Machine Learning and the Implementable Efficient Frontier

Theis Ingerslev Jensen, Bryan Kelly, Semyon Malamud, and Lasse Heje Pedersen*

This version: February 12, 2023

Abstract

We propose that investment strategies should be evaluated based on their net-of-trading-cost return for each level of risk, which we term the "implementable efficient frontier." While numerous studies use machine learning return forecasts to generate portfolios, their agnosticism toward trading costs leads to excessive reliance on fleeting small-scale characteristics, resulting in poor net returns. We develop a framework that produces a superior frontier by integrating trading-cost-aware portfolio optimization with machine learning. The superior net-of-cost performance is achieved by learning directly about portfolio weights using an economic objective. Further, our model gives rise to a new measure of "economic feature importance."

Keywords: asset pricing, machine learning, transaction costs, economic significance, investments
JEL Codes: C5, C61, G00, G11, G12

*Jensen is at Copenhagen Business School. Kelly is at Yale School of Management, AQR Capital Management, and NBER; www.bryankellyacademic.org. Malamud is at Swiss Finance Institute, EPFL, and CEPR, and is a consultant to AQR. Pedersen is at AQR Capital Management, Copenhagen Business School, and CEPR; www.lhpedersen.com. We are grateful for helpful comments from Cliff Asness, Jules van Binsbergen, Darrell Duffie, Marc Eskildsen, Markus Ibert, Leonid Spesivtsev, and seminar participants at Copenhagen Business School and the NBER Big Data and Securities Markets Conference, Fall 2021. 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. Semyon Malamud gratefully acknowledges the financial support of the Swiss Finance Institute and the Swiss National Science Foundation, Grant 100018_192692. All four authors appreciate the financial support of INQUIRE Europe.

This paper studies how security information can be used for portfolio selection in a flexible and realistic setting with transaction costs. The goal is thus both to provide a powerful tool for portfolio choice and to shed new light on which security characteristics are economically important drivers of asset pricing.

The financial machine learning (ML) literature provides a flexible framework to combine several characteristics into a single measure of overall expected returns (e.g. Gu et al., 2020). The same literature documents the relative "feature importance" of different return prediction characteristics (e.g. Chen et al., 2021). These findings suggest that the prediction success of ML methods is often driven by short-lived characteristics that work well for small and illiquid stocks (e.g. Avramov et al., 2021), suggesting that they might be less critical for the real economy (e.g. Van Binsbergen and Opp, 2019). The high transaction costs of portfolio strategies based on ML imply that these strategies are difficult to implement in practice and, more broadly, raise questions about the relevance and interpretation of the predictability documented in this literature. Do ML-based expected return estimates merely tell us about mispricings that investors don't bother to arbitrage away because the costs are too large, the mispricing too fleeting, and the relevant stocks too small to matter? Or, do trading-cost-aware ML-based predictions also work for large stocks over significant periods and in a valuable way for large investors, thus providing information about their preferences?

This paper seeks to generate economically useful predictions. We are interested in deriving ML-driven portfolios that can be realistically implemented by market participants with a substantial fraction of aggregate assets under management, such as large pension funds or other professional asset managers. If a strategy is implementable at scale, then the predictive variables that drive such portfolio demands are informative about the equilibrium discount rates of major companies (Koijen and Yogo (2019)).

While ML with transaction costs is challenging to attack with brute force, we deliver a tractable solution through the help of economic theory. Specifically, we show how to integrate the ML problem into a generalized version of the optimal portfolio selection framework of Gârleanu and Pedersen (2013). The main thrust of our approach is to feed the objective function explicit knowledge of implementability, so it knows to search for perhaps subtle but "usable" predictive patterns while discarding more prominent but costly predictive patterns. We develop an ML method to produce optimal portfolios while considering realistic frictions from transaction costs of the securities it trades. Our solution also gives rise to a new

measure of "economic feature importance" that captures which characteristics provide the most investment value; in other words, which characteristics contribute the most to the overall portfolio's risk-adjusted returns after trading costs.

Our approach generalizes Gârleanu and Pedersen (2013) in three important ways. First, while Gârleanu and Pedersen (2013) assume that expected price changes are linear functions of a set of signals, we allow expected returns to be a fully general function of the signals, opening the door for flexible non-linear ML. Second, our setting is based on stationary returns, not stationary price changes, solving a vexing problem in the portfolio choice literature and providing a new coherence to empirical analysis over long horizons.[1] Third, while Gârleanu and Pedersen (2013) take the data generating process as given, we integrate the estimation process into the optimization process via ML, showing our method's practical and empirical power.

