NBER WORKING PAPER SERIES

# ARTIFICIAL INTELLIGENCE ASSET PRICING MODELS

Bryan T. Kelly
Boris Kuznetsov
Semyon Malamud
Teng Andrea Xu

Working Paper 33351
http://www.nber.org/papers/w33351

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
January 2025

Semyon Malamud gratefully acknowledges the financial support of the Swiss Finance Institute and the Swiss National Science Foundation, Grant 100018 192692. AQR Capital Management is a global investment management firm that may or may not apply similar investment techniques or methods of analysis as described herein. The views expressed here are those of the authors and not necessarily those of AQR. This work was supported by a Swiss National Supercomputing Centre (CSCS) grant under project ID sm86. The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research.

NBER working papers are circulated for discussion and comment purposes. They have not been peer-reviewed or been subject to the review by the NBER Board of Directors that accompanies official NBER publications.

© 2025 by Bryan T. Kelly, Boris Kuznetsov, Semyon Malamud, and Teng Andrea Xu. All rights reserved. Short sections of text, not to exceed two paragraphs, may be quoted without explicit permission provided that full credit, including © notice, is given to the source.

Artificial Intelligence Asset Pricing Models
Bryan T. Kelly, Boris Kuznetsov, Semyon Malamud, and Teng Andrea Xu
NBER Working Paper No. 33351
January 2025
JEL No. C45, G10, G11, G14, G17

## ABSTRACT

The core statistical technology in artificial intelligence is the large-scale transformer network. We propose a new asset pricing model that implants a transformer in the stochastic discount factor. This structure leverages conditional pricing information via cross-asset information sharing and nonlinearity. We also develop a linear transformer that serves as a simplified surrogate from which we derive an intuitive decomposition of the transformer's asset pricing mechanisms. We find large reductions in pricing errors from our artificial intelligence pricing model (AIPM) relative to previous machine learning models and dissect the sources of these gains.

Bryan T. Kelly
Yale School of Management
165 Whitney Ave.
New Haven, CT 06511
and NBER
bryan.kelly@yale.edu

Boris Kuznetsov
Swiss Finance Institute @ EPFL
Quartier UNIL-Dorigny, Extranef 244
CH-1015 Lausanne
Switzerland
boris.kuznetsov@epfl.ch

Semyon Malamud
Swiss Finance Institute @ EPFL
Quartier UNIL-Dorigny, Extranef 213
CH - 1015 Lausanne
Switzerland
semyon.malamud@epfl.ch

Teng Andrea Xu
UNIL Dorigny, Extranef
Switzerland
andrea.xu@epfl.ch

## 1 Introduction

The large language model (LLM) revolution emerged from two complementary modeling insights.[1] First, words need to be understood in context. Language models that study words without awareness of the surrounding text context cannot faithfully represent a text's meaning and, therefore, suffer in tasks like word prediction and translation. However, capturing contextual relationships among words is an extraordinarily high-dimensional problem, and early modeling attempts have failed to find a tractable solution. The so-called "transformer" model is a watershed discovery that makes it possible to model words in context while maintaining computational tractability flexibly.

The second insight is that unprecedented out-of-sample performance can be achieved through exorbitant model parameterizations, or "complexity." This surprising empirical phenomenon has prompted a search for theoretical foundations of the virtue of complexity.[2] The computational tractability of transformers is the first step toward implementing large parameterizations. Still, more recent LLMs such as GPT-4 push the limits of model complexity further by adopting "scaling" components that proliferate parameterization.

These insights that underpin LLM success have also spurred a broader AI revolution, propelling gains in computer vision, computational biology, and other AI applications by efficiently leveraging contextual information in large-scale models.

The finance literature has focused primarily on small-scale "own-asset prediction" models—those in which forecasts for asset $i$ depend on conditioning variables that are specific to asset $i$. The recent AI experience suggests that own-asset predictability may be the most limiting maintained restriction in empirical asset pricing. Like a bag-of-words language model, own-asset prediction models abstract from context by failing to use information about the broader universe of assets to describe the risk and return of each individual asset. The typically small parameterization in finance models tends to mask the full extent of predictability concealed in finance data.[3]

This paper introduces the concept of an artificial intelligence pricing model (AIPM).

[1]See, e.g., Minaee et al. (2024) for an extensive LLM review.

[2]See, Spigler et al. (2019), Belkin et al. (2018), Belkin et al. (2019), Belkin et al. (2020), and Bartlett et al. (2020), as well as Kelly et al. (2024a) and Didisheim et al. (2024).

[3]This shortcoming has been analyzed and documented in Kelly et al. (2024a) and Didisheim et al. (2024).

We define an AIPM as a model that implants in the stochastic discount factor (SDF) the same context-aware and large-scale transformer architecture from state-of-the-art LLMs and other AI. The critical attributes of an AIPM are i) that it flexibly shares conditioning information across assets to arrive at context-aware forecasts and ii) that out-of-sample model performance is maximized by pushing the limits of model parameterization. Thus, our first main contribution is an asset pricing model design that leverages the two aforementioned attributes of leading AI models.

The transformer can seem analytically impenetrable. The second contribution of our paper is to derive an intuitive characterization of the transformer's role in the context of asset pricing. To do so, we begin with a simple and interpretable variant of our model that we refer to as the "linear portfolio transformer." This SDF model is easy to describe and inspect and admits a closed-form regression-based estimator. It uses the same core information sharing apparatus—the "attention" mechanism—as a standard transformer, but removes the other nonlinearities present in the full-blown nonlinear transformer. Most importantly, the analytical tractability of the linear portfolio transformer makes it a powerful surrogate for understanding the impact of cross-asset attention on the SDF portfolio. It has a number of other surprising model properties that we derive theoretically, including an ability to vary model parameterization without changing the data inputs or deviating from linearity. This property, which we refer to as "linear multi-headed attention," is convenient for inspecting the virtue of complexity in the linear specification.

Our analytical derivations provide interpretations of the attention-based SDF. The first is related to the representation of a conditional SDF as a linear combination of characteristic-managed "factor" portfolios (e.g. Kozak et al., 2020a). For example, in the value factor portfolio, stock $i$'s weight in the factor is proportional to its book-to-market ratio. This construction is based on the view that a stock's book-to-market ratio proxies its expected return. However, if this stock characteristic is a noisy measure of expected return and if there is sufficient dependence among the true expected returns of stocks, then the noisy factor can be improved by replacing the book-to-market characteristic with a more refined measure that draws upon information in the book-to-market ratios of other stocks. This cross-sectional information sharing is our asset pricing model's analog to context awareness for words in an LLM.

The linear portfolio transformer also illustrates that AIPMs can be interpreted as performing a sophisticated version of factor timing. Factor timing models have been proposed by Gupta and Kelly (2019), Haddad et al. (2020), and Ehsani and Linnainmaa (2022), among others. Relatedly, the principal portfolios method of Kelly et al. (2023) explicitly links factor timing to the problem of cross-asset return prediction. These papers consider one factor at a time and are developed in low-complexity settings. In contrast, our portfolio transformers are high-dimensional timing models that optimize the timing of all factors jointly.

The linear portfolio transformer is an intermediate modeling step. While it is a valuable device for understanding the role of context-awareness in an AIPM, it abstracts from a variety of other model features known to be critical to the success of modern AI models. And though the linear transformer has high complexity by typical asset pricing standards, there remains significant scope for expanding model parameterization further. We, therefore, introduce an AIPM that uses a large nonlinear transformer architecture inspired by Vaswani et al. (2017). This includes multi-headed attention, softmax transformations, a feed-forward network, residual connections, and, most importantly, the deep learning benefit of stacking multiple transformer blocks together. We discuss the intuitive role of each of these AIPM model components.

Our third main contribution is to document and dissect the empirical performance of AIPMs. We focus on monthly data for US stock returns and use a standard set of 132 stock-level conditioning characteristics from Jensen et al. (2023) (JKP henceforth). As appropriate for machine learning model comparisons, we focus on out-of-sample Sharpe ratios and pricing errors for anomaly portfolios following the formulation of Didisheim et al. (2024) (DKKM henceforth).

We begin with an empirical analysis of the linear portfolio transformer. We document the benefits of cross-asset information sharing by comparing it to a linear SDF in which the attention mechanism is shut down. The benchmark establishes that a standard linear SDF with no cross-asset information achieves an out-of-sample Sharpe ratio of 3.6. The linear portfolio transformer, which introduces the attention mechanism but no other model changes (i.e., maintaining linearity and the same data inputs), produces a Sharpe ratio improvement to 3.9 and an alpha $t$-statistic of 6.8 to the attention-free linear model, suggesting that the benefits of cross-asset information sharing are non-trivial even in a simple, linear setting. Next, we use the multi-head capability of the linear portfolio transformer to investigate the benefits of large-scale parameterizations. We find that the out-of-sample Sharpe ratio is strictly increasing in the number of model parameters, establishing the virtue of complexity (Kelly et al., 2024b, DKKM) in the new modeling context of the linear portfolio transformer.

Finally, we analyze the empirical properties of the nonlinear portfolio transformer. To evaluate the role of information sharing in the nonlinear setting, we require a benchmark model that incorporates nonlinearities but does so solely through its use of own-asset information. The recently proposed SDF of DKKM, which is nonlinear but lacks an attention mechanism, is well-suited for this comparison. In our analysis, DKKM achieves an out-of-sample Sharpe ratio of 3.9. In contrast, through its information-sharing capacity, the nonlinear portfolio transformer achieves a large and significant Sharpe ratio improvement to 4.6. We also design an otherwise "apples-to-apples" feedforward network with the same architecture as the nonlinear portfolio transformer but excluding the attention mechanism to shut down cross-asset information sharing. In this case, we find the same qualitative result that we found when comparing to DKKM. In other words, the excess performance of the transformer is indeed attributable to the information-sharing mechanism.

Next, we investigate how nonlinear transformer performance responds as we scale up parameterization by concatenating more transformer blocks to increase network depth. We find a clear virtue of complexity, with the Sharpe ratio increasing from 3.8 for a single-block transformer (a model with about 100,000 parameters) to 4.6 for a ten-block transformer (a roughly one million parameter model). Our analysis suggests that our nonlinear portfolio transformer can be improved further by increasing parameterization as the slope of out-of-sample performance to the number of blocks has not yet flattened (though due to computation costs for model training, we leave further parameter expansion to subsequent work). A key synthesis of our findings is a pecking order among models. Those with the simplest specifications are the weakest performers, and performance gradually increases as we move towards models with richer specifications, and those with an attention mechanism for cross-asset information sharing dominate those without.

This paper contributes to the emerging literature demonstrating that machine learning asset pricing models achieve higher out-of-sample Sharpe ratios and smaller pricing errors than their more parsimonious predecessors. Notable examples include Kozak et al. (2020b), Gu et al. (2020a), Chen et al. (2023), Bryzgalova et al. (2020), Cong et al. (2022), Fan et al. (2022), and Preite et al. (2022), among others. Our work extends this literature by introducing transformer architectures into asset pricing models, leveraging cross-asset information sharing, and embracing model complexity to capture intricate relationships among assets.

In relation to machine learning methods for analyzing factor models, our approach aligns with efforts by Connor et al. (2012), Fan et al. (2016), Kelly et al. (2020a), Lettau and Pelger (2020), Giglio and Xiu (2021), Giglio et al. (2022), and He et al. (2023), who employ advanced statistical techniques to improve factor model estimation and prediction (see Kelly and Xiu, 2023; Rapach and Zhou, 2020, for comprehensive surveys on these topics). By embedding transformer architectures within the stochastic discount factor (SDF), we push the boundaries of this literature, demonstrating how attention mechanisms can effectively capture cross-sectional dependencies and enhance predictive performance.

By utilizing transformers and their attention mechanisms, we provide evidence that the factor zoo (Harvey et al., 2016; McLean and Pontiff, 2016; Hou et al., 2020; Feng et al., 2020; Jensen et al., 2023; Chen and Zimmermann, 2021) is important for capturing subtle dimensions of cross-asset information flows. Effectively, transformers embed stocks into the space of high-dimensional characteristics and then use this embedding to learn characteristics-based networks for information transmission.[4] The demonstrated success of machine learning models in predicting the cross-section of returns, as shown in Chinco et al. (2019), Han et al. (2019), Freyberger et al. (2020), Rapach and Zhou (2020), Gu et al. (2020b), Avramov et al. (2023), and Guijarro-Ordonez et al. (2021), provides additional evidence of the benefits of embracing model complexity, a central theme in our application of transformers to asset pricing.

The paper proceeds as follows. In Section 2, we introduce the linear portfolio transformer

[4] **?** estimate a transformer model that embeds holdings data in a low-dimensional space that captures demand-related asset and investor characteristics.

that includes cross-asset attention in an otherwise vanilla conditional SDF, derive its theoretical properties, and present a closed-form regression-based estimator. Section 3 introduces a large-scale AIPM that embeds a deep nonlinear transformer in the SDF. Lastly, in Section 4, we evaluate the empirical properties of these models in monthly US stock data.

## 2 The (Interpretable) Linear Portfolio Transformer

In this section, we introduce the linear portfolio transformer. The linear model is estimable in closed form, providing a particularly transparent environment for dissecting its functionality in intuitive terms. Most conceptual insights drawn from our linear transformer carry over to its nonlinear counterpart in Section 3, which uses similar model components and is based on popular LLM transformers.

### 2.1 Model Specification

Our setting consists of a universe of $N_t$ risky assets in the economy at time $t$ whose excess returns are given by the vector $R_{t+1}$. Financial machine learning strives to incorporate large conditioning sets in richly parameterized models with the aim of accurately characterizing the expected out-of-sample behavior of asset prices. To this end, each asset is accompanied by a high-dimensional vector of $D$ characteristics $X_{i,t}$. We use $X_t$ to denote the $N_t \times D$ matrix stacking characteristics for all stocks at time $t$. We assume that the conditioning variables in $X_t$ span the time $t$ information set. Our framework accommodates a time-varying universe of $N_t$ assets by conditioning assets' behavior on a fixed number $D$ of asset-level characteristics, as in Kelly et al. (2020b).

Kelly and Xiu (2023) discuss the conveniences of representing machine learning asset pricing models in terms of their SDF or, equivalently, in terms of the maximum Sharpe ratio portfolio of risky assets. We follow this approach in defining an AIPM whose conditional SDF representation (Hansen and Richard, 1987) is

$$W_{t+1} = 1 - w(X_t)' R_{t+1} \tag{1}$$

where the vector function $w(X_t)$ maps conditioning information into the SDF's conditional weights on individual risky assets, where $w(X_t)'R_{t+1}$ is the conditionally mean-variance optimal portfolio.

The basic linear portfolio transformer model is:

$$w_t = A_t X_t \lambda = (X_t W X_t') X_t \lambda \,, \tag{2}$$

where $W$ ($D \times D$) and $\lambda$ ($D \times 1$) are the model parameters to be estimated. We refer to the matrix $A_t = X_t W X_t'$ as the "attention" matrix. The following subsections describe and interpret the model's functionality.

## 2.2 Transformers and Cross-asset Information Sharing

At the core of the portfolio transformer architecture is the so-called attention mechanism, introduced to the machine learning literature in various forms by Vaswani et al. (2017), Cho (2014), and Bahdanau (2014), and most prominently known for its success in large language models like ChatGPT. In the AIPMs we develop, the attention mechanism provides a flexible and tractable framework for introducing cross-asset information sharing.

The literature on financial machine learning has focused primarily on own-asset prediction in which the optimal portfolio weight of asset $i$ depends solely on conditioning variables that are specific to asset $i$.[5] In an early formulation of this approach, Brandt et al. (2009) specify the optimal portfolio weight for asset $i$ as a linear function of its characteristics, $w_{i,t} = X_{i,t}'\lambda$. This is a special case of (2) in which the matrix $A_t$ is the identity matrix:

$$w_t = X_t \lambda. \tag{3}$$

Likewise, the high complexity analysis of DKKM centers on the formulation $w_{i,t} = f(X_{i,t})\lambda$, where $f$ is a heavily parameterized neural network. All assets share the same network architecture and model parameters, but each asset's portfolio weight continues to depend, ultimately, only on its own characteristics.

[5]An exception is Cong et al. (2021), though they do not distinguish the role of attention from the kitchen sink of other machine learning architectures included in their model.

As argued in the introduction, own-asset predictability is a potentially costly restriction in empirical asset pricing models. However, loosening this restriction can be a difficult task. A simple formulation of cross-asset predictability would be a static version of (2) with $A_t = A$:[6]

$$w_t = AX_t\lambda. \tag{4}$$

The $d^{th}$ characteristic of stock $i$ is thus modified from $X_{i,d,t}$ to $\tilde{X}_{i,d,t}$ by multiplying the $d^{th}$ column of $X_t$ with the $i^{th}$ row of $A$. In other words, a cross-prediction formulation like (4) uses information sharing to enhance an own-asset predictive characteristic with a weighted average of that same characteristic's value across all assets. In other words, the matrix $A$ summarizes the "attention" one should pay to characteristics of all other assets when deciding on the ideal portfolio weight for asset $i$. The optimal attention matrix $A$ can be learned as part of the training objective (9) along with the weights $\lambda$ that optimally map the "cross-sectionally smoothed" characteristics into final portfolio weights.

However, the formulation of (4) is unattractive for a few reasons. For one, a static $A$ matrix cannot handle time-varying universes of $N_t$ assets, as is necessary, for example, in the cross-section of individual stocks. More importantly, it ignores potentially valuable information about the similarity among stocks in the conditioning set $X_t$. We can remedy these shortcomings with a dynamic attention matrix $A_t$ that exploits conditional information sharing. But the question then is how to operationalize conditional attention. Fortunately, the attention mechanism within a transformer is well suited for this problem. It parameterizes $A_t$ as a function of observables in $X_t$. In particular, the conditional attention mechanism in (2) shares information based on the *conditional similarity* between the characteristics of asset $i$ and asset $j$,

$$A_t = X_t W X_t'. \tag{5}$$

In this formulation, information is more actively shared among assets with similar attributes.

[6]This is related to the cross-asset prediction technique of principal portfolios proposed by Kelly et al. (2023) and their multivariate extension by He et al. (2022).

In fact, if the rows of $X_t$ are variance-standardized, and $W$ is the identity matrix, then $(i,j)^{th}$ element of $A_t$ reports the correlation across characteristics for asset pair $(i,j)$. If $W$ is not identity but is instead a $D \times D$ matrix of free parameters, then the model has the flexibility to learn which conditioning characteristics represent the most fruitful pathways of cross-asset prediction.[7]

Through algebraic manipulation of (2), we see

$$\begin{aligned} w_t &= (X_t W X_t')X_t\lambda \\ &= X_t(\lambda' \otimes W)\text{vec}(X_t'X_t) \\ &= \big(\text{vec}(X_t'X_t)' \otimes X_t\big)\text{vec}\big(\lambda' \otimes W\big) \\ &= \breve{X}_t\breve{\lambda}. \end{aligned} \tag{6}$$

The portfolio weights in (6) are linear in $\breve{X}_t = \text{vec}(X_t'X_t)' \otimes X_t$, which is an $N_t \times D^3$ matrix of three-way interactions among characteristics of all assets, and with up to $D^3$ distinct linear parameters $\breve{\lambda}$. This brings the linear portfolio transformer full circle back to the simple SDF model of (3). The linearity of portfolio weights in an extremely large number of nonlinear functions of $X_t$ is reminiscent of the high complexity SDF formulation of DKKM. But unlike DKKM, the linear transformer achieves this complexity by exploiting cross-asset predictability in the form of cross-asset characteristic interactions.

## 2.3 Transformers and Factor Timing

It is common in the literature to use the set of $D$ characteristics in $X_t$ to construct characteristic-managed portfolios, or "factors," whose returns are defined by the $D \times 1$ vector $F_{t+1} = X_t'R_{t+1}$. A basic linear portfolio specification, $w_t = X_t\lambda$ as in (3) can thus be viewed as a portfolio of factors:

$$w_t'R_{t+1} = F_{t+1}'\lambda. \tag{7}$$

[7]The information entering the attention function $A_t$ can differ from the core information that maps into portfolio weights in (4) without loss of generality. For example, we may wish to consider $w_t = (Y_t W Y_t')X_t\lambda$ for some $Y_t$ with first dimension $N_t$ and $Y_t \neq X_t$. For simplicity of exposition, we work from a single matrix of conditioning variables $X_t$ throughout.

Recent literature on "factor timing" has documented robust evidence of time series predictability in anomaly factor returns.[8] The literature also documents portfolio gains from strategies that form dynamic combinations of factors to exploit time variation in factor expected returns. One example of this approach is factor momentum, which builds a portfolio that combines factors in proportion to their recent average returns.

In light of this, equations (6) and (7) together give another perspective on how the portfolio transformer approaches cross-asset prediction. It introduces an enormous number of ways to time each factor in $F_t$ using each element of $X_t'X_t$, as captured by the term $\check{X}_t = \text{vec}(X_t'X_t) \otimes X_t$. The trained portfolio transformer coefficient vector $\text{vec}\big(\lambda' \otimes W\big)$ selects the best combination of these factor timing strategies.

Kelly et al. (2023) develop an asset pricing theory for simple timing models and establish a rigorous connection with the problem of cross-asset return prediction. Their analysis deals with only one factor at a time and in low-complexity settings. In contrast, the portfolio transformer pushes cross-asset information sharing to extreme levels of complexity and conditionality while jointly optimizing the timing of all factors.

