling window of 360 months to create the optimal portfolio for every timestep. The same experiment was repeated for several ridge penalties − here we show only the best performing ones − and for all 4 size groups.




(a) Mega

(b) Large

(c) Small

(d) Micro

Figure 1: Comparison of Sharpe ratio vs $P$ across different ridge penalties

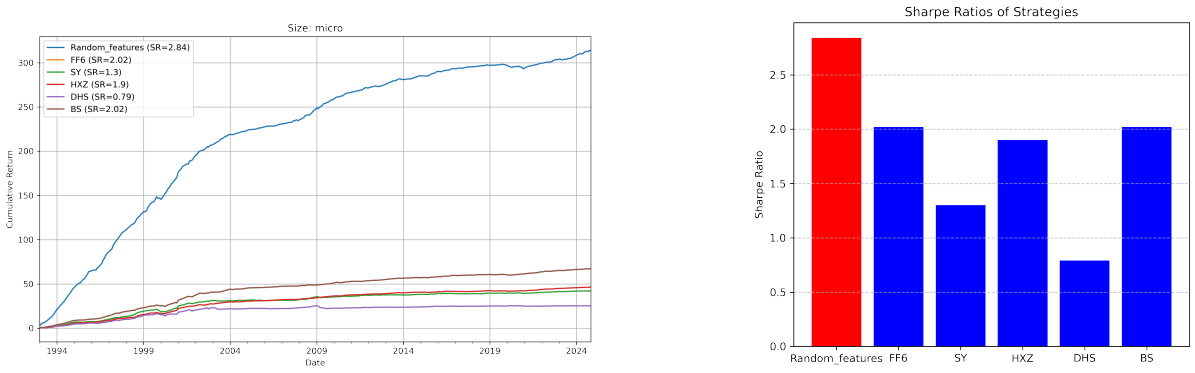
The monotonicity of the curve is strictly connected to the ridge penalty: for higher values, the Sharpe ratio keeps on increasing when adding random features, suggesting that a further increase in complexity would keep improving the performance. On the other hand, for lower ridge penalties the plots show a higher Sharpe ratio for small values of $P$, but fail to show a monotonic increase, suggesting that adding more features beyond that threshold would not lead to further performance gains.
As observed in the previous model, the group with the highest Sharpe ratio is again "micro", likely because it contains a significantly larger number of stocks compared to the other groups. Table 1 shows the alpha of the model with $z = 1000$ and $P = 1000$ against the benchmark proposed. In all the comparisons the model is able to generate positive and significant alpha.

| | FF6 | SY | HXZ | DHS | BS |
|---|---|---|---|---|---|
| Alpha | 0.49 | 0.59 | 0.45 | 0.70 | 0.49 |
| t-stat | 5.42 | 7.29 | 6.56 | 7.43 | 5.42 |

Table 1: Complex model alphas and t-statistics against benchmarks

To fully assess the model's performance, we compared its results with those of several benchmark models as in [2]

- **FF6**: a six-factor model including the five factors of Fama and French (2015) plus the UMD momentum factor. [3]
- **SY**: a four-factor model based on the "mispricing" factors proposed by Stambaugh and Yuan (2017). [4]
- **HXZ**: a five-factor q-model from Hou et al. (2015), extended with the expected growth factor from Hou et al. (2021). [5, 6]
- **DHS**: a three-factor behavioral model incorporating both long- and short-horizon factors, developed by Daniel et al. (2020). [7]
- **BSV**: a model using stochastic discount factor (SDF) weights linear in stock characteristics, following Brandt et al. (2009), estimated with a ridge penalty due to high dimensionality. [8]




(a) Cumulative Returns by Model

(b) Sharpe Ratios Comparison

Figure 2: Performance evaluation of Ridge-regularized models: cumulative returns and Sharpe ratios.