To understand the difference between our approach and the typical use of ML in finance, note that the latter takes a two-step approach: First, find a function of characteristics that predicts gross returns, and second, use the resulting forecasts to build portfolios. This typical approach abstracts from transaction costs and turnover, and the resultant investment strategies produce negative returns net of transaction costs.

Our approach builds transaction costs directly into the objective function, thus ensuring that the algorithm learns about usable predictability. One element of usable predictability is that it is relevant for large stocks with low transaction costs. Another essential element is alpha decay, that is, how persistent a predictor is. With transaction costs, you will likely own whatever you buy today for a while because the trading costs encourage you only slowly to enter or exit positions. Naturally, understanding the expected return both now and further into the future is relevant.

Empirically, the optimal ML predictor of near-term returns is indeed different from the optimal ML predictor of returns far into the future. In other words, if f_h is the function that best predicts returns h months into the future, $E_t[r_{i,t+h}] = f_h(s_{it})$, then this function is

[1] Gârleanu and Pedersen (2013) show that the portfolio problem simplifies by looking at numbers of shares and price changes because this sidesteps the issue of portfolio growth that has plagued the literature. The portfolio growth is the issue that, if you put 10% of your wealth in IBM stock today, then you will not have 10% of your wealth in IBM next period before trading – because of the price change of IBM and other stocks. Working with the number of shares sidesteps this issue (the number of shares only changes when you trade). Still, the cost is that profit is equal to shares times price changes, so the model cannot be specified in terms of percentage returns, making empirical analysis difficult. We have found a way to work with empirically relevant units and preserve tractability via an approximately optimal solution.

different across h. Given that the standard ML approach uses only f_1, we see that it misses the information contained in f_h at other horizons, $h > 1$.

One way to implement our approach is to forecast returns across many time horizons $h = 1, 2, ...$, then to use all of the predictive functions, f_h, appropriately discounted given risk, risk aversion, and the form of the trading cost function. However, this approach requires a highly complex ML formulation to accommodate all predictive functions simultaneously. Using this approach either leads to serious technical challenges (like massive computing costs) or requires cutting essential corners.

Our preferred approach instead learns directly about portfolio weights instead of expected returns.[2] This simple approach delivers an essentially closed-form solution to the highly complex portfolio problem in a single step!

To evaluate the performance of our method, we propose that portfolio choice methods and ML predictions are assessed based on the net-of-cost investment opportunities they produce. Indeed, a fundamental insight in portfolio choice is that investors can depict their investment opportunity set as all the achievable combinations of risk and expected return, giving rise to the "efficient frontier" depicted in most finance textbooks as the tangency line to the hyperbola generated by risky investments. The textbook frontier is drawn in a frictionless setting that abstracts from trading costs, but real-world investors care about their net return. What does the frontier look like when we take trading costs into account?

Panel A of Figure 1 illustrates frontiers for various methods that we study. The baseline for comparison is the cost-agnostic Markowitz-ML solution and the hyperbola of risky investments – both in *gross* terms– that is, our implementation of the textbook frontier using ML. Specifically, these portfolios use ML to build stock-level expected returns, use the academic analog of Barra to build a covariance matrix, and then form ex-ante efficient portfolios from these two inputs. Figure 1.A plots the portfolios' realized out-of-sample performance. As seen in the figure, while the Markowitz solution is tangent to the hyperbola in a textbook analysis with known means and variances, the Markowitz solution is not precisely tangent in our out-of-sample analysis. In any event, the Markowitz portfolio performs very well out-of-sample, delivering a Sharpe ratio of roughly 2.0 per annum. But this is in *gross* terms. The portfolio's turnover is enormous and the textbook frontier is non-implementable in practice.

[2]This idea builds on Brandt et al. (2009) who propose using portfolio weights that are parametric (linear) functions of characteristics in a setting without trading costs. We extend their idea to handle the much more complex portfolio problem with transaction costs and by using ML to learn about the optimal weights.