## 2.4 Multiple Heads

In the machine learning literature, the structure in (5) is referred to as attention "head." Transformers typically possess multiple attention heads. In analogy, we define the $H$-head linear portfolio transformer as

$$\begin{aligned} w_t &= \underbrace{(X_t W_1 X_t')X_t\lambda_1}_{head\ \#1} + \cdots + \underbrace{(X_t W_H X_t')X_t\lambda_H}_{head\ \#H} \\ &= \Big(\text{vec}(X_t'X_t) \otimes X_t\Big)\text{vec}\left(\sum_{h=1}^{H} \lambda_h' \otimes W_h\right) = \check{X}_t\check{\lambda}. \end{aligned} \tag{8}$$

Why incorporate a variety of heads in the portfolio transformer? Equation (8) shows that multi-head attention allows for multiple pathways of cross-asset predictability. Increasing the number of heads increases the model's parameterization and thus its flexibility without

[8] See Gupta and Kelly (2019), Haddad et al. (2020), Kelly et al. (2023), and Ehsani and Linnainmaa (2022).

changing the regressors, $\check{X}_t$, which are the same regardless of the number of heads. Relatedly, we can think of the multi-head formulation as performing model averaging—(8) is an ensemble of $H$ different single-head models for the optimal portfolio.[9]

## 2.5 Estimation and Identification

A natural objective function for training an AIPM with SDF weight $w\left(X_t;\Theta_{\mathcal{T}}\right)$ is:

$$\min_{\Theta_{\mathcal{T}}} E_t\left[\left(1 - w\left(X_t;\Theta_{\mathcal{T}}\right)' R_{t+1}\right)^2\right] + g(\Theta_{\mathcal{T}};z), \tag{9}$$

where $w\left(X_t;\Theta_{\mathcal{T}}\right)$ is the SDF weight function with parameters $\Theta_{\mathcal{T}}$, and $g(\Theta_{\mathcal{T}};z)$ is a shrinkage penalty term with a shrinkage parameter $z$. This objective is built around the equivalence between an SDF and the mean-variance efficient portfolio. Kelly and Xiu (2023) refer to this penalized least squares problem as maximum Sharpe ratio regression (MSRR) since its solution is the (penalized) conditionally efficient portfolio. DKKM prove that the estimator in (9) approximates to the conditionally mean-variance efficient portfolio, and they derive technical conditions under which this solution consistently recovers the true optimal portfolio.[10] Similarly, (9) maps to the standard asset pricing Euler condition that the true SDF $W_{t+1}$ conditionally prices all assets with zero error:

$$E_t[W_{t+1}R_{t+1}] = 0, \quad \text{where} \quad W_{t+1} = 1 - w\left(X_t;\Theta_{\mathcal{T}}\right)' R_{t+1}. \tag{10}$$

When $g(\Theta_{\mathcal{T}};z) = 0$, the Euler equation (10) is the first-order condition of the optimization problem in (9). It is also the vector of SDF pricing errors for all risky assets. In other words, the objective (9) can be viewed as a portfolio optimization problem or, equivalently, as a pricing error minimization problem.

The estimation problem in (9) is presented in general terms and also accommodates the nonlinear specification introduced in Section 3. In the case of the linear portfolio transformer

[9]One technical caveat regarding multi-head attention is that the parameters in this formulation are not uniquely identified. We discuss our approach to identification in the following section.

[10]Britten-Jones (1999) establishes the link between an unconditional version (9) and the unconditional maximum Sharpe ratio portfolio.

with a ridge penalty function, the objective is to

$$\min_{\check{\lambda}}\{E_t\left[\left(1-\check{\lambda}'\check{X}_t'R_{t+1}\right)^2\right]+z\check{\lambda}'\check{\lambda}\}\,, \tag{11}$$

which delivers a convenient closed-form expression for linear portfolio transformer estimator:

$$\hat{\check{\lambda}}=\left(\sum_t \check{X}_t'R_{t+1}R_{t+1}'\check{X}_t+zI\right)^{-1}\left(\sum_t \check{X}_t'R_{t+1}\right). \tag{12}$$

This formula begins to shed light on the issue of identification in the multi-head transformer. The number of unique parameters in an $H$-head model is $P=H(D^2+D)$ as there are $D^2$ parameters for each $W_h$ and $D$ parameters for each $\lambda_h$. The dimension of the regression parameter $\check{\lambda}$ in (8) is $D^3$. If the number of heads is large enough that $H(D^2+D)\geq D^3$, MSRR can recover up to $D^3$ unique parameters. We refer to a model with $D^3$ unique parameters as "saturated" because it is the richest unique parameterization that can be achieved with a linear transformer using $D$ conditioning variables.

If the number of heads is small enough that $H(D^2+D)<D^3$, then there are cross-parameter restrictions on the elements of $\check{\lambda}$ and the estimator in (12) must be modified to impose these constraints. In Appendix B, we provide an algorithm for recovering a unique set of $H(D^2+D)<D^3$ restricted parameters, which amounts to a form of constrained least squares. By varying the number of attention heads, we can investigate the role of model complexity on the performance of the linear transformer.

## 3 The Nonlinear Portfolio Transformer

The linear portfolio transformer offers a window into the role of attention for an AIPM. However, it abstracts from a variety of nonlinearities typically employed in the transformers that underly LLMs and other leading AI models. In this section, we develop a deep nonlinear portfolio transformer. Its architecture is an asset pricing adaptation of such well-known transformer models as Vaswani et al. (2017).

First, we will give a complete statement of the nonlinear portfolio transformer. Then, we discuss the intuition behind its various nonlinearities and how they aid model performance.

## 3.1 Model Specification

The nonlinear portfolio weight function is a $K$-block transformer architecture. Each block is composed of two sublayers. The first sublayer applies a multi-head attention unit defined as

$$\mathcal{A}(Y) = \sum_{h=1}^{H} \sigma\left(Y W_h Y'\right) Y V_h \tag{13}$$

to the generic $N_t \times D$ matrix input $Y$. $W_h$ and $V_h$ are $D \times D$ matrices of parameters. The function $\sigma : \mathbb{R}^{N_t \times N_t} \rightarrow \mathbb{R}^{N_t \times N_t}$ is a row-wise softmax operator applied to the $N_t \times N_t$ matrix $Y W_h Y'$.

After the softmax operation, the original $Y$ is added back in a so-called "residual connection:"

$$\mathcal{A}^{\mathcal{R}}(Y) = \mathcal{A}(Y) + Y. \tag{14}$$

The second sublayer is a fully connected feed-forward network with one hidden layer of $d_{\mathcal{F}}$ neurons:

$$\mathcal{F}(Y) = \max\left[0, Y\mathcal{W}_1 + \iota b_1'\right] \mathcal{W}_2 + \iota b_2', \tag{15}$$

where the parameter matrix $\mathcal{W}_1$ is $D \times d_f$, $b_1$ is $d_f \times 1$, $\mathcal{W}_2$ is $d_f \times D$, $b_2$ is $D \times 1$, and $\iota$'s are conforming vectors of ones.[11] The output of the feed-forward step is, therefore, the same dimension as the input, $N_t \times D$. Again, we include a residual connection after the feed-forward network:

$$\mathcal{F}^{\mathcal{R}}(Y) \;=\; \mathcal{F}(Y) + Y\,. \tag{16}$$

[11]The max operator in (15) is applied row-wise. More specifically, the network can be understood as operating on asset-by-asset observations, so it takes the $D \times 1$ input $y_i$ (the $i^{th}$ row of $Y$) and produces a $D \times 1$ output. These outputs are then stacked into the $N_t \times D$ matrix $\mathcal{F}(Y)$.

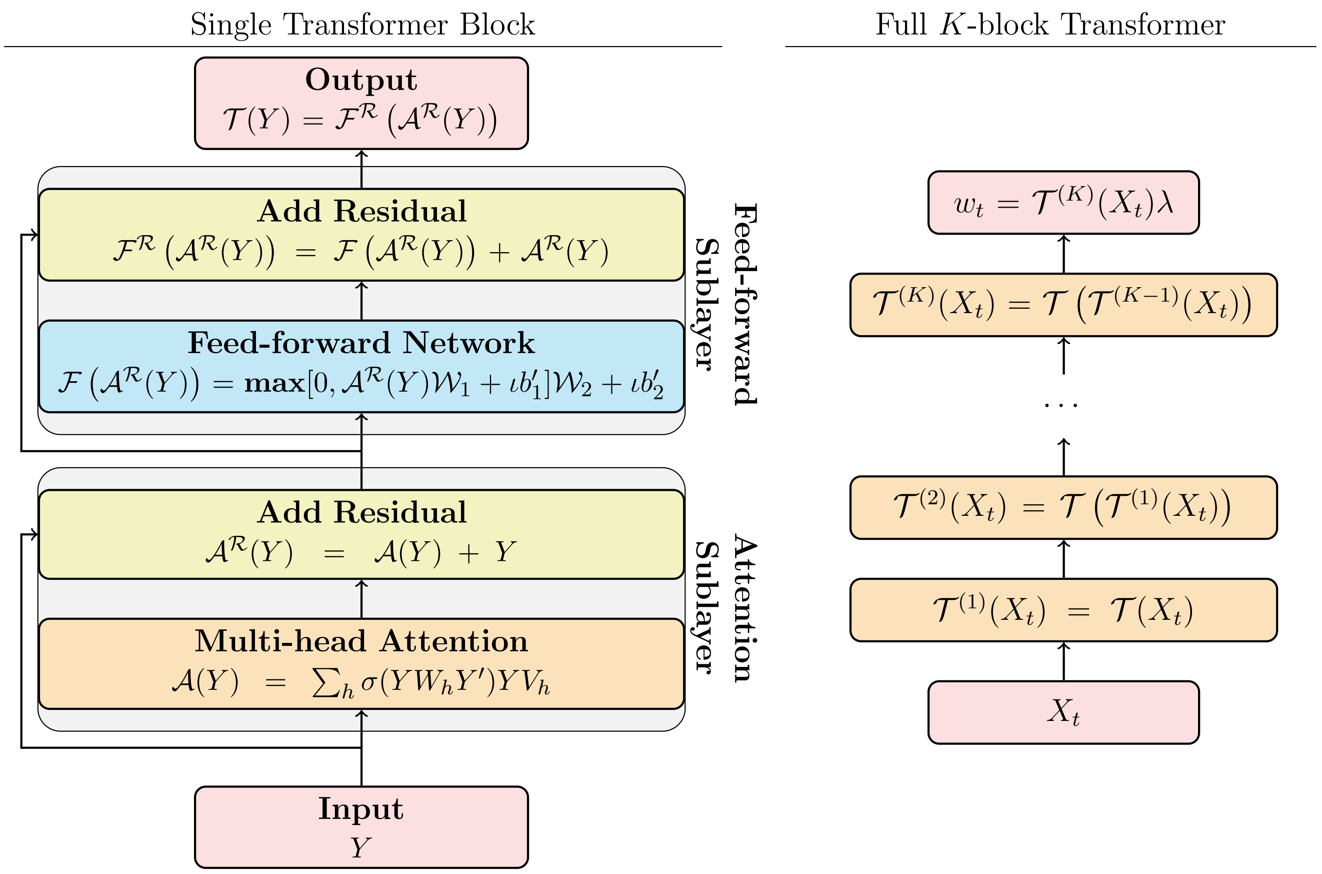

Single Transformer Block
Output
$\mathcal{T}(Y) = \mathcal{F}^{\mathcal{R}}(\mathcal{A}^{\mathcal{R}}(Y))$
Add Residual
$\mathcal{F}^{\mathcal{R}}(\mathcal{A}^{\mathcal{R}}(Y)) = \mathcal{F}(\mathcal{A}^{\mathcal{R}}(Y)) + \mathcal{A}^{\mathcal{R}}(Y)$
Feed-forward Network
$\mathcal{F}(\mathcal{A}^{\mathcal{R}}(Y)) = \mathbf{max}[0, \mathcal{A}^{\mathcal{R}}(Y)\mathcal{W}_1 + \iota b_1']\mathcal{W}_2 + \iota b_2'$
Feed-forward Sublayer
Add Residual
$\mathcal{A}^{\mathcal{R}}(Y) = \mathcal{A}(Y) + Y$
Multi-head Attention
$\mathcal{A}(Y) = \sum_h \sigma(YW_hY')YV_h$
Attention Sublayer
Input
$Y$
Full $K$-block Transformer
$w_t = \mathcal{T}^{(K)}(X_t)\lambda$
$\mathcal{T}^{(K)}(X_t) = \mathcal{T}(\mathcal{T}^{(K-1)}(X_t))$
$\cdots$
$\mathcal{T}^{(2)}(X_t) = \mathcal{T}(\mathcal{T}^{(1)}(X_t))$
$\mathcal{T}^{(1)}(X_t) = \mathcal{T}(X_t)$
$X_t$

Figure 1: **Illustration of Nonlinear Portfolio Transformer**

The right figure shows the full architecture of a $K$-block portfolio transformer. The left figure shows the structural details within a transformer block.

A complete transformer block $\mathcal{T}$ is a composition of these two sublayers:

$$\mathcal{T}(Y) = \mathcal{F}^{\mathcal{R}}\big(\mathcal{A}^{\mathcal{R}}(Y)\big). \tag{17}$$

Given the transformer block in (17), we can state the full recursive definition of the portfolio transformer. The recursion is initialized with an identity layer whose input and output are the raw $N_t \times D$ conditioning matrix $X_t$:

$$\mathcal{T}^{(0)}(X_t) = X_t. \tag{18}$$

The input to each additional transformer block $k = 1, ..., K$ is the output from the previous block:

$$\mathcal{T}^{(k)}(X_t) \;=\; \mathcal{T}\left(\mathcal{T}^{(k-1)}(X_t)\right). \tag{19}$$

The recursion culminates in an $N_t \times D$ output matrix of reconstituted asset characteristics, $\mathcal{T}^{(K)}(X_t)$. The final linear layer maps these characteristics into the conditional SDF portfolio weights with parameter a final $D \times 1$ parameter vector $\lambda$:

$$w_t = \mathcal{T}^{(K)}(X_t)\lambda. \tag{20}$$

Figure 1 illustrates the tortuous propagation of input data $Y$ through the network. Transformer block $k$ is equipped with block-specific parameters,

$$\theta^{(k)} = \left((W_h^{(k)}, V_h^{(k)})_{h=1}^{H}, \mathcal{W}_1^{(k)}, b_1^{(k)}, \mathcal{W}_2^{(k)}, b_2^{(k)}\right),\; k = 1, \cdots, K\,,$$

and the complete set of AIPM parameters is then given by

$$\Theta_{\mathcal{T}} = \left\{(\theta^{(k)})_{k=1}^{K}, \lambda\right\}, \tag{21}$$

for a total of $K(2D^2H + 2Dd_f + d_f + D) + D$ parameters. In the subsequent analysis, we use $d_f = 256$ and $H = 1$. [12]

## 3.2 The Role of Nonlinearities

DKKM establish the surprising result that the out-of-sample performance of an SDF improves with the number of parameters. This is a counterpart to the virtue of complexity Kelly et al. (2024b) (KMZ henceforth) in the context of asset pricing models. Heavy parameterizations of AI models enhance their capacity to approximate unknown data-generating processes. This phenomenon is well known in machine learning domains like image and language modeling.

[12]Our experiments suggest the gains from increasing $H$ are small; hence, we focus on $H = 1$ to reduce computational costs.

The nonlinear portfolio transformer achieves extremely high complexity through the repeated composition of heavily parameterized functions. This seemingly gratuitous layering of nonlinearity on top of nonlinearity is motivated by the virtue of complexity. However, the nonlinear design choices in the Vaswani et al. (2017) architecture are by no means arbitrary. They are the result of continual research refinements discovered largely in the language modeling domain. In this section, we attempt to provide intuition for each of the portfolio transformer's nonlinearities and why they are likely to be beneficial in the asset pricing context. A key facet of our empirical analysis explores the sensitivity of AIPM model performance as a function of model complexity.

### 3.2.1 Softmax Attention

The first nonlinearity to appear in the full portfolio transformer is the softmax function, $\sigma(YW_hY')$, in equation (13). This function works row-wise (i.e., asset-by-asset) on the attention matrix $A_t = YW_hY'$, converting each row to a probability distribution (each element is non-negative and rows sum to one):

$$\sigma(A_t)_{i,j} = \frac{\exp(A_{t,i,j})}{\sum_l \exp(A_{t,i,l})}.$$

While the general attention mechanism gathers information about asset $i$ from the characteristics of all other assets, the softmax operation selectively focuses attention on relatively few related assets while ignoring the rest. This beneficial property of softmax is described in the following lemma.

**Lemma 1 (Selective Softmax Attention)** *Consider the effect of multiplying $W_h$ by a constant $\alpha$ in $\sigma(X_tW_hX_t')$. As $\alpha \to +\infty$, we have*

$$\sigma(X_tW_hX_t')_{i,j} = \frac{1}{|\chi_t(i)|}\mathbf{1}_{j\in\chi_t(i)}, \tag{22}$$

*where $\chi_t(i)$ is the attention-worthy set of assets for gathering information about asset $i$:*

$$\chi_t(i) = \{j \in \{1, \cdots, N_t\} : \; X_{i,t}'W_hX_{j,t} \;\in\; \arg\max_{j_1} X_{i,t}'W_hX_{j_1,t}\}. \tag{23}$$

Recent theoretical results (Tarzanagh et al., 2023) suggest that transformers tend to operate in the asymptotic regime described by Lemma 1, achieving a binary partition of assets (or, more commonly, language tokens) into those that are relevant and those that are not for predicting the behavior of each other asset $i$.

### 3.2.2 Feed-forward Network

The second nonlinearity of the portfolio transformer is denoted $\mathcal{F}$ in equation (15). In each transformer block $k$, $\mathcal{F}$ receives the output from the preceding attention sublayer, $\mathcal{A}^{\mathcal{R}}\left(\mathcal{T}^{(k-1)}(X_t)\right)$, which like the original conditioning data contains $D$ characteristics for each of the $N_t$ assets. The characteristics in $\mathcal{A}^{\mathcal{R}}\left(\mathcal{T}^{(k-1)}(X_t)\right)$ have been reconfigured to incorporate attention-based information sharing and other nonlinearities in previous layers. $\mathcal{F}$ further transforms these with a shallow, fully connected feed-forward neural network and introduces a large number of new additional parameters every time the feed-forward layer is applied.

Despite the Byzantine structure of the feed-forward layer, this is perhaps the best-understood and least novel aspect of the transformer. Its role is to refine the information about each asset by parsing out the most predictive nonlinear representations of the inputs. Note that because $\mathcal{F}$ operates row-wise, it transforms the "characteristics" of each asset $i$ independently of the other assets. Thus, the nonlinearity applied in this layer applies to an asset's own features alone. Therefore, it does not directly contribute to the transformer's cross-asset information sharing.

The feed-forward component is the transformer's closest analogue to the neural network architecture employed by DKKM. Their model can be represented as

$$w_t = S(X_t)\lambda\,, \tag{24}$$

where $S$ is a neural network mapping applied to own-asset features (precluding cross-asset prediction) and achieves its complexity by expanding the original $D$ features to a much larger set of $P \gg D$ nonlinear transformations of those features. The DKKM model provides another natural benchmark for the AIPM by separating the effects of nonlinearities from the effects of cross-asset prediction. As we show in the empirical analysis, the combination of both nonlinearity and cross-asset prediction delivers the strongest model, outperforming models with only own-asset nonlinearity or only cross-asset prediction in a linear transformer.[13]

#### 3.2.3 Residual Connections

Both the attention and feed-forward sublayers undergo a residual connection step before passing on their output. Residual (or "skip") connections are common in deep learning models (Orhan and Pitkow, 2017) and are critical to the performance of transformers (Dong et al., 2021). The literature primarily attributes their benefit to stabilizing the optimization by helping to counteract so-called vanishing and exploding gradients during training.

## 4 Empirical Findings

### 4.1 Data

In this section, we dissect the empirical performance of AIPMs. To make the conclusions from this analysis as easy to digest as possible, we perform our analysis in a conventional setting with conventional data. We focus on the SDF portfolio problem using monthly data for US stocks over the period 1963 to 2022 (from Jensen et al., 2023, JKP henceforth). This open-source data set compiles an extensive collection of stock characteristics from the finance literature.[14] The universe includes NYSE/AMEX/NASDAQ stocks with CRSP share codes 10–12.

To leverage this data in a machine learning environment, it is helpful to have a stock-month panel with few missing values and consistency in the set of characteristics that are available over time. To this end, we follow the sample filters of DKKM. First, we restrict the raw dataset of 153 characteristics to a smaller group of 132 characteristics that have fewer than 30% missing values over the full sample. Next, we drop "nano" stocks whose

[13]This is consistent with evidence of nonlinear transformer performance in the language domain (Dong et al., 2021; Tarzanagh et al., 2023).

[14]JKP characteristics for individual stocks can be downloaded at `https://wrds-www.wharton.upenn.edu/pages/get-data/contributed-data-forms/global-factor-data/`. Detailed documentation and further factor portfolio data are available at `jkpfactors.com`.

market capitalization is below the 1st percentile of the NYSE sample. Then, we exclude stock-month observations with missing values for more than a third of the characteristics. Finally, we cross-sectionally rank-standardize each characteristic into the $[-0.5, 0.5]$ interval and replace any missing characteristic values with the cross-sectional median value of zero. After this filtering procedure, the resulting matrix $N_t \times D$ matrix of characteristics each month constitutes the conditioning set $X_t$ used in our empirical analysis.

## 4.2 Performance Metrics and Benchmark Models

We use two primary metrics to evaluate the out-of-sample performance of an asset pricing model. The first is the out-of-sample Sharpe ratio of the SDF portfolio. This is a natural model comparison statistic because the true SDF coincides with the mean-variance efficient portfolio.

The second metric is the out-of-sample Hansen and Jagannathan (1997) distance or HJD. As discussed in DKKM, the out-of-sample HJD statistic has attractive model comparison properties. It averages an SDF's squared pricing errors among all test assets with a fixed weighting matrix defined as the out-of-sample inverse covariance matrix of test assets. With this weighting matrix, the HJD can be interpreted as the out-of-sample pricing error of the portfolio of test assets that is most mispriced by the model. And because the weighting matrix is the same for all models, the HJD can be directly compared across models (unlike other alpha or GMM-based statistics). The set of test assets we study are the 132 JKP anomaly factors, though none of our conclusions are sensitive to this choice.