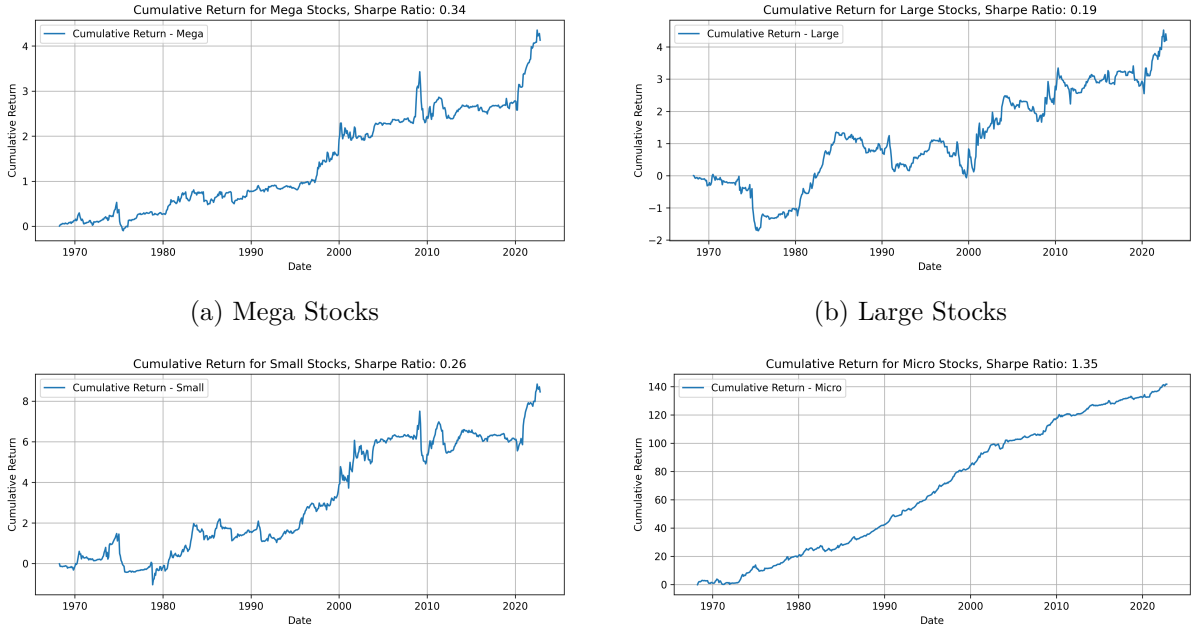
The results show that the portfolio returns obtained by the best model that uses random features (with parameters: $P = 300$, $z = 1000$) outperforms the same model but applied to few significant factors. This outcome suggests that when trying to form optimal portfolios by mean of ridge regression, having a higher number of features can lead to better results.

### 3.3 Transformer Model Performance

Unlike traditional asset-specific models, which determine each asset's weight solely from its own characteristics, this transformer leverages cross-sectional attention across all available assets, capturing broader interactions and delivering the "virtue of complexity" benefits outlined in Kelly et al. 2025.[1]

We trained our model using 60-month rolling windows, starting with signals from 01-1963 to 12-1967, and repeated over 20 epochs. After each training phase, a one-month out-of-sample SDF portfolio was constructed, the window was rolled forward by one month, and the training repeated.

We initialized the self-attention matrices $W_h, V_h \sim \mathcal{N}(0, 1/D)$; feed-forward layers as $W_1^{(k)} \sim \mathcal{N}(0, 1/d_f)$, $W_2^{(k)} \sim \mathcal{N}(0, 1/D)$, with zero biases; and output weights as $\mathcal{N}(0, 1/1000^2)$. All experiments were performed with the same hyperparameters as in [1][1], except for the ridge penalty. We tested various penalties on a dataset's subsample[2], and selected 10 as the value that maximized the Sharpe ratio over a log-spaced grid from 0.01 to 100.




(a) Mega Stocks

(b) Large Stocks

(c) Small Stocks

(d) Micro Stocks

Figure 3: Cumulative returns by size group. The Sharpe ratio for each group is shown in the title of each plot.

As we can clearly see from the graphs,micro stocks, which are those below the 20$^{\text{th}}$ percentile but above 1$^{\text{st}}$ in market capitalization, are those that overperform. We discovered that in some months, the Transformer was generating unrealistically large returns (300%) by shorting a broad set of micro-cap stocks and levering up to 30×, a behavior that is simply not feasible given their low liquidity.

[1] $d_f = 256$, $H = 1$, $K = 10$.

[2] We selected micro stocks with IDs divisible by 6 and 5.