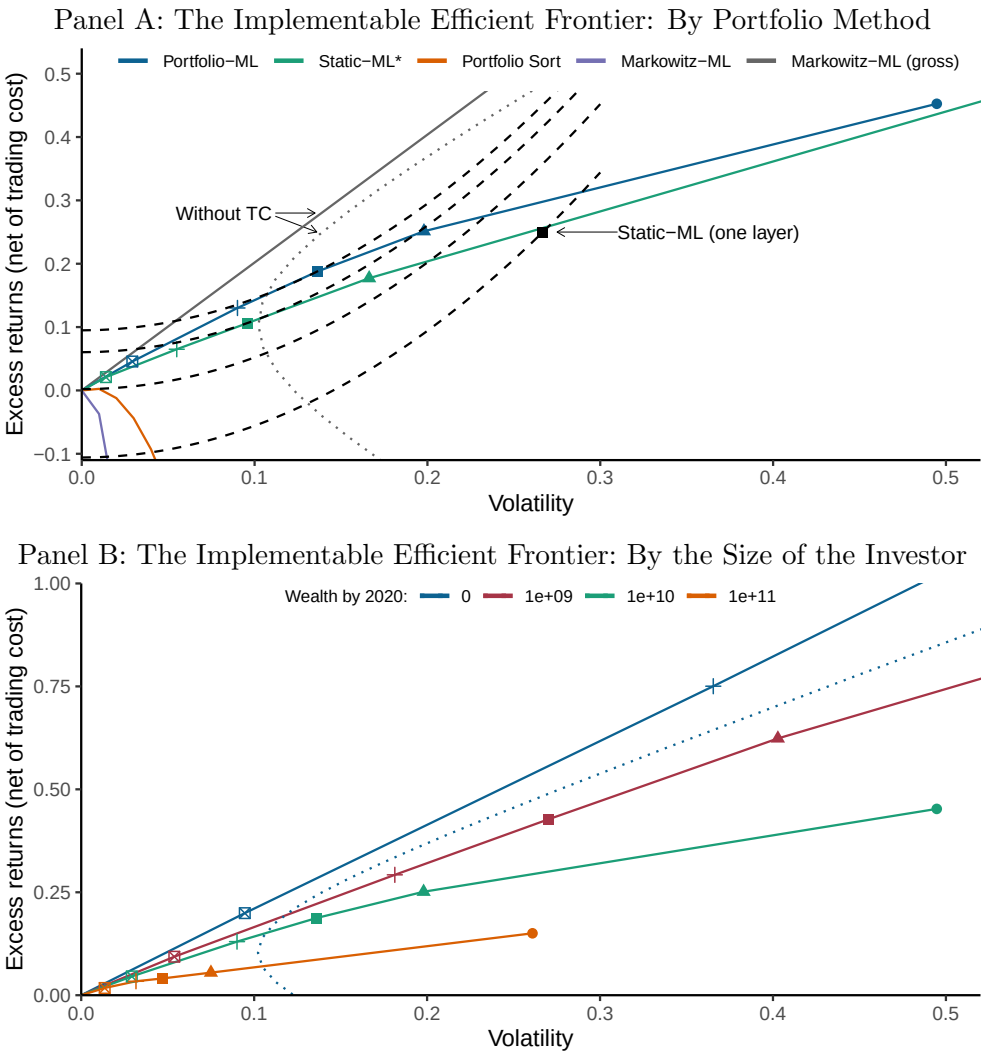

Figure 1: The Implementable Efficient Frontier: Risk vs. Return Net of Trading Costs

Note: Panel A shows the implementable efficient frontier for different portfolio methods with a wealth of \$10B by 2020. The dashed lines show indifference curves. The dotted hyperbola is the mean-variance frontier of risky assets without trading costs, implemented by estimating risk and expected return separately, out-of-sample. The grey line is the Markowitz-ML efficient frontier before trading cost. After trading costs, Markowitz-ML and portfolio sort have downward bending frontiers as these methods are not implementable. Static-ML produces a positive net Sharpe but negative utility, but it works well with an extra tuning layer, denoted Static-ML*. Our Portfolio-ML works significantly better out-of-sample. Panel B shows the implementable efficient frontier at different wealth levels. The dotted hyperbola is the same mean-variance frontier as in Panel A. The blue line is the optimal portfolio of risky and risk-free assets for an investor with zero wealth, corresponding to no trading costs, estimated using our Portfolio-ML method for different relative risk aversions. The blue line would be the tangency line to the hyperbola in a standard in-sample textbook analysis, but it is not exactly tangent out-of-sample. The lower lines illustrate the mean-variance frontiers with larger wealth levels, also estimated using Portfolio-ML. In both panels, the relative risk aversions are 1 (circle), 5 (triangle), 10 (square), 20 (plus), and 100 (boxed cross) and the sample period is 1981-2020. Further details are provided in Section 5.2.