To infer the statistical significance of differences between models, we report pairwise SDF performance comparisons. In particular, for two models, A and B, we run an alpha regression

$$R_{A,t} = \alpha + \beta R_{B,t} + \epsilon_t \,, \tag{25}$$

where the out-of-sample SDF returns on either side of the regression are re-scaled to have 15% annualized volatility (to aid comparability of alphas across models). We report the annualized alpha estimate as well as its $t$ statistic.

We compare AIPMs to a variety of benchmark models. The first four benchmarks are standard small linear factor models from the literature. We also compare against a moderately large linear model and two large nonlinear models:

*FF6:* This is a six-factor model that includes the five factors of Fama and French (2015) plus their UMD momentum.

*SY:* This is a four-factor model that includes the "mispricing" factors of Stambaugh and Yuan (2017).

*HXZ:* This is a five-factor model that includes the $q$-factor model specification of Hou et al. (2015) augmented to include the expected growth factor of Hou et al. (2021).

*DHS:* This is a three-factor model that includes the long-horizon and short-horizon behavioral factors of Daniel et al. (2020).

*BSV:* This model uses SDF weights that are linear in stock characteristics, $w_t = X_t\lambda$, following Brandt et al. (2009). Because the number of characteristics $D$ in the JKP data is relatively large, we estimate $\lambda$ using a ridge penalty as in (11).[15]

*DKKM:* This is the shallow neural network SDF of DKKM based on random features. It is nonlinear and high-dimensional, but it precludes cross-asset predictability. The SDF weight specification is $w_t = S(X_t)\lambda$ where $S_{i,t}(k) = \text{ReLU}(\gamma_k' X_{i,t})$ for all assets $i$ with $k = 1, ..., P$. Our empirical analysis sets $P = 25{,}000$.[16]

*MLP:* This neural network SDF uses a multi-layer perceptron (MLP). Unlike DKKM, the MLP allows for network depth into the nonlinear specification, but it continues to use only own-asset predictability.[17] The corresponding network function, $w^{MLP}(X_t; \Theta^{MLP})$ uses over 300,000 parameters and minimizes the MSRR objective (9) (without a penalty term) with the same training algorithm as the nonlinear transformer (Section 4.3).

[15]We use the same ridge penalty selection as for the linear transformer (Section 4.3).

[16]DKKM use sin and cos activation functions in their construction of random features, while we use the more common ReLU activation. Our results are insensitive to the particular choice of non-linear activation.

[17]We consider 4 hidden layers, each layer of width 256, and ReLU activations, mimicking the feedforward layer of our nonlinear transformer model. Our data experiments suggest that the performance is increasing in the network width and saturates around a width of 256. By contrast, performance is U-shaped in the network depth and saturates at a depth of 4, making our benchmark choice conservative.

## 4.3 Training

We train all models in 60-month rolling training windows, with the first training window using signals from January 1963 through December 1967 (and corresponding returns data from February 1963 to January 1968). At the end of the training window we construct one-month ahead out-of-sample SDF portfolio returns for each model, then we roll the training sample forward one month and re-train. The first out-of-sample return observation is in February 1968, and the last is in December 2022.

Our linear attention model is the fully saturated specification of (11). We consider a grid of shrinkage parameters $z = 10^i,\ i \in \{-10, \cdots, 3\}$. We select the ridge penalty that optimizes the MSRR objective with leave-one-out cross-validation. We report statistics of model performance over the 1968–2022 test sample. With $D$=132 characteristics, the linear attention model is equivalent to a regression with $D^3$=2.2 million parameters. As we explain in Appendix B, this can be interpreted as a multi-head attention model with approximately 129 ($\approx D^3/(D^2+D)$) heads. The model complexity of linear attention—that is, the ratio of parameters to training observations—is over 35,000.

For the nonlinear transformer, we randomly initialize all elements of the self-attention matrices $Q, K$, and $V$ as $\mathcal{N}(0, \frac{1}{D})$. We initialize feed-forward network weights $\mathcal{W}_1^{(k)}$ as $\mathcal{N}\left(0, \frac{1}{d_f}\right)$ and $\mathcal{W}_2^{(k)}$ as $\mathcal{N}\left(0, \frac{1}{D}\right)$, and we initialize feed-forward biases at 0. We initialize the final layer output weights $\lambda$ as $\mathcal{N}\left(0, \frac{1}{D}\right)$. We train to minimize the MSRR objective (9) using the Adam gradient descent algorithm (Kingma and Ba, 2014). Because the initial network weights are randomly generated, model estimates are sensitive to the random seed. To minimize this dependency, we repeat the training ten times for different random seeds and then average the outcomes.

For the four simple benchmark models, we estimate the SDF as the sample tangency portfolio of factors in the training sample (with no ridge shrinkage).[18] For the BSV and DKKM models, ridge shrinkage is employed using the same ridge parameter selection method that we use for the linear attention model.

[18]Because the number of parameters in these benchmarks is small, the benefits of ridge shrinkage are negligible. We verify this is the case in our data. Allowing for ridge shrinkage results in no discernible improvement in SDF performance for the low-dimension benchmark models.

Table 1: **Portfolio Transformer Performance**

The table reports annualized out-of-sample Sharpe ratios and HJD pricing errors for each model over the full sample (Panel A) and the post-2002 sample (Panel B).

| | FF6 | HXZ | SY | DHS | BSV | DKKM | Lin. Attn. | MLP | Trans. |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | Panel A: 1968–2022 | | | | |
| Sharpe Ratio | 1.05 | 1.80 | 1.37 | 1.21 | 3.60 | 3.91 | 3.89 | 4.31 | 4.57 |
| Pricing Error | 0.55 | 0.42 | 0.53 | 0.61 | 0.15 | 0.13 | 0.14 | 0.13 | 0.09 |
| | | | | | Panel B: Post-2002 | | | | |
| Sharpe Ratio | 0.80 | 0.97 | 0.73 | 0.46 | 2.03 | 2.42 | 2.25 | 3.07 | 3.37 |
| Pricing Error | 0.75 | 0.73 | 0.88 | 0.87 | 0.52 | 0.44 | 0.48 | 0.42 | 0.34 |

Some of the simple benchmark models are unavailable for small portions of our February 1968 to December 2022 sample. In particular, out-of-sample SDF returns for FF6 are available from July 1968 to December 2022, HXZ from January 1972 to December 2022, SY from February 1968 to December 2016, and DHS from July 1977 to December 2018. Individual benchmark model statistics use that model's full available sample, and pairwise model comparisons use the intersection of the models' samples.

## 4.4 Anatomy of Model Performance

### 4.4.1 The Benchmark

Panel A of Table 1 reports the performance of asset pricing models in the full out-of-sample period from 1968 to 2022. The four low complexity benchmarks from the literature deliver similar performance, with out-of-sample SDF Sharpe ratios ranging from 1.05 on the low end (FF) to 1.8 on the high end (HXZ). In terms of pricing errors among anomaly factors, the smallest HJD is 0.42 (HXZ), while the largest pricing error is 0.61 (DHS).

Figure 2 shows the cumulative out-of-sample SDF returns of all models. A striking feature of this plot is the flattening of model returns around 2002. The scale of the plot is dictated by the performance of machine learning models, but on close inspection, the post-2002 flattening is evident for simple benchmarks as well. For this reason, Panel B of Table 1 reports out-of-sample model performance in the post-2002 subsample. In this later period, the performance of all simple benchmarks is notably worse, and the ranking of simple

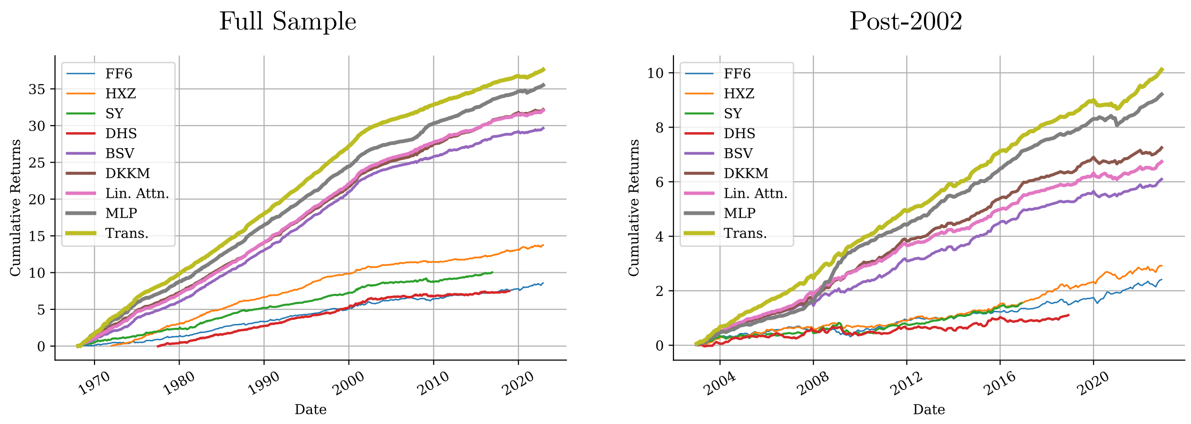


Figure 2: **Cumulative SDF Returns**

This figure shows the cumulative sum of SDF returns for the linear attention and nonlinear transformer models and benchmark models. The left panel shows the full out-of-sample period beginning in 1968, and the right panel begins in 2003.

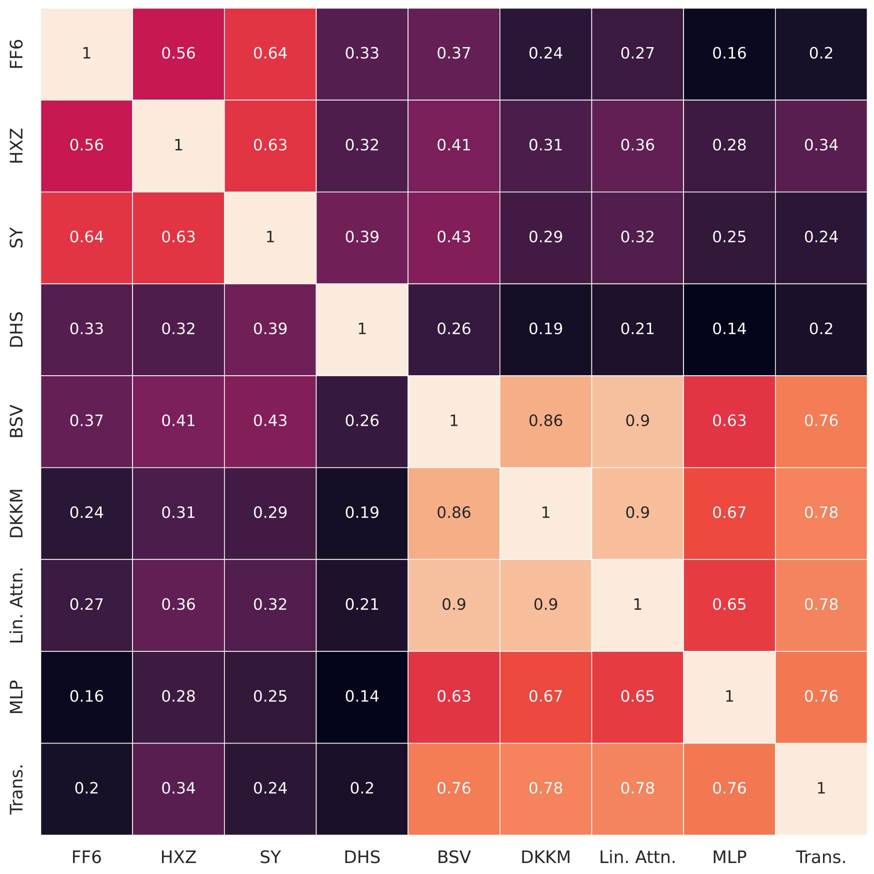
benchmarks changes. The best model in terms of the Sharpe ratio is still HXZ (0.97), but the worst is now DHS (0.46). In the later sample, HXZ achieves the smallest pricing errors (0.73), while SY has the largest (0.88). Figure 3 shows that the FF6, HXZ, and SY models are roughly 60% correlated with each other, while DHS is less correlated.

Table 2 reports alphas and their $t$-statistics in pairwise comparisons of out-of-sample SDF returns for all models (the result in each position of the table corresponds to regressing the row SDF on the column SDF). The last column reports the alpha from the regression of the row model on all other models. While simple benchmarks all perform somewhat similarly, Table 2 indicates that their performance differences are generally statistically significant at the 1% level, and HXZ appears dominant among simple factor models.

These four simple benchmark models summarize the behavior of asset pricing models that i) use small conditioning sets, ii) impose pre-specific linear structure, iii) rely on a small number of estimated parameters, and iv) restrict factors to use only own-asset predictive characteristics.

### 4.4.2 Expanding the Conditioning Set

The BSV model maintains the basic linear structure of standard low-dimensional benchmarks but increases the dimensionality of the SDF to 132 factors, corresponding to a complexity



| | FF6 | HXZ | SY | DHS | BSV | DKKM | Lin. Attn. | MLP | Trans. |
|---|---|---|---|---|---|---|---|---|---|
| FF6 | 1 | 0.56 | 0.64 | 0.33 | 0.37 | 0.24 | 0.27 | 0.16 | 0.2 |
| HXZ | 0.56 | 1 | 0.63 | 0.32 | 0.41 | 0.31 | 0.36 | 0.28 | 0.34 |
| SY | 0.64 | 0.63 | 1 | 0.39 | 0.43 | 0.29 | 0.32 | 0.25 | 0.24 |
| DHS | 0.33 | 0.32 | 0.39 | 1 | 0.26 | 0.19 | 0.21 | 0.14 | 0.2 |
| BSV | 0.37 | 0.41 | 0.43 | 0.26 | 1 | 0.86 | 0.9 | 0.63 | 0.76 |
| DKKM | 0.24 | 0.31 | 0.29 | 0.19 | 0.86 | 1 | 0.9 | 0.67 | 0.78 |
| Lin. Attn. | 0.27 | 0.36 | 0.32 | 0.21 | 0.9 | 0.9 | 1 | 0.65 | 0.78 |
| MLP | 0.16 | 0.28 | 0.25 | 0.14 | 0.63 | 0.67 | 0.65 | 1 | 0.76 |
| Trans. | 0.2 | 0.34 | 0.24 | 0.2 | 0.76 | 0.78 | 0.78 | 0.76 | 1 |

Figure 3: **Pairwise Correlations**

This figure shows pairwise correlations of SDF returns for all models.

of 2.2, given our 60-month training window. By increasing the set of conditioning variables to include more anomaly patterns from the literature, this linear and moderate complexity model shows a marked improvement in performance. Its out-of-sample Sharpe ratio is more than double that of the best simple model, and its pricing error drops by two-thirds. It has a large and significant alpha versus all four simple benchmark models. Like all models, its performance is weaker in the latter sample, but its relative outperformance of simple benchmarks is preserved.

### 4.4.3 Introducing Nonlinearities

DKKM uses the same large conditioning set as BSV but more flexibly leverages this conditioning information in a nonlinear specification. The thousands of DKKM factors are

Table 2: **Pairwise Model Comparison, Full Sample**

The table reports alphas and their $t$-statistics in pairwise comparisons of out-of-sample SDF returns for the sample from July 1977 to December 2016. For each pair of models, we report the alpha from regressing out-of-sample SDF returns from the "column" model on the SDF of the "row" model: $R_{\text{column},t} = \alpha + \beta R_{\text{row},t} + \epsilon_t$. The SDF returns on both sides of the regression are re-scaled to have 15% annualized volatility to aid comparability of alphas across models. We report alphas as annualized percentages along with $t$-statistics in parentheses. The last row reports the alpha from regressing the column model SDF on all other models jointly. Asterisks indicate statistical significance at the 99% confidence level.

| | FF6 | HXZ | SY | DHS | BSV | DKKM | Lin. Attn. | MLP | Trans. |
|---|---|---|---|---|---|---|---|---|---|
| FF6 | | 1.5* | 0.8* | 1.0* | 4.4* | 5.2* | 5.0* | 6.1* | 6.4* |
| | | (8.8) | (5.0) | (5.3) | (23.5) | (26.0) | (25.3) | (29.9) | (32.0) |
| HXZ | 0.1 | | 0.2 | 0.8* | 3.9* | 4.7* | 4.5* | 5.7* | 5.9* |
| | (0.7) | | (1.0) | (3.7) | (19.9) | (22.6) | (21.9) | (26.6) | (28.5) |
| SY | 0.3 | 1.2* | | 0.9* | 4.3* | 5.0* | 4.9* | 5.9* | 6.3* |
| | (2.0) | (7.4) | | (4.4) | (22.4) | (25.0) | (24.2) | (28.9) | (31.0) |
| DHS | 0.9* | 1.8* | 1.1* | | 4.6* | 5.3* | 5.1* | 6.2* | 6.5* |
| | (4.5) | (9.1) | (5.7) | | (22.8) | (25.6) | (24.9) | (29.6) | (31.5) |
| BSV | -0.8* | -0.1 | -0.5 | 0.2 | | 1.2* | 0.9* | 3.0* | 2.6* |
| | (-2.8) | (-0.5) | (-1.9) | (0.6) | | (8.0) | (6.8) | (13.3) | (15.6) |
| DKKM | -0.3 | 0.2 | -0.2 | 0.4 | 0.1 | | 0.3 | 2.5* | 2.3* |
| | (-1.0) | (0.6) | (-0.7) | (1.4) | (0.8) | | (2.5) | (10.8) | (12.0) |
| Lin. Attn. | -0.4 | 0.1 | -0.2 | 0.4 | 0.1 | 0.6* | | 2.7* | 2.3* |
| | (-1.4) | (0.5) | (-0.7) | (1.2) | (0.6) | (4.8) | | (11.5) | (12.9) |
| MLP | -0.1 | 0.5 | -0.1 | 0.6 | 0.7* | 1.2* | 1.1* | | 1.9* |
| | (-0.4) | (1.4) | (-0.2) | (1.8) | (2.8) | (4.6) | (4.3) | | (8.4) |
| Trans. | -0.5 | -0.3 | -0.5 | 0.2 | -0.7* | 0.1 | -0.2 | 1.2* | |
| | (-1.4) | (-1.0) | (-1.4) | (0.5) | (-3.4) | (0.6) | (-1.1) | (4.9) | |
| Others | -0.0 | 0.2 | -0.2 | 0.5 | -0.5* | 0.3 | 0.0 | 1.1* | 1.3* |
| | (-0.0) | (0.7) | (-0.8) | (1.5) | (-3.8) | (2.0) | (0.2) | (4.6) | (7.7) |

long-short portfolios based on stock characteristics that have been (jointly) transformed through a wide, shallow neural network. While the nonlinearities in DKKM allow for interactive effects among predictor variables, DKKM maintains the separation of stock-level information and rules out cross-stock prediction.

Extracting nonlinear information from characteristics with a shallow network improves model performance by about 10% over the linear BSV specification (an 8.6% rise in Sharpe ratio and a 13.3% reduction in pricing error). The benefits of nonlinearity are perhaps best

Table 3: **Pairwise Model Comparison, Post-2002 Sample**

The table repeats the analysis of Table 2 for the sample beginning in 2003. * indicates statistical significance at the 99% confidence level.

| | FF6 | HXZ | SY | DHS | BSV | DKKM | Lin. Attn. | MLP | Trans. |
|---|---|---|---|---|---|---|---|---|---|
| FF6 | | 0.7 | 0.4 | 0.3 | 2.4* | 3.4* | 3.0* | 4.4* | 4.4* |
| | | (2.3) | (1.4) | (0.9) | (8.4) | (10.6) | (9.6) | (13.2) | (14.3) |
| HXZ | 0.3 | | 0.0 | 0.2 | 2.3* | 3.3* | 2.9* | 4.4* | 4.2* |
| | (1.0) | | (0.1) | (0.6) | (7.8) | (10.3) | (9.2) | (12.9) | (14.1) |
| SY | 0.4 | 0.5 | | 0.3 | 2.4* | 3.4* | 3.0* | 4.4* | 4.4* |
| | (1.5) | (2.1) | | (0.8) | (8.4) | (10.7) | (9.6) | (13.3) | (14.8) |
| DHS | 0.8 | 1.0* | 0.7 | | 2.7* | 3.6* | 3.2* | 4.6* | 4.7* |
| | (2.4) | (3.1) | (2.3) | | (8.7) | (11.1) | (10.0) | (13.6) | (14.4) |
| BSV | -0.7 | -0.4 | -0.7 | -0.6 | | 1.2* | 0.8* | 3.0* | 2.4* |
| | (-2.0) | (-1.1) | (-1.9) | (-1.7) | | (6.0) | (4.3) | (8.9) | (10.9) |
| DKKM | -0.4 | -0.3 | -0.6 | -0.5 | -0.3 | | -0.0 | 2.2* | 1.8* |
| | (-0.9) | (-0.8) | (-1.4) | (-1.2) | (-1.4) | | (-0.2) | (6.5) | (6.9) |
| Lin. Attn. | -0.5 | -0.2 | -0.4 | -0.5 | -0.1 | 0.7* | | 2.7* | 2.1* |
| | (-1.2) | (-0.4) | (-1.1) | (-1.2) | (-0.5) | (3.8) | | (7.7) | (8.4) |
| MLP | 0.0 | 0.1 | -0.4 | 0.1 | 0.4 | 0.8 | 0.7 | | 1.8* |
| | (0.0) | (0.3) | (-0.9) | (0.3) | (0.9) | (2.1) | (1.8) | | (4.8) |
| Trans. | -1.2 | -1.3* | -1.6* | -0.8 | -1.2* | -0.2 | -0.5 | 1.4* | |
| | (-2.6) | (-2.9) | (-3.6) | (-1.7) | (-4.3) | (-0.7) | (-1.9) | (3.7) | |
| Others | 0.2 | 0.1 | -0.6 | 0.2 | -0.5 | 0.1 | 0.0 | 1.1* | 1.4* |
| | (0.4) | (0.2) | (-1.8) | (0.4) | (-2.5) | (0.6) | (-0.2) | (2.9) | (6.0) |

illustrated by the fact that DKKM has a large and significant alpha versus BSV, while the alpha of BSV over DKKM is essentially zero (both economically and statistically).