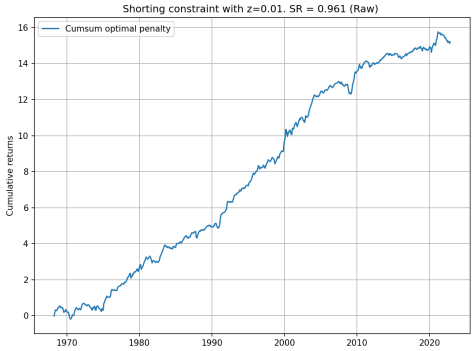
To investigate this phenomenon, we imposed a no-shorting constraint via a ReLU activation on the model's output layer:

$$w_t = \mathrm{ReLU}(\mathcal{T}^{(K)}(X_t)\,\lambda).$$

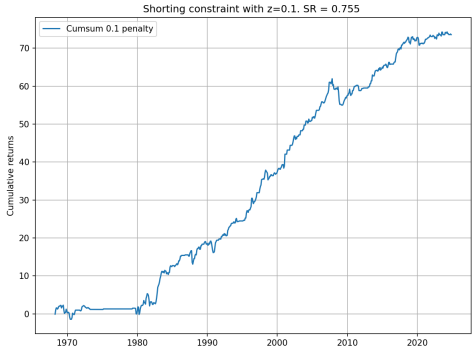
This constraint drastically reduced overall portfolio performance. The cumulative-return curve exhibits flat segments, which, since it is a cumulative sum, correspond to zero portfolio return over those intervals.

This effect stems from our loss function, $\mathcal{L} = \left(1 - R_{t+1}\, w_t\right)^2 + z\,\|w_t\|^2$. The first term drives the model toward predicting 100 % returns, while the ridge penalty $z$ guards against extreme leverage. Under the ReLU constraint, the model can no longer chase such returns, so it aims to minimize the penalty term simply collapsing all weights to zero.

Recognizing this, we re-tuned the ridge coefficient and found that $z = 0.01$ restores much of the lost performance without exposing to high leverages. Figure 4 compares the cumulative returns of the constrained Transformer with $z = 0.01$ versus $z = 0.1$ highlighting how now the returns are way more reasonable.




(a) Penalty $z = 0.01$




(b) Penalty $z = 0.1$

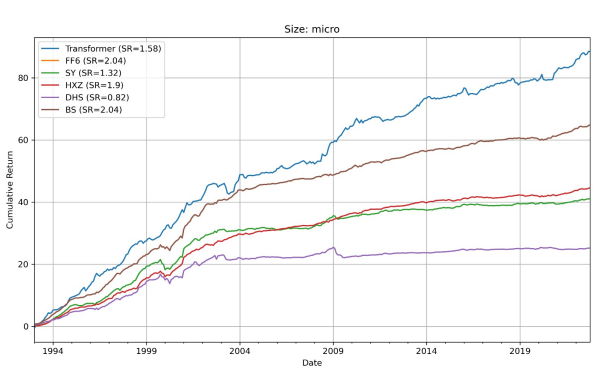
Figure 4: Cumulative returns of the Transformer with shorting constraints under different ridge penalties.

To avoid the plateau issue while preserving practical constraints, we relaxed the no-shorting condition introducing a leverage constraint: $\sum_{i=1}^{N_t} |w_i| \leq 1.5$. This new constraint allowed us to obtain better general performance. The highest improvement in the sharpe ratio was detected with a ridge penalty of 0.1, an increase from 0.75 to 0.93. (7)

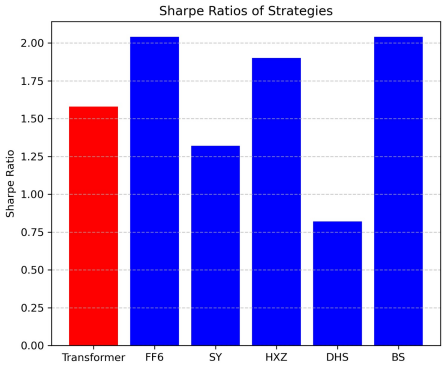
Finally, we followed the findings of Moreira and Muir [9] to test whether normalizing returns by past volatility would lead to better results. Using a 1-year volatility window, performance improved significantly for micro stocks (see Figure 8 in the appendix) but worsened for larger stocks. In particular, we noticed that during the 2008-crisis the raw return of larger stocks suffered a big drowdown, which appeared accentuated when normalized for the 1-year volatility [9]. This appears to be due to an overly long window. Reducing the window size led to a notable improvement in Sharpe ratios and smaller drawdowns. We present in Table 2 the different values of Sharpe ratio against different lengths of volatility window.