### 4.4.4 Information Sharing Via Attention

Linear attention introduces the possibility of cross-asset information sharing in the SDF. It does so in a particularly tractable way that pre-multiplies the BSV model's linear SDF weights with a dynamic attention matrix. The linear attention model achieves an out-of-sample Sharpe ratio of 3.89 and a pricing error of 0.14 in the full sample. This is a significant improvement over the BSV model.

The linear attention model can also be interpreted as a nonlinear SDF relying on triple interactions among characteristics for all stocks. As a nonlinear model, the linear attention only marginally improves over the nonlinear DKKM model, despite DKKM's more narrow reliance on own-stock predictive information, while DKKM has a significant alpha over linear attention. In the later part of the sample, linear attention continues to outperform the BSV specification, but it underperforms the shallow neural network of DKKM in terms of both Sharpe ratio and pricing error. Perhaps more surprisingly, the linear attention model is highly correlated (90%) with BSV and DKKM. Evidently, the simple linear attention specification largely fails to unlock any benefits of cross-asset information sharing.

### 4.4.5 Interactive Effects of Information Sharing and Nonlinearity

We now turn to the full nonlinear transformer specification. This uses multiple telescoping layers of attention blocks and nonlinearity to more flexibly incorporate predictive information from the conditioning variables. However, the transformer's depth and nonlinearity can aid the model's recovery of own-asset effects as well as cross-asset information sharing. Thus, to drill more specifically into the cross-asset information mechanism, we must benchmark the transformer to a multi-layer neural network that has similar nonlinearities and depth but that shuts down the cross-asset prediction. The MLP provides the necessary benchmark. With it, we can evaluate the information-sharing role of the transformer while controlling for deep nonlinearities in own-asset prediction.

MLP model performance shows that pure depth without cross-asset prediction is a powerful modeling device for asset returns. The out-of-sample MLP Sharpe ratio of 4.31 is a large improvement compared to the shallow DKKM model (Sharpe ratio of 3.91), though their pricing errors are about the same. In the more recent sample, the gains from depth are more pronounced with an MLP Sharpe ratio of 3.07 versus 2.42 for DKKM. In both samples the MLP has large and highly significant alpha versus DKKM. Thus, for the transformer to excel in cross-asset information sharing, it must exceed the MLP's high-performance bar.

By leveraging deep cross-asset information sharing, the transformer raises the out-of-

Table 4: **Tangency Portfolio Weights Across Models**

The table reports the estimated ex post optimal portfolio weights for the tangency portfolio of all factors subject to a non-negativity constraint. The top row reports tangency weights for the full sample, and the bottom row for the post-2002 sample.

| | FF6 | HXZ | SY | DHS | BSV | DKKM | Lin. Attn. | MLP | Trans. |
|---|---|---|---|---|---|---|---|---|---|
| Full sample | 0 | 0 | 0 | 0.03 | 0 | 0 | 0 | 0.37 | 0.60 |
| Post-2002 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.44 | 0.56 |

sample Sharpe ratio to 4.57. It reduces anomaly pricing errors by about 30%, from 0.13 for MLP to 0.09. In the post-2002 sample, the transformer Sharpe ratio is 3.37 versus 3.07 for MLP, and the pricing error is 0.34 versus 0.42 for MLP. These improvements are economically large and statistically significant, as indicated in the performance comparison of Table 2.

While BSV, DKKM, and linear attention are roughly 90% correlated with one another, MLP and the transformer are more differentiated from that group and from each other (the correlation of MLP and transformer is 76%). Rather than relying solely on pairwise comparisons and alpha tests, we can understand the relative performance and differentiation/diversification of a given SDF versus the set of all models by estimating the ex-post mean-variance combination of models. We do this by fixing the sample variance of all SDFs to 15% to put models on equal risk footing and impose a no-shorting constraint. The resulting portfolio weights are reported in Table 4. In the full sample, the transformer commands 60% of the tangency portfolio, followed by 37% for MLP and 3% for DHS (though DHS is insignificant based on its 95% bootstrap confidence interval). The Sharpe ratio of the ex-post tangency portfolio is 5.66, while the Sharpe ratio of the transformer model alone is 5.46.[19] Thus, the gains are small from supplementing the transformer with other model specifications.

The tangency weights are roughly the same in the post-2002 sample. In summary, asset pricing models evidently benefit from cross-asset information sharing when the attention mechanism is incorporated in a sufficiently deep network.

[19]As in the analysis of Table 2, the sample is restricted to the intersection of the available data for benchmark models, in this case July 1977 through December 2016. In this restricted sample, Sharpe ratio differ from the full sample results reported earlier. For example, the nonlinear transformer Sharpe ratio is 5.46 compared to 4.57 in the full period. This main discrepancy arises from the exclusion of the period 2017 to 2022 during which all models perform worse.

### 4.4.6 Information Sharing Through the Lens of Principal Portfolios

The principal portfolio formulation of Kelly et al. (2023) provides a rigorous framework for understanding the extent of cross-asset information sharing and cross-prediction in a trading strategy. They consider a single signal $S_t \in \mathbb{R}^{N_t}$ that predicts the cross-section of returns and analyze a class of strategies $w_t = AS_t$ that exploit return predictability. Their $A$ is a "prediction matrix" that can be thought of as a static attention matrix. They note that typical asset pricing analyses focus on the own-asset predictive effects of signals (take, for example, a momentum strategy in which each asset is bought or sold in proportion to its recent average return). They show that any strategy that relies on a symmetric matrix $A$ can be fully explained by a strategy that uses "own-asset" predictive information.

Kelly et al. (2023) also show that one can identify the distinctive cross-asset predictive effects of a strategy by analyzing the anti-symmetric component of $A$. In particular, the matrix $A$ can be decomposed as

$$A = A^s + A^a, \quad \text{where } A^s = 0.5(A + A') \quad \text{and } A^a = 0.5(A - A')$$

with the symmetric component of $A$ denoted $A^s$ and the anti-symmetric component denoted $A^a$. A strategy that relies purely on the anti-symmetric cross-prediction properties of the signal $S_t$ will be unexplained by a symmetric (own-prediction) strategy based on $S_t$.

We can adapt this Kelly et al. (2023) formalism to the linear attention modeling framework as follows:

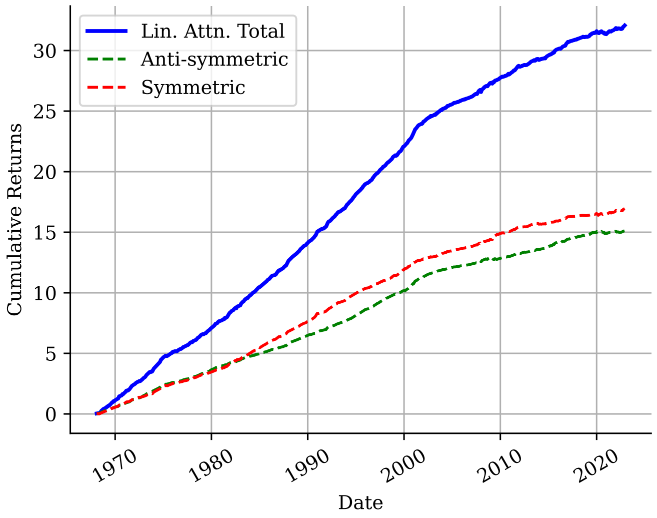
$$\begin{aligned} w_t &= A_{1,t}(X_t\lambda_1) + \cdots + A_{H,t}(X_t\lambda_H) \\ &= (A^s_{1,t} + A^a_{1,t})(X_t\lambda_1) + \cdots + (A^s_{H,t} + A^a_{H,t})(X_t\lambda_H) \\ &= \underbrace{A^s_{1,t}(X_t\lambda_1) + \cdots + A^s_{H,t}(X_t\lambda_H)}_{\text{symmetric component}} + \underbrace{A^a_{1,t}(X_t\lambda_1) + \cdots + A^a_{H,t}(X_t\lambda_H)}_{\text{anti-symmetric component}} \\ &= w^s_t + w^a_t. \end{aligned}$$

Table 5 reports the performance of the linear attention model through the lens of this symmetry decomposition. We find that both symmetric and anti-symmetric sub-strategies exhibit similar performance, with the symmetric part slightly outperforming. This is consistent

Table 5: **Symmetry Decomposition of Linear Attention Model**

The table reports the performance of symmetric and anti-symmetric components of the linear attention model SDF over the full sample.

| | Lin. Attn. Total | Anti-symmetric Component | Symmetric Component |
|---|---|---|---|
| Sharpe Ratio | 3.89 | 3.10 | 3.23 |
| Pricing Error | 0.14 | 0.26 | 0.21 |



with the general belief that symmetric dependencies are more stable and robust (own-asset prediction, but also shared market conditions, shared industry dynamics, and symmetric economic ties). The behaviors of the two components, however, are quite distinct. Their out-of-sample returns share only a 32% correlation. Thus, the high return on the overall linear attention SDF derives from its ability to complement the usual "own-prediction" effects of conditioning characteristics with remarkably potent cross-asset predictive effects.

The symmetry decomposition is convenient to apply to the linear attention model. However, the multi-layer non-linear transformer builds cross-attention recursively, propagating cross-asset information through all the transformer blocks. This makes extraction of anti-symmetric cross-prediction effects much more cumbersome, and we leave this for future research. However, Table 3 shows that the performance of the linear attention model is subsumed by the nonlinear transformer, providing indicative evidence that a substantial component of the nonlinear transformer's performance is due to cross-asset information sharing as well.

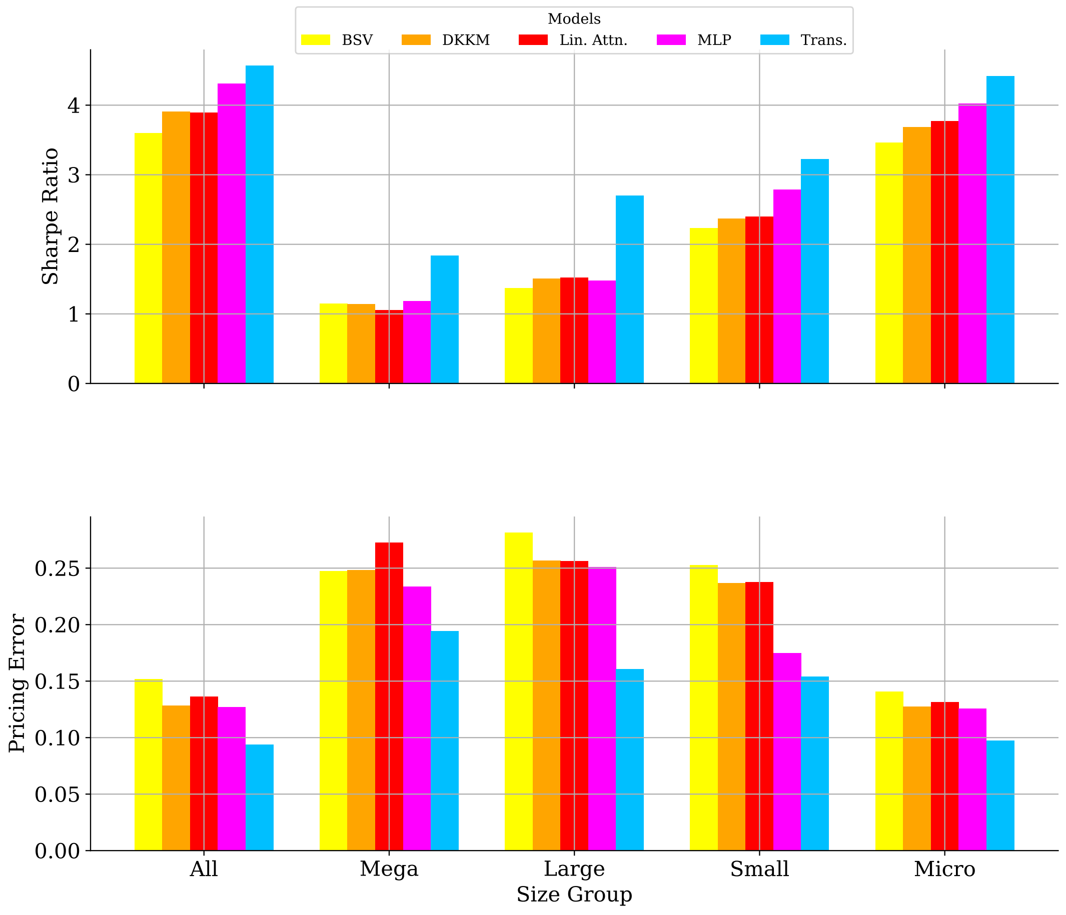


Figure 4: **Model Performance By Size Group**

This figure reports Sharpe ratios (top panel) and HJD pricing errors (bottom panel) for SDF models trained in data sets restricted to stocks in a given size group.

#### 4.4.7 Large Caps and Information Sharing

The models that we have studied thus far use a fixed specification that applies uniformly to all stocks, large and small, liquid and illiquid. And given the very high Sharpe ratios of all the machine learning specifications that we study, it is likely that much of the performance of these models is generated from the relatively illiquid subset of stocks in our universe (note, however, that our sample already filters out "nano" stocks that fall below the first percentile of the NYSE size distribution).

To investigate the role of cross-asset information sharing in more detail, we re-estimate

Table 6: **Tangency Portfolio Weights Across Models By Size Group**

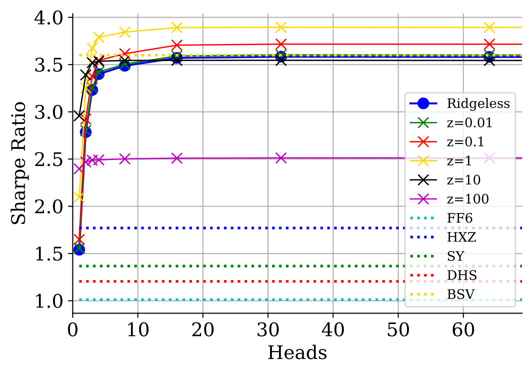
The table reports the estimated ex post optimal portfolio weights for the tangency portfolio of all factors subject to a non-negativity constraint for the full sample. Models are re-estimated for data sets restricted to stocks in a given size group, and then the ex-post tangency portfolio combines the size-group-specific SDF models.

| | Mega | Large | Small | Micro |
|---|---|---|---|---|
| BSV | 0.09 | 0 | 0 | 0 |
| DKKM | 0 | 0 | 0 | 0 |
| Lin. Attn. | 0 | 0 | 0 | 0.23 |
| MLP | 0.01 | 0 | 0.20 | 0.23 |
| Trans. | 0.91 | 1.00 | 0.80 | 0.54 |
| Tangency SR | 1.84 | 2.70 | 3.25 | 4.59 |
| Trans. SR | 1.84 | 2.70 | 3.23 | 4.42 |

each model separately in sub-universes based on market capitalization. “Mega” stocks are those above the 80th percentile of the NYSE size distribution each month, “large” includes percentiles 50 to 80, “small” includes 20 to 50, and “micro” includes stocks below the 20th but above the 1st NYSE size percentile.

Comparative model performance by size universe is reported in Figure 4. The first major conclusion from this figure is that there is a high degree of similarity in performance metrics for the full universe and the micro-universe, confirming that models trained in the full cross-section of stocks are most heavily influenced by the predictability of micro stocks. The second major conclusion is that the transformer is especially impactful among large and mega universes. Among mega stocks, the transformer Sharpe ratio is 1.84 versus 1.05 to 1.18 for the remaining machine learning models. This gain in Sharpe ratio from transformer-based information (i.e., versus MLP) is larger than the gain of MLP versus the BSV model. The same pattern emerges for the large-cap universe, with the transformer outperforming all other machine-learning models by a large margin while the other machine-learning models are largely undifferentiated. The pattern among pricing errors mirrors the Sharpe ratio results by size group.

The distinctive benefits of the transformer model versus other machine learning specifications are well summarized in the ex-post tangency portfolio weights reported in Table 6. Because there is no natural notion of size sub-groups for the standard simple benchmark



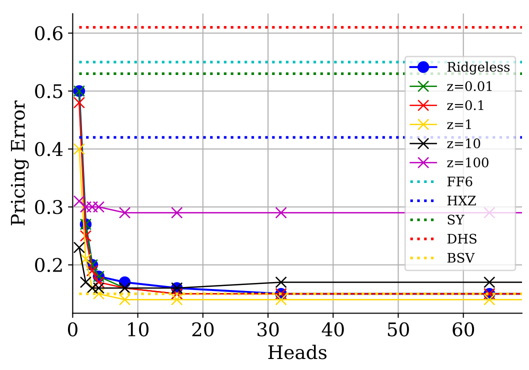


Figure 5: **Complexity and Linear Attention Model Performance**

This figure reports out-of-sample Sharpe ratio (left panel) and pricing error (right panel) as a function of the number of heads in the linear attention model with $D = 132$ characteristics. The simplest model we consider uses a single attention head and the most complex model is fully saturated with $D^3$ unique parameters (132 heads). Different curves illustrate how the degree of ridge shrinkage affects model performance. For reference, dotted lines show the performance of select benchmark models.

models, they are excluded from this analysis. For large and mega stocks, the transformer is the only model that receives substantial weight in the tangency portfolio of all candidate SDFs. In the full cross-section, the transformer continues to be responsible for the lion's share of the tangency portfolio, but other machine learning models, such as MLP, also earn meaningful weight, driven by their usefulness in the micro-cap universe.

### 4.4.8 Notions of Complexity

DKKM demonstrate that out-of-sample SDF performance is increasing in model parameterization, or "complexity," in the context of asset pricing models that rely on shallow neural networks. They show theoretically that complexity is beneficial for asset pricing models when there are a large number of common factors underlying the cross-section of returns, and they argue that US stocks appear to satisfy the conditions that give rise to a "virtue of complexity."

In attention-based asset pricing models, complexity arises through the use of multiple attention heads (particularly in the case of linear attention) and by stacking together multiple transformer blocks. Figure 5 reports the performance of linear attention as we vary the model from a single attention head to the maximum possible number of heads. Here we see a benefit from using more heads, all else equal, with out-of-sample performance eventually

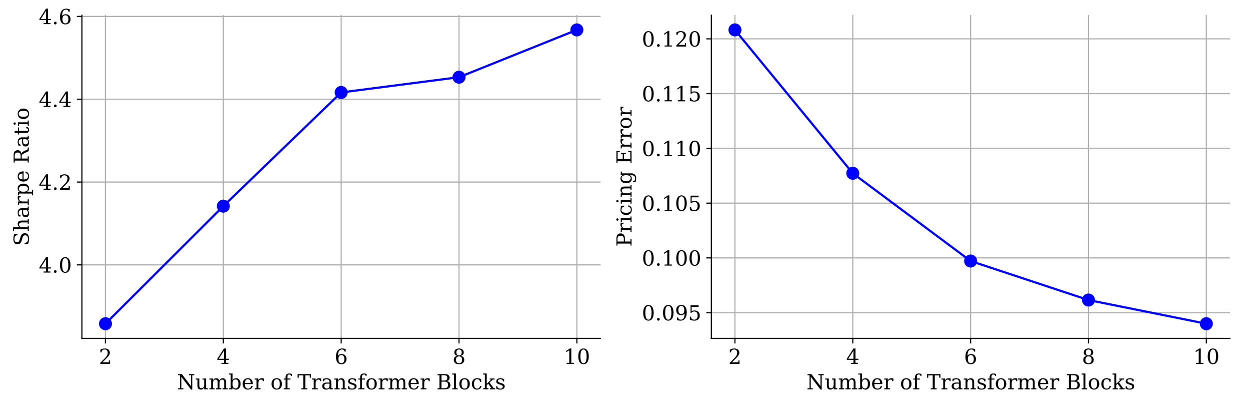


Figure 6: **Complexity and Transformer Model Performance**

This figure reports out-of-sample Sharpe ratio (left panel) and pricing error (right panel) as a function of the number of blocks in the nonlinear transformer model. The simplest model we consider uses a single transformer block and the most complex model uses 10 blocks.

flattening after about 20 heads (corresponding to over 5,000 times as many parameters as training observations). Similarly, in Figure 6, we see that the out-of-sample performance of the portfolio transformer model improves with the model depth up to 10 transformer blocks. In the case of transformer models, there may be additional benefits to even deeper specifications, but due to the computational costs of training deep transformers, we leave further exploration for future research.

In summary, in the context of AIPMs, more heavily parameterized model specifications tend to dominate simpler model variants. Empirical evidence from the literature on large language models (e.g. Kaplan et al., 2020) indicates that adding more transformer blocks enhances models' abilities to effectively represent language. Deeper architectures enable the model to capture more abstract features and longer-range dependencies than shallower models. Each additional layer refines the attention distributions, allowing the model to consider both short-term and long-term relationships. It is interesting that the same benefits of transformer complexity emerge in the asset pricing model context.

## 5 Conclusion

This paper introduces the concept of an Artificial Intelligence Pricing Model (AIPM) by embedding transformer architectures into the stochastic discount factor framework. Leveraging the principles that have propelled advances in natural language processing—context awareness and model complexity—we demonstrate how transformers can significantly enhance asset pricing models through cross-asset information sharing and nonlinearity.

Our first major contribution is the development of the linear portfolio transformer, an interpretable model that incorporates the attention mechanism to facilitate information sharing across assets while maintaining analytical tractability. While more simplistic than the full nonlinear transformer, the linear attention model is a useful surrogate for understanding how attention enhances the SDF by capturing conditional relationships among assets. Through theoretical derivations, we illustrate that the attention-based SDF can be viewed as a refined combination of characteristic-managed factor portfolios.

Building on this foundation, our second contribution involves the implementation of a full nonlinear transformer architecture within the SDF. This model incorporates advanced features such as multi-headed attention, softmax transformations, and deep stacking of transformer blocks, further exploiting the benefits of context awareness and model complexity. We describe the role played by each component for enhancing the model's predictive capabilities.