Table 2: Sharpe ratios for *large* stocks across different volatility windows

| Vol. window (months) | 12 | 6 | 3 |
|---|---|---|---|
| Sharpe ratio | 0.18 | 0.47 | 0.56 |




(a) Cumulative Returns by Model




(b) Sharpe Ratios Comparison

Figure 5: Performance evaluation of Transformer model: Cumulative returns and Sharpe ratios.

We conclude this result section by comparing the performance of our transformer with the benchmarks (see Figure 5). As we notice from the metrics our model beats only SY and DHS. Further studies could try to repeat our experiments with a fine-tuned volatility window and applying transaction costs, to inspect how the results could change.

# 4 Conclusion

This project explored the virtue of complexity through two portfolio construction models: a ridge regression and a transformed-based deep learning model.

Considering the simplicity of the first model, its performance, judging by the out of sample Sharpe ratios, can be considered very strong. The empirical results showed that the random feature model consistently outperformed traditional models (higher Sharpe ratios and significant and positive alphas), suggesting that increasing complexity can lead to better risk-adjusted performances. The main source of complexity of this framework lies in the computational costs associated with the drastic size changes of the dataset due to the introduction of random features. Future research could further explore this complexity, pushing it beyond our current computational constraints.

The transformer model, in contrast, did not achieve the performance levels reported in Kelly et al. (2025) [1], but still delivered informative and meaningful results. Our analysis focused on the impact of practical constraints, such as short-selling and leverage limits, which improved the model's reliability. While the model often showed higher returns compared to the benchmarks, its Sharpe ratio was not consistently superior due to higher out of sample volatility. One likely explanation for this underperformance is the model's highly complex architecture, involving roughly one million parameters, which demands significant computational resources. Future research could build on our work by leveraging greater computational capacity and conducting more refined hyperparameter tuning.

## References


[1] Bryan T Kelly et al. *Artificial Intelligence Asset Pricing Models*. Working Paper 33351. National Bureau of Economic Research, Jan. 2025. DOI: 10.3386/w33351. URL: http://www.nber.org/papers/w33351.

[2] Antoine Didisheim et al. "APT or "AIPT"? The Surprising Dominance of Large Factor Models". In: 23-19 (Mar. 2023). Available at SSRN: https://ssrn.com/abstract=4388526. URL: http://dx.doi.org/10.2139/ssrn.4388526.

[3] Eugene F Fama and Kenneth R French. "A five-factor asset pricing model". In: *Journal of Financial Economics* 116.1 (2015), pp. 1–22.

[4] Robert F Stambaugh and Yu Yuan. "Mispricing factors". In: *Review of Financial Studies* 30.4 (2017), pp. 1270–1315.

[5] Kewei Hou, Chen Xue, and Lu Zhang. "Digesting anomalies: An investment approach". In: *Review of Financial Studies* 28.3 (2015), pp. 650–705.

[6] Kewei Hou, Chen Xue, and Lu Zhang. "Replicating anomalies". In: *Review of Financial Studies* 34.5 (2021), pp. 2475–2527.

[7] Kent Daniel, David Hirshleifer, and Lin Sun. "A behavioral model of asset pricing". In: *Review of Financial Studies* 33.3 (2020), pp. 1455–1496.

[8] Michael W Brandt, Pedro Santa-Clara, and Rossen Valkanov. "Characteristics-based portfolio choice". In: *Review of Financial Studies* 22.9 (2009), pp. 3411–3447.

[9] Alan Moreira and Tyler Muir. "Volatility-Managed Portfolios". In: *Journal of Finance* (Oct. 2016). Forthcoming; Available at SSRN: https://ssrn.com/abstract=2659431. URL: http://dx.doi.org/10.2139/ssrn.2659431.