Empirically, we evaluate the performance of AIPMs using monthly U.S. stock return data and a comprehensive set of 132 stock-level conditioning characteristics. The linear portfolio transformer demonstrates that even in a linear setting, cross-asset information sharing yields non-trivial improvements in out-of-sample Sharpe ratios and pricing errors compared to models without attention mechanisms. The nonlinear portfolio transformer further amplifies these gains, outperforming existing machine learning models by achieving higher Sharpe ratios, lower pricing errors, and significant alpha over benchmarks lacking cross-asset attention. Our findings reveal a clear hierarchy among models: as we incorporate more complex specifications and attention mechanisms, performance consistently improves.

## References

Avramov, Doron, Si Cheng, and Lior Metzker, 2023, Machine learning vs. economic restrictions: Evidence from stock return predictability, *Management Science* 69, 2587–2619.

Bachlechner, Thomas, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley, 2021, Rezero is all you need: Fast convergence at large depth, in *Uncertainty in Artificial Intelligence*, 1352–1361, PMLR.

Bahdanau, Dzmitry, 2014, Neural machine translation by jointly learning to align and translate, *arXiv preprint arXiv:1409.0473* .

Bai, Yu, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei, 2023, Transformers as statisticians: Provable in-context learning with in-context algorithm selection, *arXiv preprint arXiv:2306.04637* .

Bartlett, Peter L, Philip M Long, Gábor Lugosi, and Alexander Tsigler, 2020, Benign overfitting in linear regression, *Proceedings of the National Academy of Sciences* 117, 30063–30070.

Belkin, M, D Hsu, S Ma, and S Mandal, 2018, Reconciling modern machine learning and the bias-variance trade-off. arxiv e-prints.

Belkin, Mikhail, Daniel Hsu, and Ji Xu, 2020, Two models of double descent for weak features, *SIAM Journal on Mathematics of Data Science* 2, 1167–1180.

Belkin, Mikhail, Alexander Rakhlin, and Alexandre B Tsybakov, 2019, Does data interpolation contradict statistical optimality?, in *The 22nd International Conference on Artificial Intelligence and Statistics*, 1611–1619, PMLR.

Bolya, Daniel, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman, 2022, Hydra attention: Efficient attention with many heads, in *European Conference on Computer Vision*, 35–49, Springer.

Brandt, Michael W, Pedro Santa-Clara, and Rossen Valkanov, 2009, Parametric portfolio policies: Exploiting characteristics in the cross-section of equity returns, *The Review of Financial Studies* 22, 3411–3447.

Britten-Jones, Mark, 1999, The sampling error in estimates of mean-variance efficient portfolio weights, *The Journal of Finance* 54, 655–671.

Bryzgalova, Svetlana, Markus Pelger, and Jason Zhu, 2020, Forest through the trees: Building cross-sections of stock returns, *Available at SSRN 3493458* .

Chen, Andrew Y, and Tom Zimmermann, 2021, Open source cross-sectional asset pricing, *Critical Finance Review, Forthcoming* .

Chen, Luyang, Markus Pelger, and Jason Zhu, 2023, Deep learning in asset pricing, *Management Science* .

Chinco, Alex, Adam D Clark-Joseph, and Mao Ye, 2019, Sparse signals in the cross-section of returns, *The Journal of Finance* 74, 449–492.

Cho, Kyunghyun, 2014, Learning phrase representations using rnn encoder-decoder for statistical machine translation, *arXiv preprint arXiv:1406.1078* .

Cong, Lin William, Guanhao Feng, Jingyu He, and Xin He, 2022, Growing the efficient frontier on panel trees, *NBER Working Paper* .

Cong, Lin William, Ke Tang, Jingyuan Wang, and Yang Zhang, 2021, Alphaportfolio: Direct construction through deep reinforcement learning and interpretable ai, *SSRN Electronic Journal. https://doi. org/10.2139/ssrn* 3554486.

Connor, Gregory, Matthias Hagmann, and Oliver Linton, 2012, Efficient semiparametric estimation of the fama–french model and extensions, *Econometrica* 80, 713–754.

Daniel, Kent, David Hirshleifer, and Lin Sun, 2020, Short-and long-horizon behavioral factors, *The review of financial studies* 33, 1673–1736.

Didisheim, Antoine, Shikun Barry Ke, Bryan T Kelly, and Semyon Malamud, 2024, Apt or aipt: The surprising dominance of large factor models, Technical report, National Bureau of Economic Research.

Dong, Yihe, Jean-Baptiste Cordonnier, and Andreas Loukas, 2021, Attention is not all you need: Pure attention loses rank doubly exponentially with depth, in *International Conference on Machine Learning*, 2793–2803, PMLR.

Ehsani, Sina, and Juhani T Linnainmaa, 2022, Factor momentum and the momentum factor, *The Journal of Finance* 77, 1877–1919.

Fama, Eugene F, and Kenneth R French, 2015, A five-factor asset pricing model, *Journal of financial economics* 116, 1–22.

Fan, Jianqing, Zheng Tracy Ke, Yuan Liao, and Andreas Neuhierl, 2022, Structural deep learning in conditional asset pricing, *Available at SSRN 4117882* .

Fan, Jianqing, Yuan Liao, and Weichen Wang, 2016, Projected principal component analysis in factor models, *Annals of statistics* 44, 219.

Feng, Guanhao, Stefano Giglio, and Dacheng Xiu, 2020, Taming the factor zoo: A test of new factors, *The Journal of Finance* 75, 1327–1370.

Freyberger, Joachim, Andreas Neuhierl, and Michael Weber, 2020, Dissecting characteristics nonparametrically, *The Review of Financial Studies* 33, 2326–2377.

Giglio, Stefano, Bryan Kelly, and Dacheng Xiu, 2022, Factor models, machine learning, and asset pricing, *Annual Review of Financial Economics* 14, 337–368.

Giglio, Stefano, and Dacheng Xiu, 2021, Asset pricing with omitted factors, *Journal of Political Economy* 129, 1947–1990.

Gu, Shihao, Bryan Kelly, and Dacheng Xiu, 2020a, Autoencoder asset pricing models, *Journal of Econometrics* .

Gu, Shihao, Bryan Kelly, and Dacheng Xiu, 2020b, Empirical asset pricing via machine learning, *The Review of Financial Studies* 33, 2223–2273.

Guijarro-Ordonez, Jorge, Markus Pelger, and Greg Zanotti, 2021, Deep learning statistical arbitrage, *arXiv preprint arXiv:2106.04028* .

Gupta, Tarun, and Bryan Kelly, 2019, Factor momentum everywhere, *The Journal of Portfolio Management* 45, 13–36.

Haddad, Valentin, Serhiy Kozak, and Shrihari Santosh, 2020, Factor timing, *The Review of Financial Studies* 33, 1980–2018.

Han, Yufeng, Ai He, David Rapach, and Guofu Zhou, 2019, Expected stock returns and firm characteristics: E-lasso, assessment, and implications, *SSRN* .

Hansen, Lars Peter, and Ravi Jagannathan, 1997, Assessing specification errors in stochastic discount factor models, *The Journal of Finance* 52, 557–590.

Hansen, Lars Peter, and Scott F Richard, 1987, The role of conditioning information in deducing testable restrictions implied by dynamic asset pricing models, *Econometrica: Journal of the Econometric Society* 587–613.

Harvey, Campbell R, Yan Liu, and Heqing Zhu, 2016, ... and the cross-section of expected returns, *The Review of Financial Studies* 29, 5–68.

He, Ai, Dashan Huang, Jiaen Li, and Guofu Zhou, 2023, Shrinking factor dimension: A reduced-rank approach, *Management science* 69, 5501–5522.

He, Songrun, Ming Yuan, and Guofu Zhou, 2022, Principal portfolios: The multi-signal case, *Available at SSRN 4245333* .

Hou, Kewei, Haitao Mo, Chen Xue, and Lu Zhang, 2021, An augmented q-factor model with expected growth, *Review of Finance* 25, 1–41.

Hou, Kewei, Chen Xue, and Lu Zhang, 2015, Digesting anomalies: An investment approach, *The Review of Financial Studies* 28, 650–705.

Hou, Kewei, Chen Xue, and Lu Zhang, 2020, Replicating anomalies, *The Review of Financial Studies* 33, 2019–2133.

Jensen, Theis Ingerslev, Bryan Kelly, and Lasse Heje Pedersen, 2023, Is there a replication crisis in finance?, *The Journal of Finance* 78, 2465–2518.

Jensen, Theis Ingerslev, Bryan T Kelly, Semyon Malamud, and Lasse Heje Pedersen, 2022, Machine learning and the implementable efficient frontier, *Available at SSRN 4187217* .

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei, 2020, Scaling laws for neural language models, *arXiv preprint arXiv:2001.08361* .

Kelly, Bryan, Semyon Malamud, and Lasse Heje Pedersen, 2023, Principal portfolios, *The Journal of Finance* 78, 347–387.

Kelly, Bryan, Semyon Malamud, and Kangying Zhou, 2024a, The virtue of complexity in return prediction, *The Journal of Finance* 79, 459–503.

Kelly, Bryan, Semyon Malamud, and Kangying Zhou, 2024b, The virtue of complexity in return prediction, *The Journal of Finance* 79, 459–503.

Kelly, Bryan, Seth Pruitt, and Yinan Su, 2020a, Characteristics are covariances: A unified model of risk and return, *Journal of Financial Economics* .

Kelly, Bryan, Seth Pruitt, and Yinan Su, 2020b, Instrumented principal component analysis, *Working paper* .

Kelly, Bryan, and Dacheng Xiu, 2023, Financial machine learning, *Foundations and Trends® in Finance* 13, 205–363.

Kelly, Bryan T, Semyon Malamud, and Kangying Zhou, 2022, The virtue of complexity everywhere, *Available at SSRN* .

Kingma, Diederik P, and Jimmy Ba, 2014, Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* .

Kozak, Serhiy, Stefan Nagel, and Shrihari Santosh, 2020a, Shrinking the cross-section, *Journal of Financial Economics* 135, 271–292.

Kozak, Serhiy, Stefan Nagel, and Shrihari Santosh, 2020b, Shrinking the cross-section, *Journal of Financial Economics* 135, 271–292.

Lettau, Martin, and Markus Pelger, 2020, Factors that fit the time series and cross-section of stock returns, *The Review of Financial Studies* 33, 2274–2325.

McLean, R David, and Jeffrey Pontiff, 2016, Does academic research destroy stock return predictability?, *The Journal of Finance* 71, 5–32.

Minaee, Shervin, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao, 2024, Large language models: A survey, *arXiv preprint arXiv:2402.06196* .

Orhan, Emin A, and Xaq Pitkow, 2017, Skip connections eliminate singularities, *arXiv preprint arXiv:1701.09175* .

Preite, Massimo Dello, Raman Uppal, Paolo Zaffaroni, and Irina Zviadadze, 2022, What is missing in asset-pricing factor models?

Rahimi, Ali, and Benjamin Recht, 2007, Random features for large-scale kernel machines., in *NIPS*, volume 3, 5, Citeseer.

Rapach, David E, and Guofu Zhou, 2020, Time-series and cross-sectional stock return forecasting: New machine learning methods, *Machine learning for asset management: New developments and financial applications* 1–33.

Spigler, Stefano, Mario Geiger, Stéphane d'Ascoli, Levent Sagun, Giulio Biroli, and Matthieu Wyart, 2019, A jamming transition from under-to over-parametrization affects generalization in deep learning, *Journal of Physics A: Mathematical and Theoretical* 52, 474001.

Stambaugh, Robert F, and Yu Yuan, 2017, Mispricing factors, *The review of financial studies* 30, 1270–1315.

Tarzanagh, Davoud Ataee, Yingcong Li, Christos Thrampoulidis, and Samet Oymak, 2023, Transformers as support vector machines, Preliminary version at NeurIPS M3L Workshop, 2023.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, 2017, Attention is all you need.(nips), 2017, *arXiv preprint arXiv:1706.03762* 10, S0140525X16001837.

## A Transformers and the Markowitz Plug-in Portfolio

In this section, we show that the linear portfolio transformer emerges rather naturally in artificial intelligence approaches to portfolio choice. Suppose we agree that $X_t$ reflects all relevant conditioning information for the SDF. However, we are unsure about the mapping from $X_t$ to the conditional moments of returns and opt for a nonparametric approach to estimating these moments.

More specifically, suppose that the conditional mean and precision matrix of returns are

$$E_t[R_{i,t+1}] = \mu(x_{i,t}) \quad \text{and} \quad (Cov_t(R_{t+1}))^{-1} = \Sigma^{-1}(X_t) = (\kappa(x_{i,t}, x_{j,t}))_{i,j=1}^{N} . \tag{26}$$

By the Mercer theorem[20] there exist nonlinear "feature functions" $f_\kappa(x) = (f_{\kappa,\ell}(x))_{\ell=1}^{L}$ such that, for large enough $L$, we can approximate $\Sigma^{-1}$ through its feature representation:

$$\kappa(x_{i,t}, x_{j,t}) \approx \sum_{\ell=1}^{L} f_{\kappa,\ell}(x_{i,t})\, f_{\kappa,\ell}(x_{j,t}) \tag{27}$$

Equation (27) is based on a standard argument that the unknown moment functions can be reconstituted from a large basis expansion of the variables in $X_t$. For simplicity, let us assume further that the approximations in (27) are exact.

Finding the nonparametric functions (27) requires the knowledge of the precise structure of the conditional covariance function $\kappa$. In reality, these functions need to be estimated, which is a non-trivial statistical problem. Instead, one can proceed as follows. Pick a basis $f(x) = (f_k(x))_{k=1}^{L}$ of non-linear functions with a universal approximation property[21]. Then, this basis possesses all the information needed to reconstitute both $\mu$ and $\Sigma^{-1}$ so long as each function is allowed its own set of weights on the basis terms:

$$\mu(x_{i,t}) \approx f(x_{i,t})\Lambda_\mu \quad \text{and} \quad f_\kappa(x_{i,t}) \approx f(x_{i,t})'\Lambda_\kappa \tag{28}$$

for the weight matrices $\Lambda_\mu \in \mathbb{R}^L$ and $\Lambda_\kappa \in \mathbb{R}^{L\times L}$. Based on these assumptions, the nonparametric approximations of $\mu$ and $\Sigma^{-1}$ give rise to the linear portfolio transformer as the

[20]See, e.g., Rahimi and Recht (2007).

[21]That is, such that any function can be approximated with linear combinations of $f_k$.

plug-in solution[22] to the Markowitz problem.

**Theorem 1** *The true SDF satisfies*

$$w_t = f(X_t) W f(X_t)' f(X_t)\lambda, \tag{29}$$

*where the $N_t \times L$ matrix $f(X_t)$ is the basis expansion of $X_t$ applied row-wise, $W = \Lambda_\kappa {\Lambda_\kappa}'$, and $\lambda = \Lambda_\mu$.*

This perspective on the portfolio transformer has a close association to the SDF design of DKKM. They advocate embedding the conditioning variables in a higher dimensional feature space in order to access the nonlinear predictive information in $X_t$. Specifically, they use random Fourier features (or RFF, see Rahimi and Recht, 2007) to embed the $D$-dimensional raw characteristics $x_{i,t}$ in an $L$-dimensional vector $f(x_{i,t})$ where $L \gg D$:

$$f(x_{i,t}) = \left(\sin(\omega_l' x_{i,t}), \cos(\omega_l' x_{i,t})\right)_{l=1}^{L/2}, \quad \omega_l \sim N(0, \gamma I) \ \forall l.$$

The corresponding $N_t \times L$ matrix of embeddings is $f(X_t)$, and their SDF model is defined as

$$w_t = f(X_t)\lambda. \tag{30}$$

The attractive aspect of (30) is that portfolio weights are determined by conditioning variables in complex, nonlinear fashioned. Yet, because the embedding parameters $\omega_l$ are randomized and not estimated, the RFF-MSRR problem maintains its tractability and closed-form solution. At last, we see that the model in (29) is thus the linear portfolio transformer analogue of the DKKM.

[22]The Markowitz plug-in solution substitutes for the true mean and covariance of returns moments with their corresponding estimators.

## B Theory of Linear Transformers

Let

$$L^{linear}(X_t) \; = \; X_t M' X_t' \tag{31}$$

be a linear attention matrix. Then, we have:

$$\begin{aligned}
\tilde{F}^{linear}_{k,t+1} &= X_t(k)'\, L^{linear}(X_t)\, R_{t+1} \\
&= X_t(k)'\,(X_t M' X_t') R_{t+1} \\
&= (X_t(k)'[X_t(1), \cdots, X_t(D)]) M' ([X_t(1), \cdots, X_t(D)]' R_{t+1}) \\
&= [X_t(k)' X_t(1), \cdots, X_t(k)' X_t(D)]\, M' \begin{pmatrix} F_{1,t+1} \\ \vdots \\ F_{D,t+1} \end{pmatrix} \\
&= \sum_{k_1=1}^{D} \underbrace{\Theta_t(k_1)}_{timing\ variables} \quad F_{k_1,t+1},
\end{aligned} \tag{32}$$

where we have defined

$$F_{k,t+1} \; = \; X_t(k)' R_{t+1}\,. \tag{33}$$

The formula (32) uncovers a striking algebraic identity: Although the attention-based factor $\tilde{F}^{linear}_{k,t+1}$ is formally based on $X_t(k)$, it is actually a *factor-timing portfolio,* combining all $D$ factors together, we timing variables that depend on $X_t(k)$.[23] That is, it is not attention that is helping the factor characteristic. It is the factor characteristic that helps attention; attention is the primary channel of alpha generation.

The attention mechanism in LLMs is commonly believed to be responsible for the phenomenon of "emergence": The unexpected ability of ML algorithms to solve problems they were formally not designed to solve. The powerful connection (32) between attention-based factors and factor timing portfolios could also potentially be viewed as a simple form of

[23]Namely, $\Theta_t(k_1) = \sum_{k_2=1}^{D} X_t(k)' X_t(k_2) M(k_1, k_2)$.

emergence, where the attention mechanism "discovers" factor timing, just like it "discovers" classical statistical algorithms in Bai et al. (2023).

Armed with this insight, we can now understand the low correlation between $\tilde{F}^{nonlinear}_{k,t+1}$ and $\tilde{F}^{linear}_{k,t+1}$. When optimal attention is trained with a non-linear $f$, it once again uses $X_{i,t}(k)$ not as a directional signal for stock $i$ but as a timing variable for other (highly complex and non-linear) factors "hidden" inside the $L^f$ matrix. We provide extensive evidence that such *non-linear cross-predictability* dominates the performance of attention-based models and

## B.1 Selecting Attention Heads with OLS and MSRR Objectives

We consider an extension of the basic linear transformer from (8). To this end, we assume that there are two types of features, $X_t \in \mathbb{R}^{N_t \times D}$ and $Z_t \in \mathbb{R}^{N_t \times d}$. The first type of features are used to construct attention matrices. The second one is used to construct features. For example, one could construct $Z_t$ or $X_t$ as *random features*, as in DKKM. Then, we define

$$\begin{aligned}\text{LMHA}(X_t) &= N_t^{-1}\underbrace{(X_t W_1 X_t')Z_t\lambda_1}_{head\ \#1} + \cdots + N_t^{-1}\underbrace{(X_t W_H X_t')Z_t\lambda_H}_{head\ \#H} \\ &= N_t^{-1}\left(\text{vec}(X_t'X_t)\otimes Z_t\right)\text{vec}\left(\sum_{h=1}^{H}\lambda_h'\otimes W_h\right) = S_t\check{\lambda} \qquad (34)\end{aligned}$$

be the Linear Multi-Head Attention (LMHA) portfolio, where we have defined

$$S_t \;=\; N_t^{-1}(X_t'X_t)\otimes Z_t\,. \qquad (35)$$

We consider ridge-penalized versions of the return prediction and portfolio optimization versions:

$$\begin{aligned}&OLS: \min_{\{W_h,\lambda_h\}_{h=1}^H}\left\{\frac{1}{T}\sum_{t=1}^{T}\|R_{t+1}-\text{LMHA}(X_t)\|^2 \;+ z\,\|\check{\lambda}\|^2\right\} \\ &MSRR: \min_{\{W_h,\lambda_h\}_{h=1}^H}\left\{\frac{1}{T}\sum_{t=1}^{T}(1-R_{t+1}'\text{LMHA}(X_t))^2 \;+\; z\,\|\check{\lambda}\|^2\right\}.\end{aligned} \qquad (36)$$

Our next key observation is that the nonlinearity of the optimal LMHA problem arises from its finite head structure. Indeed, a simple dimension counting implies that the set of $\check{\lambda} \in \mathbb{R}^{D^2 d}$

that can be represented as an $H$-head model (34) has dimension $(D^2+d)H$. If the number of heads is sufficiently large (e.g., $H \geq \min(D^2, d)$, this set coincides with $\mathbb{R}^{D^2d}$ and, hence, we can simply optimize (36) directly over all $\check{\lambda} \in \mathbb{R}^{D^2d}$; the number of heads $H$ plays no role for such "infinite heads" models.[24] We refer to the corresponding vectors $\check{\lambda}$ solving (36) as $W_{OLS}(H)$ and $W_{MSRR}(H)$, respectively. We abuse notation and use $W_{OLS}(\infty), W_{MSRR}(\infty)$ to denote $W_{OLS}(H)$ and $W_{MSRR}(H)$ for all $H \geq \min(D^2, d)$. The following is true.