# Appendix

## MSSR Derivation

The MSSR allows to estimate the vector $\lambda$ as the sample Markowitz portfolio of factors:

$$\hat{\lambda} = \arg\min_{\lambda} \left( \frac{1}{T} \sum_{t=1}^{T} (1 - \lambda^\top F_t)^2 + z\|\lambda\|^2 \right)$$

Indeed we have that

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^{T} (1 - \lambda^\top F_{t+1})^2 \approx \mathbb{E}\left[ (1 - \lambda^\top F_{t+1})^2 \right] &= 1 - 2\mathbb{E}[\lambda^\top F_{t+1}] + \mathbb{E}[(\lambda^\top F_{t+1})^2] \\ &= 1 - 2\mathbb{E}[U(\lambda^\top F_{t+1})] \end{aligned}$$

where

$$U(x) = x - \frac{x^2}{2}$$

## Transformer Architecture

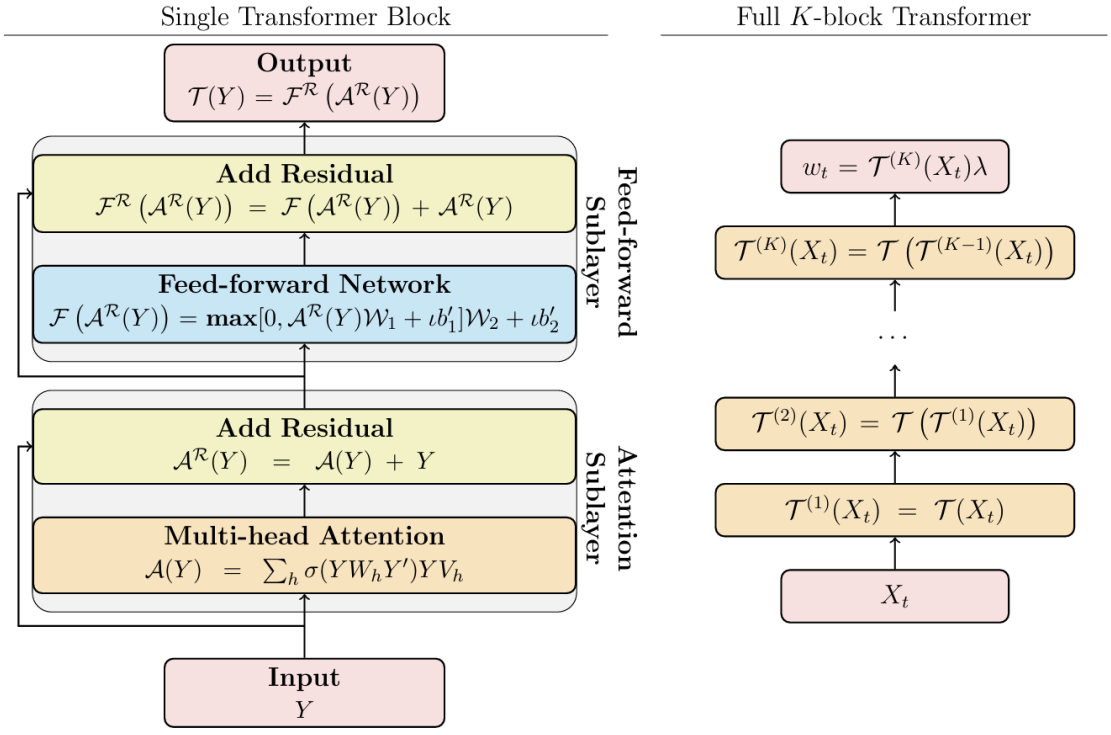



Figure 6: Illustration of the Transformer Architecture (as seen in [1]).