**Lemma 2** *Suppose that $H \geq \min(D^2, d)$. Let $S = (S_t)_{t=1}^T \in \mathbb{R}^{T\times D^2 d}$. Then, the solutions $\check{\lambda}_{OLS}(H)$, $\check{\lambda}_{MSRR}(H)$ to the problems (36) are given by*

$$\begin{aligned}
\check{\lambda}_{OLS}(\infty) &= (T^{-1}S'S + zI_{D^2d\times D^2d})^{-1}\,T^{-1}S'R \\
\check{\lambda}_{MSRR}(\infty) &= (T^{-1}S'RR'S + zI_{D^2d\times D^2d})^{-1}\,T^{-1}S'R
\end{aligned} \tag{37}$$

Lemma 2 shows that the optimal LMHA $\check{\lambda}$ can be found in closed-form. However, such "infinite head" models lose the attractive interpretability of the finite-head models and make it difficult to understand how exactly the model learns the cross-predictability patterns among the different stocks. Most importantly, the infinite head $\check{\lambda}$ does not admit a unique decomposition into the single-attention-head components. For example, any decomposition $\lambda_h = \lambda_h^1 + \lambda_h^2$ leads to a different decomposition of $\check{\lambda}$. However, it is possible to characterize the optimal multi-head attention policies $\check{\lambda}_{OLS}(H)$, $\check{\lambda}_{MSRR}(H)$ directly in terms of $\check{\lambda}_{OLS}(\infty)$, $\check{\lambda}_{MSRR}(\infty)$. Furthermore, the decomposition we construct is unique, with the attention heads being pairwise orthogonal and ordered according to their importance.

To derive such an orthogonal multi-head decomposition, we start by noting that $\check{\lambda}(\infty) \in \mathbb{R}^{D^2d}$ can be "reshaped" as a $(D^2)\times d$ matrix. We abuse notation and use $\check{\lambda}$ to denote this reshaped vector. We can then use the singular value decomposition and write

$$\check{\lambda}_{type}(\infty) = \sum_{h=1}^{d} W_h \otimes \lambda_h,\ W_h \ \in\ \mathbb{R}^{D\times D},\ \lambda_h \in \mathbb{R}^D,\ type \in \{OLS, MSRR\}\,, \tag{38}$$

where $W_h \in \mathbb{R}^{D\times D}$ are $D^2$-dimensional eigenvectors of the matrix $\check{\lambda}(\infty)\check{\lambda}(\infty)' \in \mathbb{R}^{(D^2)\times(D^2)}$, while $\lambda_h \in \mathbb{R}^d$ are the $d$-dimensional eigenvectors of the matrix $\check{\lambda}(\infty)'\check{\lambda}(\infty) \in \mathbb{R}^{d\times d}$, with

[24]They are also known as "hydra attention" models in the literature; see, Bolya et al. (2022).

the singular values ordered in the decreasing order. We use

$$\check{\lambda}_{type}(\infty; H) = \sum_{h=1}^{H} W_h \otimes \lambda_h, \ \ type \in \{OLS, MSRR\} \tag{39}$$

to denote the approximation of $\check{\lambda}(\infty)$ with the top $H$ attention heads based on the singular value decomposition. By direct calculation and standard properties of the singular value decomposition, we have

$$\check{\lambda}_{OLS}(H) = \arg\min_{\check{\lambda}(H)} \|(\check{\lambda}_{OLS}(\infty) - \check{\lambda}(H))\|^2 \tag{40}$$

$$\check{\lambda}_{MSRR}(H) = \arg\min_{\check{\lambda}(H)} \|(\check{\lambda}_{OLS}(\infty) - \check{\lambda}(H))\|^2 \,. \tag{41}$$

As we now show, the true optimal $H$-head models solve closely related optimization problems. Indeed, by the properties of ordinary least squares regression, $R - S\check{\lambda}_{OLS}$ is orthogonal to the span of $S$ (and, similarly, for MSRR), and we get

$$\begin{aligned} \|R - S\check{\lambda}\|^2 &= \|R - S\check{\lambda}_{OLS}\|^2 \ + \ \|S(\check{\lambda}_{OLS} - \check{\lambda})\|^2 \\ \|1 - R'S\check{\lambda}\|^2 &= \|1 - R'S\check{\lambda}_{OLS}\|^2 \ + \ \|R'S(\check{\lambda}_{OLS} - \check{\lambda})\|^2 \,. \end{aligned} \tag{42}$$

These identities immediately imply the following result.

**Proposition 2** *For any $H < \min(D^2, d)$,*

$$\check{\lambda}_{OLS}(H) = \arg\min_{\check{\lambda}(H)} \|S(\check{\lambda}_{OLS}(\infty) - \check{\lambda}(H))\|^2 \tag{43}$$

$$\check{\lambda}_{MSRR}(H) = \arg\min_{\check{\lambda}(H)} \|R'S(\check{\lambda}_{OLS}(\infty) - \check{\lambda}(H))\|^2 \tag{44}$$

*where the minimum is over all $H$-heads $\check{\lambda}(H)$. Thus,*

- *If $S'S = I$, then $\check{\lambda}_{OLS}(\infty; H)$ is the solution to (43);*
- *If $S'RR'S = I$, then, $\check{\lambda}_{OLS}(\infty; H)$ is the solution to (44).*

Proposition 2 establishes a striking connection between finite-head LMHA models and singular value decomposition. For example, $W_{type}(1)$ is the "top" attention head, capturing the largest amount of cross-predictability. We refer to $W_{type}(1)$ as the *principal attention head*. This attention head is closely related to principal portfolios introduced in Kelly et al. (2023). Namely, Kelly et al. (2023) show how to find the optimal attention matrix $L$ for signals $S_t$, building a portfolio $LS_t$ using a singular value decomposition. Here, we show how to find the optimal $L$ of the form $L = X_t W X_t'$ with signals $S_t = Z_t \lambda$.

While, for generic $S$, the problems (43), (44) do not admit closed-form solutions, we believe the structure of these solutions is similar to that described in Proposition 2. The rotationally symmetric case of Proposition 2 implies that the problems (43), (44) have a tremendous number of local extrema: Any collection of $H$ different singular values corresponds to a local extremum, while the global minimum is unique, given by the $H$ largest singular values.

## B.2 Attention and Factor Timing

While Lemma 2 formally provides a simple, closed-form solution for the infinite-head attention, actually computing it and understanding its structure is highly non-trivial because it is *complex* in the sense of Kelly et al. (2022) and DKKM: The number of model parameters is very large even when compared to the size of the panel dataset. For example, already in the simpler case where $Z_t = X_t \in \mathbb{R}^D$, the optimal vector of parameters, $\check{\lambda}$, has the dimension $D^3$. A sufficiently rich set of characteristics (e.g., Jensen et al. (2023) suggest using $D = 150$ characteristics, while DKKM use $D = 130$) immediately leads to $D > 10^6$. As Kelly et al. (2022) explain, for the OLS prediction problem, the complexity should measured as $c_{OLS} = \dim(\check{\lambda}_{OLS})/(TN) = D^3/(TN)$, where $N$ is the average number of stocks in the panel. Even if we use a rolling window of 30 years, $TN < D^3$. The situation worsens for the MSRR problem where, as DKKM explain, complexity should be computed as $c = D^3/T$, implying a tremendous degree of complexity. Abstracting from the purely statistical considerations and the impossibility of correctly estimating the "true model" (due to the complexity wedge, DKKM), simply computing the solutions to (37) numerically is often infeasible because it requires inverting $D^2d$-dimensional matrices. This section shows that the optimal policies exhibit a remarkable mathematical structure, allowing us to compute these policies efficiently even for very large dimensions. Furthermore, this mathematical structure reveals important economic insights about the precise nature of predictive patterns that linear attention models are able to identify.

We start our analysis by defining $\Xi_t = X_t' Z_t / N_t \in \mathbb{R}^{Dd}$ and noting that the signals (35) can be written as $S_{i,t} = X_{i,t} \otimes \Xi_t$. As a result, defining the *characteristics-based factors* (also known as managed portfolios)

$$F_{t+1} = X_t' R_{t+1} \;\in\; \mathbb{R}^D\,, \tag{45}$$

portfolio returns can be rewritten as

$$\begin{aligned} \pi_t' R_{t+1} &= (S_t \check{\lambda})' R_{t+1} = \check{\lambda}'(\Xi_t' \otimes (X_t' R_{t+1})) \;=\; \check{\lambda}'(\Xi_t' \otimes F_{t+1}) \\ &= \sum_{j_1,j_2,j_3} \check{\lambda}(j_1,j_2,j_3) \underbrace{\Xi_t(j_1,j_2)}_{factor\ timing} \underbrace{F_{j_1,t+1}}_{factor\ returns}\,. \end{aligned} \tag{46}$$

The formula (46) shows how a generic linear attention model admits an intuitive factor-timing representation, whereby portfolio returns, $\pi_t' R_{t+1}$, optimally combine interactions of each factor $F_{j_1,t+1}$ with each timing variable $\Xi_t(j_1,j_2)$. The timing variables are cross-sectional averages of characteristics,

$$\Xi_t(j_1,j_2) = N_t^{-1} \sum_{i=1}^{N_t} X_{i,t}(j_1) Z_{i,t}(j_2)\,. \tag{47}$$

Such cross-sectional averages have indeed been shown to have factor timing ability. See, Haddad et al. (2020) and Kelly et al. (2023). In particular, Kelly et al. (2023) study cross-predictability. In our notation, they fix $j_2$ and study whether $\Xi_t(j_1,j_2)$ predicts not only $F_{j_1,t+1}$, but also $F_{j_2,t+1}$ for $j_2 \neq j_1$. Kelly et al. (2023) find strong evidence that this is indeed the case. The linear attention model in this section pushes the cross-predictability idea of Kelly et al. (2023) to extreme levels of complexity, optimizing factor timing *jointly*, as a portfolio (46) of $D^2 d$ factor timing combinations ($D$ factors timed with $Dd$ timing variables), and estimating the $D^2 d$-dimensional portfolio that optimally combines these $D^2 d$

timed factors. We now derive a representation for the optimal portfolio $\check{\lambda}$. Define

$$\Sigma^{\Xi}(t,\tau) = \Xi_t'\Xi_\tau = \underbrace{\Xi_t \cdot \Xi_\tau}_{timing\ attention} \tag{48}$$

to be the dot product similarity measure between timing variables $\Xi_t$ and $\Xi_\tau$ at two different time instants.

**Theorem 3** *For each type $\in \{OLS,\ MSRR\}$, there exists a sequence of vectors $q_t(type) \in \mathbb{R}^D,\ t \in [1, \cdots, T]$, such that*

$$\check{\lambda}_{type} = \sum_{t=1}^{T} q_t(type) \otimes \Xi_t \ \in \mathbb{R}^{D^2 d}\,. \tag{49}$$

*As a result, optimal portfolio returns are given by*

$$\pi_t(type)'R_{t+1} = F_{t+1}' \sum_{\tau=1}^{T} q_\tau(type)\, \Sigma^{\Xi}(t,\tau)\,. \tag{50}$$

Theorem 3 shows how optimal portfolio returns have a natural attention-based structure, whereby the optimal portfolio is a timing strategy, with the vector of factor weights is a linear combination of vectors $q_\tau(type)$, and the weights are determined by the similarity $\Sigma^{\Xi}(t,\tau)$ between timing variables today (at time $t$) and the in-sample time periods $\tau$.

## B.3 Computing Optimal Attention Heads

The following is true.

**Theorem 4 (Computing the Multi-Head Decomposition)** *Let*

$$\check{\lambda} = \sum_t q_t \otimes \Xi_t\,. \tag{51}$$

*be an LMHA model. Define $\Sigma^q \in \mathbb{R}^{T\times T}$ via $\Sigma^q(t_1,t_2) = q_{t_1}'q_{t_2}$, and recall than $\Xi_t \in \mathbb{R}^{D\times d}$.*

*Then,*

$$\check{\lambda}'\check{\lambda} = \sum_{t_1,t_2} \Sigma^q(t_1,t_2)\Xi'_{t_1}\Xi_{t_2} \ \in \mathbb{R}^{d\times d}\,. \tag{52}$$

*Let*

$$\check{\lambda}'\check{\lambda} = \sum_{h=1}^{d} \lambda_h\lambda'_h \tag{53}$$

*be its singular value decomposition. Then, the optimal multi-head attention is given by*

$$\check{\lambda}(\infty;H) = \sum_{h=1}^{H} W_h \otimes \lambda_h, \tag{54}$$

*with the optimal attention heads given by*

$$W_h = \check{\lambda}\lambda_h\,. \tag{55}$$

*Defining $\hat{b}(H) \in \mathbb{R}^{d\times H}$ to be the first $H$ singular vectors $[\lambda_1,\cdots,\lambda_h]$ and*

$$\tilde{\Xi}_t(H) = \Xi_t(\hat{b}(H)\hat{b}(H)'), \tag{56}$$

*we get that*

$$\pi_t(H)'R_{t+1} = \sum_\tau (F'_{t+1}q_\tau) \ \underbrace{(\Xi'_t\tilde{\Xi}_\tau(H))}_{timing\ attention} \tag{57}$$

## B.4 The Structure of Optimal Attention Vectors

Theorem 3 implies that the vectors $q_\tau$ depend in a complex fashion on the joint covariance structure of factors and timing variables. We now characterize this dependence in closed-form for both OLS and MSRR objectives.

**Proposition 5 (Optimal Timing Portfolios)** *The vectors $q_t$ in Theorem 3 can be com-*

*puted as follows:*

$$q_t(type) = z^{-1}F_{t+1} \ + \ v_t(type)\tilde{Q}(type)\,, \tag{58}$$

*where $v_t(type)$, $\tilde{Q}(type)$ can be computed as follows:*

- *For $type = OLS$, define*

$$\hat{\Gamma} = ((\Sigma^{\Xi})^{1/2} \otimes I)\,\mathrm{diag}(X_\tau' X_\tau)_{\tau=1}^T \ ((\Sigma^{\Xi})^{1/2} \otimes I) \in \mathbb{R}^{(DT)\times(DT)}\,. \tag{59}$$

*and let*

$$\hat{\Gamma} = \widetilde{V}\Lambda(MSE)\widetilde{V}' \tag{60}$$

*be its eigenvalue decomposition. Define*

$$\hat{V} = ((\Sigma^{\Xi})^{-1/2} \otimes I)\widetilde{V} \ \in \ \mathbb{R}^{(DT)\times(DT)} \tag{61}$$

*and write it as $\hat{V} = (v_1, \cdots, v_T) \in \mathbb{R}^{(DT)\times(DT)}$, where $v_t \in \mathbb{R}^{D\times(DT)}$. Then,*

$$\tilde{Q}(MSE) = \big((zI + \Lambda(MSE))^{-1} - z^{-1}\big) \circ (\sum_{\tau=1}^{T}\sum_{t=1}^{T} \Sigma^{\Xi}_{\tau,t}(v_\tau' F_{t+1})) \in \ \mathbb{R}^{DT}, \tag{62}$$

- *For $type = MSRR$, define*

$$\Sigma^F = (F_{\tau+1}' F_{t+1})_{t,\tau=1}^T \in \mathbb{R}^{T\times T} \tag{63}$$

*and let*

$$V\Lambda(MSRR)V' = \Sigma^F \circ \Sigma^{\Xi}, \quad V \in \mathbb{R}^{T\times T}. \tag{64}$$

*be the eigenvalue decomposition of $\Sigma^F \circ \Sigma^{\Xi}$, and $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_T)$. Define $\tilde{V} =$*

*$(\frac{1}{\sqrt{\lambda_1}}v_1, \cdots, \frac{1}{\sqrt{\lambda_T}}v_T)$, and denote $\tilde{V}_t$ the t-th row of $\tilde{V}$. Define*

$$v_t = (F_{t+1} \otimes \tilde{V}_t) \in \mathbb{R}^{D\times T}, \tag{65}$$

*where $\otimes$ stands for the outer product of $F_{t+1} \in \mathbb{R}^D$ and $\tilde{V}_t \in \mathbb{R}^T$. Then, $\tilde{Q}(MSRR)$ is given by*

$$\tilde{Q}(MSRR) = \big((zI + \Lambda(MSRR))^{-1} - z^{-1}\big) \circ \Lambda(MSRR)\, \tilde{V}'\mathbf{1}. \tag{66}$$

## C Proofs

**Proof of Proposition 5, OLS Case**. Denote $N$ the average number of stocks $NT = \sum_{t=1}^T N_t$ in the data panel. We start by proving the result for the MSE objective

$$\mathcal{L}_{OLS}(M) = \| \underbrace{R}_{NT\times 1} - \underbrace{\pi}_{NT\times 1} \|^2 = \frac{1}{T}\sum_{t=1}^T \| \underbrace{R_{t+1}}_{N_t\times 1} - \underbrace{S_t}_{N_t\times D^2 d} M\|^2. \tag{67}$$

Optimal $\check{\lambda} \in \mathbb{R}^{D^2 d\times 1}$ that minimizes the MSE objective is a solution to the OLS regression task $R = SM$ with a coefficient vector $M$. It is given by

$$\check{\lambda} = (\frac{1}{T}\underbrace{S'S}_{D^2d\times D^2d} + zI)^{-1}\frac{1}{T}\underbrace{S'}_{D^2d\times NT}\underbrace{R}_{NT\times 1}. \tag{68}$$

Simplifying the expression above boils down to finding eigenvectors of $S'S$. First, notice that any vector $V \in \mathbb{R}^{D^2 d}$ can be decomposed into the sum of Kronecker products of vectors $V = \sum_{k=1}^K (v_k \otimes w_k)$ for some $v_k \in \mathbb{R}^D$ and $w_k \in \mathbb{R}^{Dd}$. In general, we should require that the decomposition has at least one solution, i.e., the total number of unknowns on the right-hand side should be greater or equal to the number of entries of $V$ with a unit norm, i.e., $K(D + Dd) \geq D^2 d - 1$, this sets a lower bound on the parameter $K$. However, we will show that every eigenvector of $S'S$ has a unique decomposition by construction and that the final result holds for any $K$.

Recall that $S_{i,t} = X_{i,t} \otimes \Xi_t$ and, hence, $S_t = X_t \otimes \Xi_t$ and $S$ is the matrix of

$S_\tau,\ \tau \in [1, \cdots, T]$, stacked together. Then,

$$\begin{aligned} S'S &= \sum_{\tau=1}^T S_\tau S'_\tau = \sum_{\tau=1}^T (X_\tau \otimes \Xi_\tau)(X_\tau \otimes \Xi_\tau)' \\ &= \sum_{\tau=1}^T \underbrace{(X_\tau X'_\tau)}_{=\Sigma_\tau \in \mathbb{R}^{D\times D}} \otimes \underbrace{(\Xi_\tau \Xi'_\tau)}_{(Dd)\times(Dd)} \\ &= \Big(\sum_{\tau=1}^T \Sigma_\tau \otimes (\Xi_\tau \Xi'_\tau)\Big). \end{aligned} \tag{69}$$

Any vector $V \in R^{D^2d}$ can be written as a sum of decomposable vectors

$$q \otimes w\,. \tag{70}$$

Thus, the linear operator $S'S : R^{D^2d} \to R^{D^2d}$ acts on a vector $V \in R^{D^2d}$ as follows:

$$\begin{aligned} (S'S)_t Q &= \Big(\sum_{\tau=1}^T \Sigma_\tau \otimes (\Xi_\tau \Xi'_\tau)\Big) q \otimes w = \sum_{\tau=1}^T \Sigma_\tau q \otimes (\Xi_\tau \underbrace{\Xi'_\tau w}_{\in \mathbb{R}}) \\ &= \sum_{\tau=1}^T (\Xi'_\tau w)(\Sigma_\tau q) \otimes \Xi_\tau\,. \end{aligned} \tag{71}$$

Thus, Im$(S'S)$, the image of $S'S$, satisfies

$$\mathrm{Im}(S'S) \subset \Big\{\sum_{\tau=1}^T v_\tau \otimes \Xi_\tau | v_\tau \in \mathbb{R}^{Dd}\Big\}, \tag{72}$$

and, thus, *all* eigenvectors of $S'S$ are of the form (72), and we can look for these eigenvectors as follows:

$$S'S \sum_{\tau=1}^T v_\tau \otimes \Xi_\tau = \lambda \sum_{\tau=1}^T \Big(\sum_{\tau'=1}^T \Xi'_\tau \Xi_{\tau'} \Sigma_\tau v_{\tau'}\Big) \otimes \Xi_\tau\,. \tag{73}$$

Without loss of generality, we assume that $(\Xi_\tau)_{\tau=1}^T$ are linearly independent. Then, we can use the following lemma.

**Lemma 3** *If* $(\Xi_\tau)_{\tau=1}^T$ *are linearly independent, then a representation*

$$x \;=\; \sum_{\tau=1}^T v_\tau \otimes \Xi_\tau \tag{74}$$

*is unique.*

**Proof of Lemma 3**. Suppose that

$$\sum v_\tau \otimes \Xi_\tau \;=\; \sum w_\tau \otimes \Xi_\tau, \tag{75}$$

so that

$$\sum a_\tau \otimes \Xi_\tau \;=\; 0,\; a_\tau = v_\tau - w_\tau\,. \tag{76}$$

Then, taking an inner product with $b \otimes \Xi_{\tau'}$, we get

$$\sum (\Xi_{\tau'}'\Xi_\tau)(a_\tau' b) \;=\; 0\,. \tag{77}$$

Equivalently, $\Sigma^\Xi((a_\tau' b)_{\tau=1}^T) = 0$. Since $\Xi$ are linearly independent, $\Sigma^\Xi = (\Xi_\tau'\Xi_{\tau'})_{\tau,\tau'=1}^T$ is non-degenerate. Thus, $(a_\tau' b) = 0$ for all $\tau$. Since $b$ is arbitrary, we conclude that $a_\tau = 0$ for all $\tau$.
□

Thus, $(v_\tau)_{\tau=1}^T$ corresponding to an eigenvector $\sum_{\tau=1}^T v_\tau \otimes \Xi_\tau$ is unique. Therefore, (73) is equivalent to the system

$$\lambda v_\tau = \sum_{\tau'=1}^T \Xi_\tau'\Xi_{\tau'}\Sigma_\tau v_{\tau'}. \tag{78}$$

Define $\Sigma^\Xi \in \mathbb{R}^{T\times T} : (\Sigma^\Xi)_{\tau,\tau'} = \Xi_\tau'\Xi_{\tau'}$, $\Gamma = \text{diag}(\Sigma_\tau)_{\tau=1}^T \in \mathbb{R}^{Dt\times Dt}$. Also, let $V$ be a vertical stack of eigenvectors $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_T \end{bmatrix}' \in \mathbb{R}^{Dt}$. The eigenvalue equation (78) can be rewritten in tensor form as follows:

$$\lambda V = \Gamma(\Sigma^\Xi \otimes I_{D\times D})V. \tag{79}$$

Although this representation is already "compact," it is not convenient to work with because the product $\Gamma(\Sigma^\Xi \otimes I_{D\times D})$ is not symmetric (both $\Gamma$ and $\Sigma^\Xi \otimes I_{D\times D}$ are symmetric by construction, but the product of symmetric matrices is not necessarily symmetric). Define $\widetilde{V} = ((\Sigma^\Xi)^{1/2} \otimes I)V$. Then,

$$\Gamma(\Sigma^\Xi \otimes I_{D\times D})(\Sigma^\Xi)^{-1/2} \otimes I)\widetilde{V} = \Gamma((\Sigma^\Xi)^{1/2} \otimes I)\widetilde{V} = (\lambda/Dd)((\Sigma^\Xi)^{-1/2} \otimes I)\widetilde{V}, \tag{80}$$

which leads to

$$((\Sigma^\Xi)^{1/2} \otimes I)\Gamma((\Sigma^\Xi)^{1/2} \otimes I)\widetilde{V} = \widetilde{\Gamma}\widetilde{V} = \lambda\widetilde{V}. \tag{81}$$

Now, $\widetilde{\Gamma}$ is symmetric (because for any symmetric $A, B$ the product $ABA$ is symmetric) and, thus, recovering $\hat{\Psi}_t$ no longer requires taking the inverse of the eigenvector matrix. Finally, one recovers the matrix of eigenvectors of $\hat{\Psi}_t$ by computing

$$V_{\hat{\Psi}_t} = \sum_{\tau=1}^{T} v_\tau \otimes \Xi_\tau, \tag{82}$$

where each $v_\tau$ is a component of a vertical stack of $\tau$'s column of $V = ((\Sigma^\Xi)^{-1/2} \otimes I)\widetilde{V}$. The actual eigenvectors of $\hat{\Psi}_t$ are then

$$V \;=\; \sum_{\tau=1}^{T} v_\tau \otimes \Xi_\tau \tag{83}$$

and let us verify that these vectors are orthonormal. Indeed,

$$\begin{aligned} V'V &= \sum_{\tau=1}^{T} v'_\tau \otimes \Xi'_\tau \sum_{\tau=1}^{T} v_\tau \otimes \Xi_\tau = \sum_{\tau_1,\tau_2} v'_{\tau_1} v_{\tau_2} \Xi'_{\tau_1} \Xi_{\tau_2} = \sum_{\tau_1,\tau_2} v'_{\tau_1} v_{\tau_2} \Sigma^\Xi_{\tau_1,\tau_2} \\ &= \; V'(\Sigma^\Xi \otimes I)V = \tilde{V}'((\Sigma^\Xi)^{-1/2} \otimes I)(\Sigma^\Xi \otimes I)((\Sigma^\Xi)^{-1/2} \otimes I)\widetilde{V} \\ &= \tilde{V}'\tilde{V} \;=\; I. \end{aligned} \tag{84}$$

The calculation above allows us to compute eigenvectors for all non-zero eigenvalues. We denote the corresponding eigenvector matrix by $V$. Note that $V$ is an isometry and satisfies

$V'V = I_\nu$, where $\nu$ is the number of non-zero eigenvalues. However, $VV' \neq I_{D^2d}$ and, in the case of $\nu < D^2d$, the zero eigenvalues need to be taken special care of. The next calculation uses the following lemma.

**Lemma 4** *Let $V$ be the matrix of eigenvectors for a symmetric matrix with non-zero eigenvalues $\Lambda$. Then,*

$$(zI + A)^{-1} \;=\; V(zI+\Lambda)^{-1}V' \;+\; z^{-1}(I - VV')\,. \tag{85}$$

We will also use the identity

$$S'R \;=\; \sum_{t=1}^{T} F_{t+1} \otimes \Xi_t \tag{86}$$

Now, the optimal $\check{\lambda}$ can be determined as follows:

$$\begin{aligned}
\check{\lambda} \;&=\; (T^{-1}S'S + zI)^{-1}T^{-1}S'R = (S'S + \underbrace{(zT)}_{=:\widetilde{z}}I)^{-1}S'R \\
&= (V\Lambda V' + \widetilde{z}^{-1}I)^{-1}S'R = \Big(V(\widetilde{z}I+\Lambda)^{-1}V' + \widetilde{z}^{-1}(I - VV')\Big)S'R \\
&= \Big(V(\widetilde{z}I+\Lambda)^{-1}V' + \widetilde{z}^{-1}(I - VV')\Big)\sum_{t=1}^{T} F_{t+1}\otimes \Xi_t \\
&= \widetilde{z}^{-1}\sum_{t=1}^{T} F_{t+1}\otimes \Xi_t \;+\; V\big((\widetilde{z}I+\Lambda)^{-1} - \widetilde{z}^{-1}I\big)Q \in \mathbb{R}^{D^2d},
\end{aligned} \tag{87}$$

where $Q = V'\sum_{t=1}^{T} F_{t+1}\otimes\Xi_t \in \mathbb{R}^T$. We have, with $v_\tau \in \mathbb{R}^{D\times(DT)}$, that

$$\begin{aligned}
Q \;&=\; V'\sum_{t=1}^{T} F_{t+1}\otimes \Xi_t \;=\; (\sum_{\tau=1}^{T} v_\tau \otimes \Xi_\tau)'\sum_{t=1}^{T} F_{t+1}\otimes\Xi_t \\
&=\; \sum_{\tau=1}^{T}\sum_{t=1}^{T} \underbrace{(\Xi_\tau'\Xi_t)}_{\in\mathbb{R}}(v_\tau' F_{t+1}) \;=\; \sum_{\tau=1}^{T}\sum_{t=1}^{T} \Sigma^{\Xi}_{\tau,t}(v_\tau' F_{t+1}) \in\; \mathbb{R}^{DT}
\end{aligned} \tag{88}$$

For the optimal $\check{\lambda}$, we obtain

$$\begin{aligned}\check{\lambda} &= \tilde{z}^{-1}\sum_{t=1}^{T} F_{t+1}\otimes\Xi_t + V\tilde{Q} = \tilde{z}^{-1}\sum_{t=1}^{T} F_{t+1}\otimes\Xi_t + \sum_{t=1}^{T}(v_t\tilde{Q})\otimes\Xi_t \\ &= \sum_{t=1}^{T}(\tilde{z}^{-1}F_{t+1} + v_t\tilde{Q})\otimes\Xi_t = \sum_{t=1}^{T} q_t\otimes\Xi_t \in \mathbb{R}^{D^2\times d},\end{aligned} \tag{89}$$

where we have defined

$$q_t = \tilde{z}^{-1}F_{t+1} + v_t\tilde{Q}. \tag{90}$$

Here, $v_t \in \mathbb{R}^{D\times(DT)}$, $\tilde{Q}\in\mathbb{R}^{DT}$ and $v_t\tilde{Q}\in\mathbb{R}^D$.

The Proof of Proposition 5, OLS Case is complete. □

**Lemma 5 (A low-rank approximation)** *Suppose that* $\Sigma^\Xi = (\Xi_\tau'\Xi_{\tau'})_{\tau,\tau'=1}^T$ *has approximately low rank, so that*

$$\Sigma^\Xi \;\approx\; \sum_{l=1}^{r}\lambda_l(\Sigma^\Xi)\,\hat{q}_l\hat{q}_l' \tag{91}$$

*for some* $r < t$. *Define*

$$q_l \;=\; \lambda_l(\Sigma^\Xi)^{1/4}\hat{q}_l \;=\; \lambda_l((\Sigma^\Xi)^{1/2})^{1/2}\hat{q}_l\,. \tag{92}$$

*Define*

$$\bar{\Gamma}(l_1,l_2) \;=\; \sum_{k=1}^{\tau} q_{l_1}(k)\Gamma_k q_{l_2}(k), \tag{93}$$

*and*

$$\hat{\Gamma} \;=\; (\lambda_l(\Sigma^\Xi)^{1/4}\lambda_{l_1}(\Sigma^\Xi)^{1/4}\bar{\Gamma}(l,l_1))_{l,l_1=1}^r \;\in\; \mathbb{R}^{(rD)\times(rD)} \tag{94}$$

*and let* $\hat{V}\in\mathbb{R}^{(rD)\times(rD)}$ *and* $\hat{\Lambda}\in\mathbb{R}^{rD}$ *be the eigenvectors and eigenvalues of* $\hat{\Gamma}$ *.*

*Then, define* $\hat{V} \;=\; [\hat{v}_1, \cdots, \hat{v}_r]$ *as a partition into row blocks, and let*

$$\tilde{v}_\tau \;=\; \sum_{l_1=1}^{r} \hat{q}_{l_1}(\tau)\hat{v}_{l_1} \tag{95}$$

*stacking* $\tilde{v}_\tau$ *into row blocks, we get* $\tilde{V} \in \mathbb{R}^{(tD)\times(tr)}$.

**Proof of Lemma 5**. We have that

$$((\Sigma^\Xi)^{1/2} \otimes I)\Gamma((\Sigma^\Xi)^{1/2} \otimes I)\widetilde{V} = \widetilde{\Gamma}\widetilde{V} = \lambda\widetilde{V}. \tag{96}$$

can be rewritten as

$$\tilde{v}_\tau \;=\; \sum_{\tau_1=1}^{T}\sum_{k=1}^{\tau} s_{\tau,k}\Gamma_k s_{k,\tau_1}\tilde{v}_{\tau_1}\,, \tag{97}$$

where $s = (\Sigma^\Xi)^{1/2}$. Substituting the spectral decomposition

$$s \;=\; \sum_{l=1}^{r} q_l q_l', \tag{98}$$

where $r$ is the rank of $s$ and $q_l \in \mathbb{R}^T$ are rescaled eigenvectors, we get

$$\lambda\tilde{v}_\tau \;=\; \sum_{\tau_1=1}^{T}\sum_{k=1}^{\tau}\sum_{l_1,l_2} q_{l_1}(\tau)q_{l_1}(k)\Gamma_k q_{l_2}(\tau_1)q_{l_2}(k)\tilde{v}_{\tau_1}\,. \tag{99}$$

Defining

$$\bar{\Gamma}(l_1,l_2) \;=\; \sum_{k=1}^{\tau} q_{l_1}(k)\Gamma_k q_{l_2}(k), \tag{100}$$

we can rewrite it as

$$\lambda\tilde{v}_\tau \;=\; \sum_{l_1,l_2} q_{l_1}(\tau)\bar{\Gamma}(l_1,l_2)\sum_{\tau_1=1}^{T} q_{l_2}(\tau_1)\tilde{v}_{\tau_1}\,. \tag{101}$$

Let

$$\bar{v}_l \;=\; \sum_{\tau_1=1}^{T} q_l(\tau_1)\tilde{v}_{\tau_1}\,. \tag{102}$$

Then,

$$\lambda\tilde{v}_\tau \;=\; \sum_{l_1,l_2} q_{l_1}(\tau)\bar{\Gamma}(l_1,l_2)\bar{v}_{l_2}\,. \tag{103}$$

Multiplying by $q_l(\tau)$ and summing over $\tau$, we get

$$\begin{aligned}\lambda\bar{v}_l \;&=\; \lambda\sum_{\tau} q_l(\tau)\tilde{v}_\tau \;=\; \sum_{\tau} q_l(\tau)\sum_{l_1,l_2} q_{l_1}(\tau)\bar{\Gamma}(l_1,l_2)\bar{v}_{l_2}\\ &=\; \sum_{l_1,l_2}\sum_{\tau}(q_l(\tau)q_{l_1}(\tau))\bar{\Gamma}(l_1,l_2)\bar{v}_{l_2}\\ &=\; \sum_{l_1,l_2}\lambda_s(l)\delta_{l,l_1}\bar{\Gamma}(l_1,l_2)\bar{v}_{l_2}\\ &=\; \sum_{l_2}\lambda_s(l)\bar{\Gamma}(l,l_2)\bar{v}_{l_2},\end{aligned} \tag{104}$$

where we have used that

$$\sum_{\tau}(q_l(\tau)q_{l_1}(\tau)) \;=\; \lambda_s(l)\delta_{l,l_1} \tag{105}$$

by the definition of the eigenvalue decomposition of $s = (\Sigma^{\Xi})^{1/2}$. Thus, we can define the big matrix

$$\hat{\Gamma} \;=\; (\lambda_s(l)^{1/2}\lambda_s(l_1)^{1/2}\bar{\Gamma}(l,l_1))_{l,l_1=1}^{r} \;\in\; \mathbb{R}^{(rD)\times(rD)} \tag{106}$$

and redefine

$$\hat{v}_l \;=\; \lambda_s(l)^{-1/2}\bar{v}_l\,. \tag{107}$$

Then, we can rewrite (104) as

$$\lambda\lambda_s(l)^{1/2}\hat{v}_l \;=\; \sum_{l_2}\lambda_s(l)\bar{\Gamma}(l,l_2)\lambda_s(l_2)^{1/2}\hat{v}_{l_2}, \tag{108}$$

Dividing this by $\lambda_s(l)^{1/2}$, we get with $\hat{v}=(\hat{v}_l)_{l=1}^r\in\mathbb{R}^{Dr}$ that

$$\lambda\hat{v} \;=\; \hat{\Gamma}\hat{v}\,. \tag{109}$$

When $r<T$, we need to work with matrices of significantly lower dimensions. Then, we can rewrite (103) as

$$\begin{aligned}\lambda\tilde{v}_\tau \;&=\; \sum_{l_1,l_2}q_{l_1}(\tau)\bar{\Gamma}(l_1,l_2)\bar{v}_{l_2}\\ &=\; \sum_{l_1,l_2}q_{l_1}(\tau)\lambda_s(l_1)^{-1/2}\lambda_s(l_1)^{1/2}\bar{\Gamma}(l_1,l_2)\lambda_s(l_2)^{1/2}\hat{v}_{l_2}\\ &=\; \sum_{l_1}q_{l_1}(\tau)\lambda_s(l_1)^{-1/2}\left(\sum_{l_2}\lambda_s(l_1)^{1/2}\bar{\Gamma}(l_1,l_2)\lambda_s(l_2)^{1/2}\hat{v}_{l_2}\right)\\ &=\; \sum_{l_1}q_{l_1}(\tau)\lambda_s(l_1)^{-1/2}\lambda\hat{v}_{l_1}\end{aligned} \tag{110}$$

Dividing by $\lambda$, we get

$$\tilde{v}_\tau \;=\; \sum_{l_1}q_{l_1}(\tau)\lambda_s(l_1)^{-1/2}\hat{v}_{l_1} \;=\; \sum_{l_1}\hat{q}_{l_1}(\tau)\hat{v}_{l_1} \tag{111}$$

□

**Proof of Proposition 5, MSRR Case**. Next, let us derive expressions for $v_t$ and $\tilde{Q}$ for the MSRR case. We are minimizing the objective function

$$\mathcal{L}_{MSRR}(M)=\|\underbrace{\mathbf{1}}_{T\times 1}-\underbrace{R'\pi}_{T\times 1}\|^2=\frac{1}{T}\sum_{t=1}^{T}(1-R'_{t+1}\underbrace{SM}_{N_t\times 1})^2, \tag{112}$$

where $R\in\mathbb{R}^{N_t\times T}$ is a matrix of next-month returns, $S_t=X_t\otimes\Xi_t\in\mathbb{R}^{N_t\times D}\otimes\mathbb{R}^{Dd}\equiv\mathbb{R}^{N_t\times D^2d}$,

$\check{\lambda} \in \mathbb{R}^{D^2 d}$. Let also $\tilde{S}_t = (R_{t+1}^T X_t) \otimes \Xi_t = F_{t+1} \otimes \Xi_t \in \mathbb{R}^D \otimes \mathbb{R}^{Dd}$. Then,

$$\tilde{S}'\tilde{S} \;=\; \sum_t \underbrace{(F_{t+1}F_{t+1}')}_{D\times D} \otimes \underbrace{(\Xi_t \Xi_t')}_{Dd\times Dd} \in \mathbb{R}^{D^2 d\times D^2 d}. \tag{113}$$

As a sum of $T$ rank-one matrices, $\tilde{S}'\tilde{S} = S'RR'S$ has a rank of at most $T$. Thus, $\tilde{S}'\tilde{S} \in \mathbb{R}^{D^2 d\times D^2 d}$ only contains $T < D^2 d$ non-zero eigenvalues. We denote the corresponding eigenvectors by $V(\theta), \theta = 1, \cdots, T$. As above, we will use Lemma 4 to deal with zero eigenvalues.

We have

$$\begin{aligned}(\tilde{S}'\tilde{S})(x\otimes y) \;&=\; \sum_t (F_{t+1}F_{t+1}') \otimes (\Xi_t\Xi_t')(x\otimes y)\\ =\; \sum_t (c_t F_{t+1}) \otimes \Xi_t,\;\; c_t \;&=\; (F_{t+1}'x)(\Xi_t' y)\,.\end{aligned} \tag{114}$$

Thus, the image of $\tilde{S}'\tilde{S}$ is a subset of the span of $F_{t+1}\otimes \Xi_t$.

Let $V(\theta)$ be the eigenvector of $\tilde{S}'\tilde{S}$ number $\theta \le T$. Since eigenvectors always belong to the image, it admits a representation

$$V(\theta) = \sum_{t=1}^T (v_t(\theta) F_{t+1}) \otimes \Xi_t, \tag{115}$$

where $(v_t(\theta))_{t=1}^T$ is a set of coefficients that are to be determined. We have

$$\begin{aligned}&\tilde{S}'\tilde{S}V(\theta) = \sum_{\tau=1}^T (F_{\tau+1}F_{\tau+1}') \otimes (\Xi_\tau \Xi_\tau') \sum_{t=1}^T (v_t(\theta)F_{t+1}) \otimes \Xi_t\\ &= \sum_{\tau,t=1}^T (F_{\tau+1}\underbrace{F_{\tau+1}' v_t(\theta) F_{t+1}}_{scalar}) \otimes (\Xi_\tau \underbrace{\Xi_\tau' \Xi_t}_{scalar})\\ &= \sum_{\tau=1}^T \underbrace{\Big(\sum_{t=1}^T v_t(\theta) F_{\tau+1}' F_{t+1} \Xi_\tau' \Xi_t\Big)}_{scalar} F_{\tau+1} \otimes \Xi_\tau.\end{aligned} \tag{116}$$

Without loss of generality, we may assume that $F_{\tau+1}\otimes \Xi_\tau$ are linearly independent. Then, the eigenvalue equation

$$\tilde{S}'\tilde{S}V(\theta) \;=\; \lambda(\theta)V(\theta)$$

is equivalent to the system of equations

$$\sum_t v_t(\theta)F'_{\tau+1}F_{t+1}\Xi'_\tau\Xi_t = \lambda(\theta)v_\tau(\theta),\ \tau = 1,\cdots,T\,. \tag{117}$$

Let us define

$$\Sigma^F = (F'_{\tau+1}F_{t+1})_{t,\tau=1}^T \in \mathbb{R}^{T\times T},\ \Sigma^\Xi = (\Xi'_{\tau+1}\Xi_{t+1})_{t,\tau=1}^T \in \mathbb{R}^{T\times T} \tag{118}$$

Let also $v = (v_\tau(\theta))_{\tau=1}^T$. Then, we get the system

$$(\Sigma^F \circ \Sigma^\Xi)v \;=\; \lambda(\theta)v \tag{119}$$

where $(\Sigma^F \circ \Sigma^\Xi)$ is the Hadamard (element-wise) product of the two matrices. Thus, we have just shown that eigenvectors $V(\theta)$ of $\tilde{S}'\tilde{S}$ can be computed in two steps: first, compute eigenvectors of $\Sigma^F \circ \Sigma^\Xi$, and then compute $V(\theta)$ using the formula (115). Note that each $V(\theta)$ is a function of all $v(\theta)$s: to compute $\theta$-th eigenvector $V(\theta)$, we need $\theta$-th entries of all eigenvectors $v(\theta)$. Let us also check the norms:

$$\begin{aligned}\|V(\theta)\|^2 \;&=\; \|\sum_{t=1}^T (v_t(\theta)F_{t+1})\otimes\Xi_t\|^2 \;=\; \sum_{t,\tau=1}^T (v_t(\theta)v_\tau(\theta)F'_{t+1}F_{\tau+1})\otimes\Xi'_t\Xi_\tau \\ &=\; v'(\Sigma^F\circ\Sigma^\Xi)v \;=\; \lambda(\theta)\|v\|^2 \;=\; \lambda(\theta),\end{aligned} \tag{120}$$

so the normalized eigenvectors are

$$\tilde{V}(\theta) \;=\; \lambda(\theta)^{-1/2}\sum_{t=1}^T (v_t(\theta)F_{t+1})\otimes\Xi_t\,. \tag{121}$$