## Transformer Nonlinearities

There are multiple nonlinearities in the transformer's architecture. First, we encounter the softmax function, $\sigma$. While the general attention mechanism gathers information about asset $i$ from the characteristics of all other assets, the softmax operation selectively focuses attention on fewer related assets, ignoring the rest. Secondly, we have the feed-forward neural network, $\mathcal{F}$. This network operates row-wise, thus transforming the characteristics of each asset $i$ independently of the other assets. It does not contribute to the transformer's cross-asset information sharing

but it introduces a large number of new additional parameters. Lastly, the residual connections are introduced to stabilize the optimization by helping to counteract vanishing and exploding gradients during training. [1]

## Plots

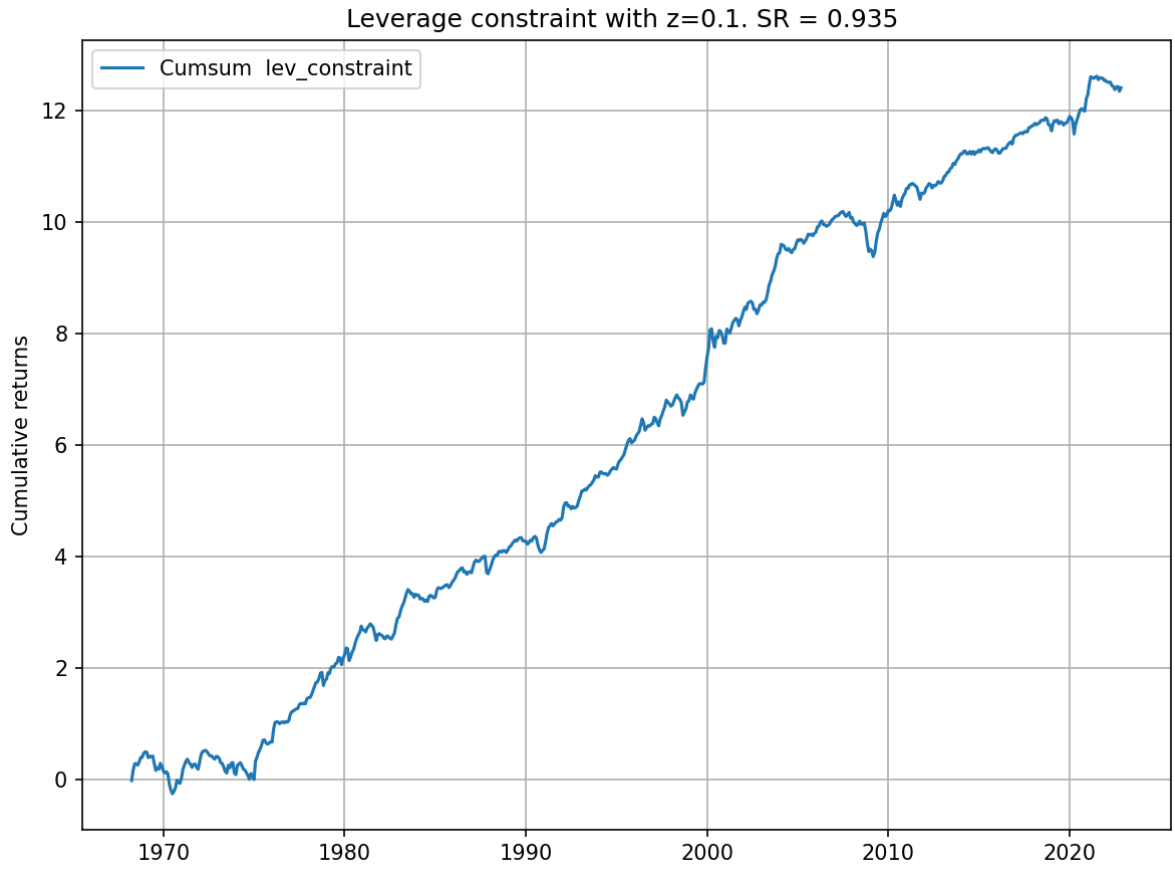



Figure 7: Cumulative return of the transformer model with maximum 1.5 leverage.