Noting that $\tilde{S}' \in \mathbb{R}^{D^2 d \times T}$, and

$$\tilde{S}'\mathbf{1} \;=\; \sum_{t=1}^{T} F_{t+1} \otimes \Xi_t \,, \tag{122}$$

we get that the optimal $\check{\lambda}$ can be computed as

$$\begin{aligned}\check{\lambda} \;&=\; (T^{-1}\tilde{S}'\tilde{S} + zI)^{-1}T^{-1}\tilde{S}'\mathbf{1} = (\tilde{S}'\tilde{S} + \underbrace{(zT)}_{=:\tilde{z}} I)^{-1}\tilde{S}'\mathbf{1} \\ &= (\tilde{V}\Lambda\tilde{V}' + \tilde{z}^{-1}I)^{-1}\tilde{S}'\mathbf{1} = \Big(\tilde{V}(\tilde{z}I + \Lambda)^{-1}\tilde{V}' + \tilde{z}^{-1}(I - \tilde{V}\tilde{V}')\Big)\tilde{S}'\mathbf{1} \\ &= \Big(\tilde{V}(\tilde{z}I + \Lambda)^{-1}\tilde{V}' + \tilde{z}^{-1}(I - \tilde{V}\tilde{V}')\Big)\sum_{t=1}^{T} F_{t+1} \otimes \Xi_t \\ &= \tilde{z}^{-1}\sum_{t=1}^{T} F_{t+1} \otimes \Xi_t \;+\; \tilde{V}\big((\tilde{z}I + \Lambda)^{-1} - \tilde{z}^{-1}I\big)Q \in \mathbb{R}^{D^2 d},\end{aligned} \tag{123}$$

where $Q = \tilde{V}'\sum_{t=1}^{T} F_{t+1} \otimes \Xi_t \in \mathbb{R}^T$. Each element of the vector $Q$ can be computed as follows:

$$\begin{aligned}Q(\theta) &= \tilde{V}(\theta)'\sum_{t=1}^{T} F_{t+1} \otimes \Xi_t = (\lambda(\theta)^{-1/2}\sum_{\tau} v_\tau(\theta) F_{\tau+1} \otimes \Xi_\tau)'\sum_{t=1}^{T} F_{t+1} \otimes \Xi_t \\ &= \lambda(\theta)^{-1/2}\sum_{\tau=1}^{T}\sum_{t=1}^{T} v_\tau(\theta)\underbrace{(\Xi_\tau'\Xi_t)}_{\in\mathbb{R}}(F_{\tau+1}'F_{t+1}) = \mathbf{1}'(\Sigma^F \circ \Sigma^\Xi)v(\theta) = \lambda(\theta)^{1/2}\mathbf{1}'v(\theta).\end{aligned} \tag{124}$$

Aggregating, let $v \in \mathbb{R}^{T\times T}$ be the matrix of $v(\theta)$ in columns (so, the eigenvector matrix of $\Sigma^F \circ \Sigma^\Xi$, and $\Lambda$ the diagonal matrix with $\lambda(\theta)$ on the diagonal. Then,

$$Q = \Lambda^{1/2}\, v'\mathbf{1} \in \mathbb{R}^T. \tag{125}$$

Notice that $\Lambda$ and $\big((\tilde{z}I + \Lambda)^{-1} - \tilde{z}^{-1}I\big)$ are diagonal matrices. Hence, we can re-write them as vectors of length $T$ and define

$$\tilde{Q}(MSRR) = \big((\tilde{z}I + \Lambda)^{-1} - \tilde{z}^{-1}I\big) \circ \Lambda^{1/2} \circ v'\mathbf{1} \in \mathbb{R}^T. \tag{126}$$

Substituting (121), we get

$$\begin{aligned}\tilde{V}\tilde{Q}(MSRR) &= \sum_{\theta=1}^{T}\lambda(\theta)^{-1/2}\sum_{t=1}^{T}(v_t(\theta)\tilde{Q}(\theta)F_{t+1})\otimes\Xi_t \\ &= \sum_{t=1}^{T}a_tF_{t+1}\otimes\Xi_t,\end{aligned} \tag{127}$$

where

$$\begin{aligned}a_t &= \sum_{\theta=1}^{T}\lambda(\theta)^{-1/2}v_t(\theta)\tilde{Q}(\theta) \\ &= \sum_{\theta=1}^{T}\lambda(\theta)^{-1/2}v_t(\theta)\lambda(\theta)^{1/2}\big((\tilde{z}I+\lambda(\theta))^{-1}-\tilde{z}^{-1}I\big)\mathbf{1}'v(\theta)\end{aligned} \tag{128}$$

Thus,

$$\check{\lambda} = \sum_{t=1}^{T}(\tilde{z}^{-1}F_{t+1})\otimes\Xi_t + \underbrace{\tilde{V}}_{D^2d\times T}\underbrace{\tilde{Q}(MSRR)}_{T\times 1} = \sum_{t=1}^{T}(\tilde{z}^{-1}+a_t)F_{t+1}\otimes\Xi_t \tag{129}$$

The Proof of Proposition 5, MSRR Case is complete.

□

**Proof of Theorem 3**. We have already shown in the Proof of Proposition 5 that (49) holds for MSE and MSRR cases, and we have derived expressions for $q_t(type)$ for any $t = 1, ..., T$. It remains to prove (50). Thus, substituting

$$\check{\lambda}_{type} = \sum_{t=1}^{T}q_t(type)\otimes\Xi_t \;\in\mathbb{R}^{D^2d}. \tag{130}$$

into the optimal portfolio

$$\pi_t(type) \;=\; S_t'\check{\lambda}_{type}, \tag{131}$$

we get

$$\begin{aligned}
\pi_t(type)' R_{t+1} \;&=\; \check{\lambda}_{type}' S_t R_{t+1} \\
&=\; \check{\lambda}_{type}' F_{t+1} \otimes \Xi_t \\
&=\; \sum_{\tau=1}^{T} F_{t+1}' q_\tau (\Xi_t' \Xi_\tau) \\
&=\; F_{t+1}' \sum_{\tau=1}^{T} q_\tau(type)\, \Sigma^{\Xi}(t,\tau)\,.
\end{aligned} \tag{132}$$

The proof is complete. $\square$

## D Model Optimization

### D.1 Linear Attention

**Algorithm 1** Return on the linear attention portfolio at time $t+1$

```
Require: Covariances $\Xi_\tau \in \mathbb{R}^{Dd}$ and linear factors $F_{\tau+1} \in \mathbb{R}^D$ for $\tau = t - N_{rol} + 1, ..., t-1$.
 1: Compute matrices $\Sigma^F = (F'_{\tau_1+1} F_{\tau_2+1})_{\tau_1,\tau_2=1}^{N_{rol}} \in \mathbb{R}^{N_{rol} \times N_{rol}}$ and $\Sigma^\Xi = (\Xi'_{\tau_1} \Xi_{\tau_2})_{\tau_1,\tau_2=1}^{N_{rol}} \in \mathbb{R}^{N_{rol} \times N_{rol}}$.
 2: Find the vector of eigenvalues $D \in \mathbb{R}^{N_{rol}}$ and the matrix of eigenvectors $v \in \mathbb{R}^{N_{rol} \times N_{rol}}$ of $\Sigma^F \circ \Sigma^\Xi \in \mathbb{R}^{N_{rol} \times N_{rol}}$.
 3: Divide each eigenvector $v_k$, $k = 1, ..., N_{rol}$ by the square root of the corresponding eigenvalue $\sqrt{\lambda_k}$. This lets us recover the normalised eigenvectors $V_k$, $k = 1, ..., N_{rol}$, of the parent matrix $\tilde{S}'\tilde{S} = VDV'$ using the formula $V_k = \sum_{\tau=1}^{N_{rol}} v_{k,\tau} F_{\tau+1} \otimes \Xi_\tau$. Notice that $v_{k,\tau}$s are scalars.
 4: Compute vector $Q = D \circ (v' \mathbf{1}_{N_{rol} \times 1}) \in \mathbb{R}^{N_{rol} \times 1}$.
 5: For each $z_k \in z \in \mathbb{R}^{N_z}$, compute vectors $\tilde{Q}_k = \left((z_k \mathbf{1}_{N_{rol} \times 1} + D)^{-1} - z_k^{-1}\right) \circ Q$. Then, stack $\tilde{Q}_k \in \mathbb{R}^{N_{rol} \times 1}$ by columns, getting $\tilde{Q} \in \mathbb{R}^{N_{rol} \times N_z}$. This way, we set the ground for fitting variables of interest ($M^*$, $W^*(M)$ and $b^*(z, M)$) simultaneously for a grid of $z_k$.
 6: Now, denote $v_\tau \in \mathbb{R}^{N_{rol}}$ the $\tau$-th *row* of the normalised eigenvector matrix $v \in \mathbb{R}^{N_{rol} \times N_{rol}}$, there are $N_{rol}$ such $v_\tau$s. Define the outer products $\tilde{v}_\tau := F_{\tau+1} v'_\tau \in \mathbb{R}^{D \times N_{rol}}$.
 7: for $\tau \in t - N_{tol}, ..., t-1$ do
 8:     Compute $h_{k,\tau} = z_k^{-1} F_{\tau+1} \in \mathbb{R}^D$ for the grid of $z_k$ and stack $h_{k,\tau}$ by columns into $h_\tau \in \mathbb{R}^{D \times N_z}$, then compute $q_\tau := h_\tau + \tilde{v}_\tau \tilde{Q} \in \mathbb{R}^{D \times N_z}$.
 9:     Normalise $q_\tau$ by $N_{rol}$, i.e., do $q_\tau = \frac{q_\tau}{N_{rol}}$. This normalises the matrix $M^*$ (we can not normalise $M^*$ directly because we do not compute $M^*$ explicitly).
10: end for
11: for $z_k \in z \in \mathbb{R}^{N_z}$ do
12:     Now that we have $q_{k,\tau} \in \mathbb{R}^D$ for all $z_k \in z$ and all $\tau = t - N_{tol}, ..., t-1$, i.e., we have a 3-dimensional tensor $q \in \mathbb{R}^{D \times N_z \times N_{rol}}$, we split it across $z$ dimension into $N_z$ vectors $q_k \in \mathbb{R}^{D \times N_{rol}}$. Compute $\Sigma_k^q = q'_k q_k \in \mathbb{R}^{N_{rol} \times N_{rol}}$.
13:     Compute the inner product of $M^*$s without explicitly computing $M^*$: $W_k^{*'} W_k^* = \sum_{\tau_1,\tau_2=t-N_{tol}}^{t-1} \Sigma_{k,\tau_1\tau_2}^q \, \Xi_{\tau_1}^{D \times d'} \, \Xi_{\tau_2}^{D \times d}$. Here, $\Xi_{\tau_1}^{D \times d'} \Xi_{\tau_2}^{D \times d} \in \mathbb{R}^{d \times d}$ and $\Sigma_{k,\tau_1\tau_2}^q$ is a scalar.
14:     Compute eigenvector matrix $m_k \in \mathbb{R}^{d \times d}$ of $W_k^{*'} W_k^* \in \mathbb{R}^{d \times d}$. Each column of $m$ corresponds to some attention head.
15: end for
16: for $h \in [1, 2, 3, 4, 8, 16, 32...]$ and $z_k \in z \in \mathbb{R}^{N_z}$ do
17:     Denote $m_{k,h} \in \mathbb{R}^{d,h}$ the matrix of eigenvectors $m_k$ with $h$ columns that correspond to the largest eigenvalues of $W_k^{*'} W_k^*$. Compute $\tilde{\Xi}_{\tau,k,h} = \Xi_\tau^{D \times d} m_{k,h} m'_{k,h} \in \mathbb{R}^{D \times d}$.
18:     Compute the $t+1$ out-of-sample return $R_{t+1,k,h}^{pf}$ on the Linear MSRR attention portfolio $R_{t+1,k,h}^{pf} = R_{t+1} \tilde{S}_{t+1} M_{k,h}^* = \sum_{\tau=t-N_{rol}}^{t-1} (\Xi'_t \tilde{\Xi}_{\tau,k,h})(F'_{t+1} q_{k,\tau})$, where $\Xi'_t \tilde{\Xi}_{\tau,k,h}$ and $F'_{t+1} q_{k,\tau}$ are scalars.
19: end for
20: We end up with $N_z \cdot N_h$ out-of-sample returns on linear attention portfolio $R_{t+1,k,h}^{pf}$ for a grid of penalty parameters $z_k$ and number of attention heads $h$.
```

## D.2 Transformer Optimization

**Algorithm 2** Transformer-based trading strategy

```
Require: Panel data of stocks characteristics $X_t \in \mathbb{R}^{N_t \times D}$, returns $R_{t+1} \in \mathbb{R}^{N_t}$, rolling window period $\tau$, a learning rate $\eta$, number of epochs $e$, and a non-linear transformer model $\mathcal{T}^{(K)}$, which comprises of $K$ transformer-blocks. Each transformer-block $\mathcal{T}^{(k)}$ is composed of a multi-head attention unit $\mathcal{A}$, a feed-forward network $\mathcal{F}$, and an objective function $\in \{\text{MSRR}, \text{MSE}\}$.
1: Denote by $\Theta_{\mathcal{T}}$ the set of learnable parameters.
2: for epoch $\in e$ do
3:     for month $\in \tau$ do
4:         if loss == MSRR then
5:             $\mathcal{L}_\Theta \leftarrow (1 - \mathcal{T}^{(K)}(X_t)' R_{t+1} \lambda)^2$.
6:         else
7:             $\mathcal{L}_\Theta \leftarrow \|R_{t+1} - \mathcal{T}^{(K)}(X_t)'\lambda\|^2$.
8:         end if
9:         $\Theta_{s+1} \leftarrow \Theta_s - \eta \nabla \mathcal{L}$.
10:    end for
11: end for
```

Table 7: **Portfolio Transformer Performance by Size Groups**

| | **BSV** | **DKKM** | **Lin. Attn.** | **MLP** | **Transformer** |
|---|---|---|---|---|---|
| | | | **All** | | |
| Sharpe ratio | 3.60 | 3.91 | 3.89 | 4.31 | 4.57 |
| HJD | 0.15 | 0.13 | 0.14 | 0.13 | 0.09 |
| | | | **Micro** | | |
| Sharpe ratio | 3.46 | 3.68 | 3.77 | 4.02 | 4.42 |
| HJD | 0.16 | 0.15 | 0.16 | 0.14 | 0.11 |
| | | | **Small** | | |
| Sharpe ratio | 2.23 | 2.36 | 2.40 | 2.79 | 3.23 |
| HJD | 0.31 | 0.29 | 0.29 | 0.23 | 0.19 |
| | | | **Large** | | |
| Sharpe ratio | 1.37 | 1.51 | 1.52 | 1.47 | 2.70 |
| HJD | 0.48 | 0.45 | 0.45 | 0.45 | 0.26 |
| | | | **Mega** | | |
| Sharpe ratio | 1.15 | 1.14 | 1.05 | 1.18 | 1.84 |
| HJD | 0.54 | 0.54 | 0.55 | 0.53 | 0.41 |

Table 8: **Portfolio Transformer Performance by Size Groups**

| | **BSV** | **DKKM** | **Lin. Attn.** | **MLP** | **Transformer** |
|---|---|---|---|---|---|
| | | | **All** | | |
| Sharpe ratio | 3.60 | 3.91 | 3.89 | 4.31 | 4.57 |
| HJD | 0.15 | 0.13 | 0.14 | 0.13 | 0.09 |
| | | | **Micro** | | |
| Sharpe ratio | 3.46 | 3.68 | 3.77 | 4.02 | 4.42 |
| HJD | 0.14 | 0.13 | 0.13 | 0.13 | 0.10 |
| | | | **Small** | | |
| Sharpe ratio | 2.23 | 2.36 | 2.40 | 2.79 | 3.23 |
| HJD | 0.25 | 0.24 | 0.24 | 0.17 | 0.15 |
| | | | **Large** | | |
| Sharpe ratio | 1.37 | 1.51 | 1.52 | 1.47 | 2.70 |
| HJD | 0.28 | 0.26 | 0.26 | 0.25 | 0.16 |
| | | | **Mega** | | |
| Sharpe ratio | 1.15 | 1.14 | 1.05 | 1.18 | 1.84 |
| HJD | 0.25 | 0.25 | 0.27 | 0.23 | 0.19 |

Figure 7: **Sharpes and HJDs for all strategies, for different size groups.**

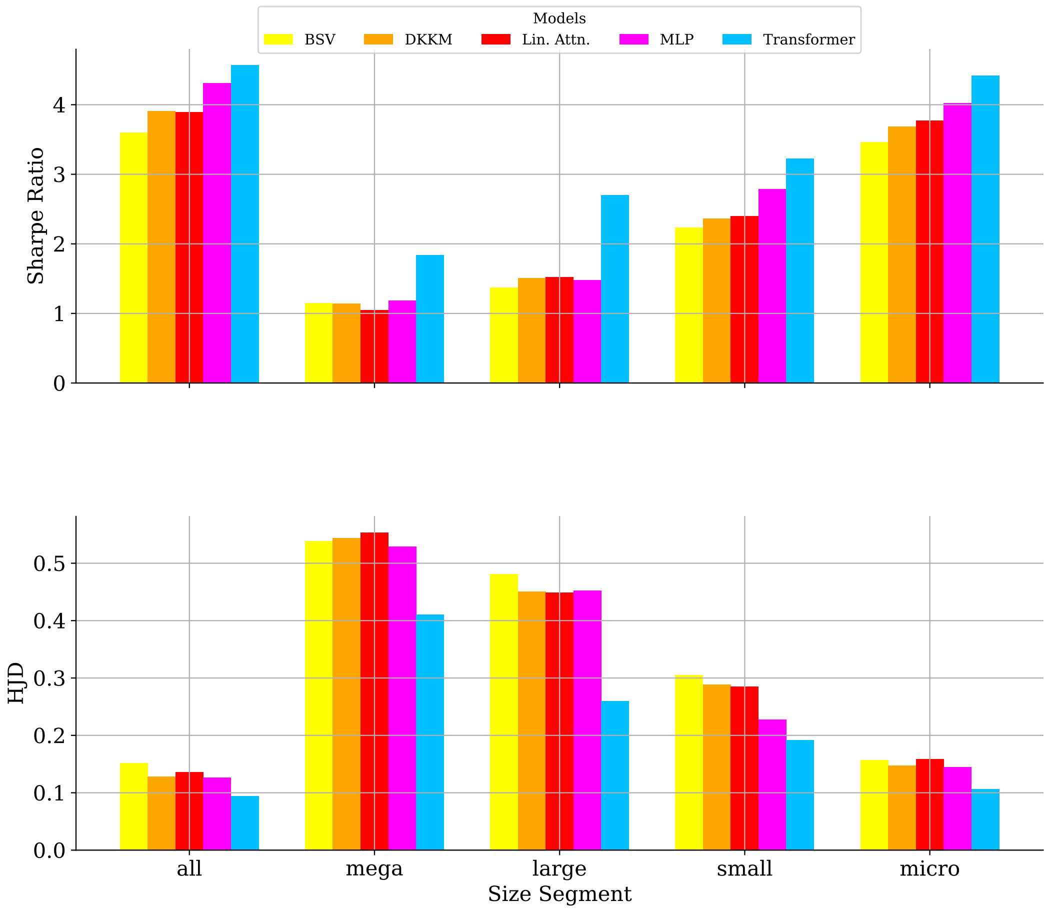
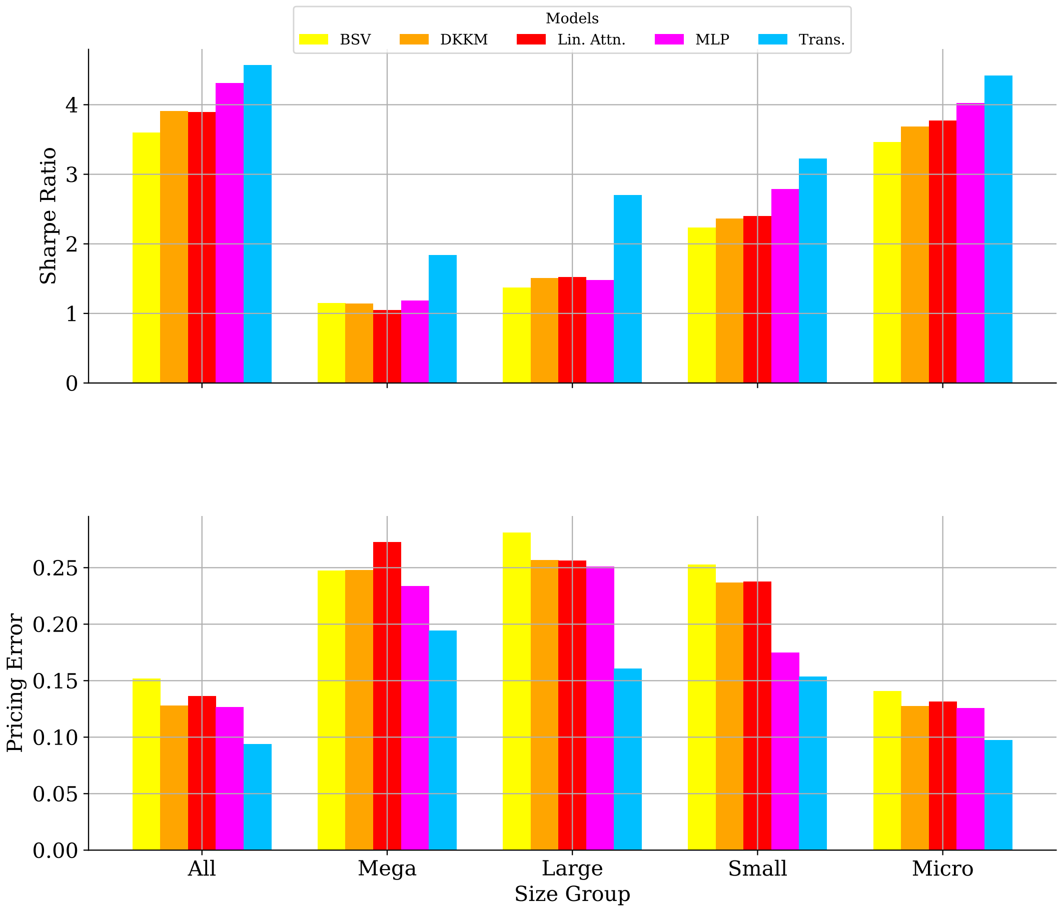


Figure 8: **Sharpes and HJDs for all strategies, for different size groups.**


Models
BSV
DKKM
Lin. Attn.
MLP
Trans.
Sharpe Ratio
Pricing Error
All
Mega
Large
Small
Micro
Size Group