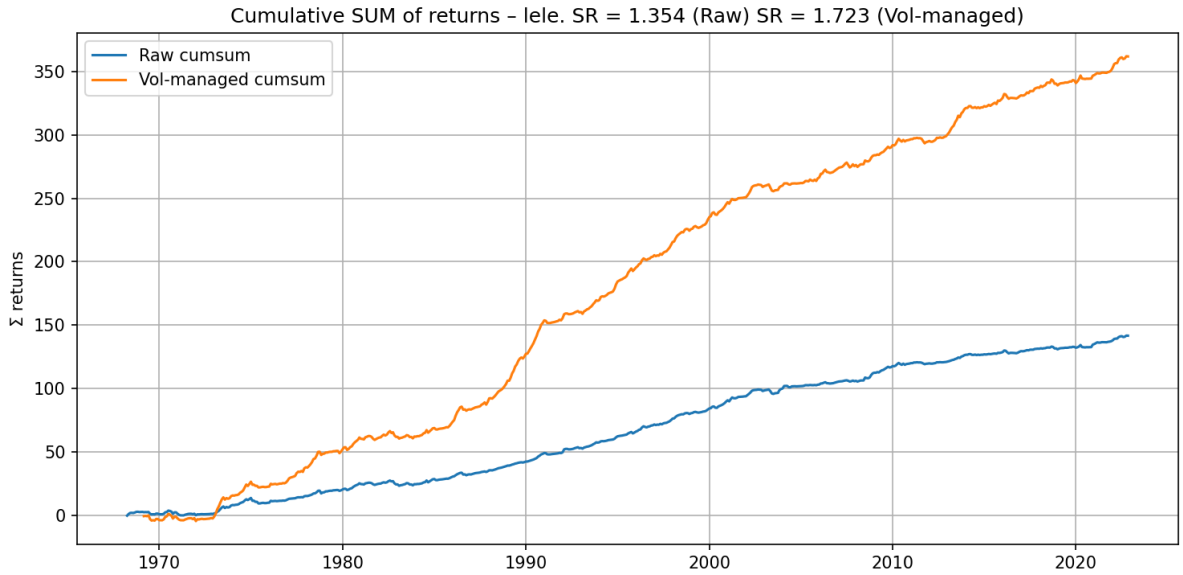



Figure 8: Cumulative return micro stocks raw and with volatility managed

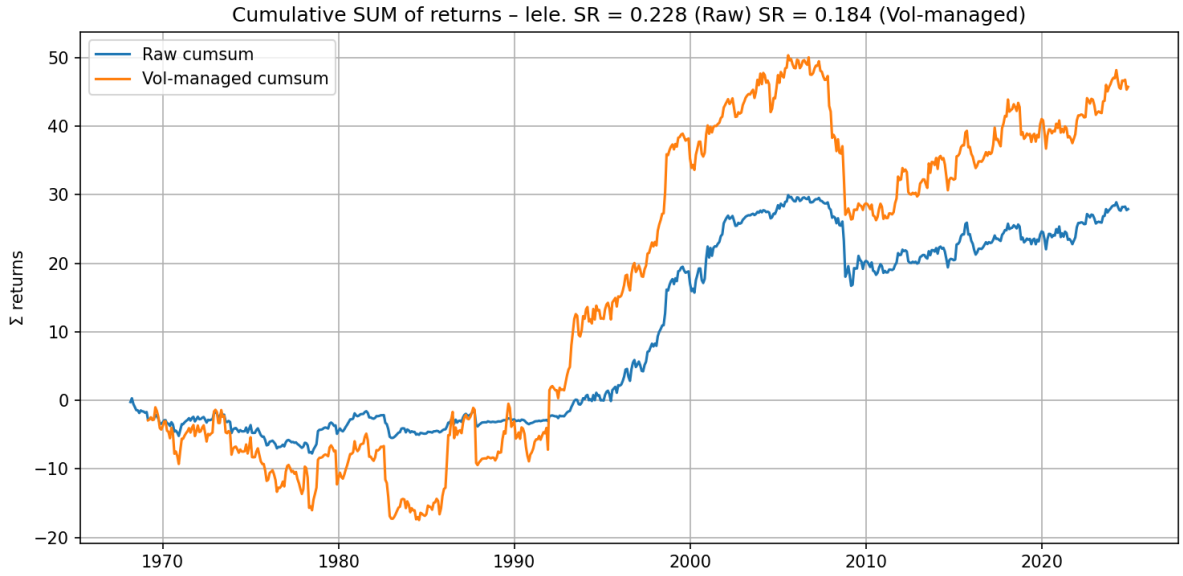



Figure 9: Cumulative return large stocks raw and with volatility managed