The other lines in Figure 1 show our concept of an "implementable efficient frontier," that is, the achievable combinations of risk and expected return, net of trading costs. Focusing first on the Markowitz portfolio, we see that its net-of-cost, implementable frontier immediately dives into negative expected return territory as soon as it moves away from a 100% risk-free allocation, as seen in the bottom curve in Panel A. The shape of the implementable efficient frontier may be surprising: Whereas the textbook frontier is a straight line when increasing the allocation to the risky securities while reducing the risk-free allocation (or applying leverage), the true implementable frontier bends down because larger positions incur larger transaction costs. Said differently, we show that the net-of-cost Sharpe ratio declines along the implementable efficient frontier.

To understand the source of the problem for Markowitz-ML at a deeper level, the feature importance of this portfolio reveals the culprit: excessive reliance on fleeting small-scale characteristics (e.g., 1-month reversal for small stocks), which bear high turnover, high trading costs, and result in poor net returns. Further, Panel A of Figure 1 also shows that a standard "portfolio sort" used in the literature is also not implementable.

The difficulty of the standard portfolios from the literature is noteworthy. Still, it is also interesting to compare our approach to a more sophisticated alternative that may be used by some large investors. This sophisticated alternative first uses ML to build expected returns (agnostic of trading costs), then in an additional second-stage optimization, takes transaction costs into account to build portfolios. This "Static-ML" approach delivers a positive net Sharpe but a lower utility than putting all the money in the risk-free asset, as seen from the indifference curves in Figure 1.A.

To create a more difficult benchmark to beat for our preferred method (Portfolio-ML), we further enhance the standard approach by adding several extra hyper-parameters that improve performance by adjusting its scale in various ways. We refer to this approach as Static-ML*, where the "*" indicates that we use an extra tuning stage. Static-ML* performs well, delivering high utility as seen in Figure 1.

Despite that Static-ML* is a sophisticated multi-stage approach that is much more highly parameterized than our Portfolio-ML method, our Portfolio-ML method nevertheless significantly outperforms Static-ML*. To understand why Static-ML* underperforms our approach, note that the first-stage ML procedure produces expected returns dominated by short-term signals. This method does not consider which predictors are persistent and which have quick

alpha decay. The second-stage optimization reduces turnover, especially for small stocks, which leads to a much better performance than portfolios that ignore transaction costs. However, this static approach can nevertheless be improved by recognizing the dynamic nature of the portfolio using a method that is sensitive to how expected returns vary across several return horizons (i.e., alpha decay).

In other words, our Portfolio-ML method delivers out-of-sample net-of-cost returns that outperform a highly sophisticated alternative by roughly 20% in Sharpe ratio terms and 60% in utility terms. Further, the feature importance across signals changes when we consider transaction costs. While short-term reversal signals highly influence naive methods, our method seeks to optimally blend return predictability across multiple future horizons, especially for liquid stocks. This leads to value and quality earning the highest feature importance.

Panel B of Figure 1 draws the implementable efficient frontier using our Portfolio-ML at different levels of wealth or asset under management (AUM). Interestingly, while the textbook efficient frontier is the same for all investors, the implementable efficient frontier depends on the investor's size via the implied trading costs. Indeed, larger investors face worse (i.e., lower) efficient frontiers that "cut into" the hyperbola.

As an interesting benchmark, the top line shows the Portfolio-ML strategy when trading costs are nearly zero since the investor has an AUM near zero. This implementable frontier is obviously good due to the near-zero trading costs, but we note that such sophisticated ML-based trading is hardly feasible for small investors in the real world.[3]

The frontier at each wealth level shows that the set of optimal implementable portfolios is strictly worse for higher AUM investors. This degradation happens for two reasons. First, trading a larger portfolio incurs higher market impact cost. However, the investor can partly mitigate direct transaction costs by trading less, but this increases opportunity costs. Indeed, an investor with larger AUM internalizes price impact from their trades, and this leads the investor to tilt away from highly predictive but costly-to-trade stocks and signals. Large cost-aware investors opt to forego some predictability in order to hold trading costs at bay.

[3]The strategies we develop would be challenging to implement for small investors as they require real-time data on many characteristics across more than a thousand stocks, computation of predictive signals, implementation of ML models, and infrastructure for continually updating and trading these models. Hence, the methods are most relevant for investors large enough to have a staff that can perform these tasks, but, given such capabilities, the implementable investment opportunity set is worse for larger AUM as shown in Figure 1.B.

Our paper is related to several large literatures. The first applies machine learning methods to enhance return prediction and enhance portfolio performance, including Gu et al. (2020), Freyberger et al. (2020), Chen et al. (2021), Kelly et al. (2019), Gu et al. (2021), Jensen et al. (2022), and Han et al. (2021) in US equity markets; Choi et al. (2021), Leippold et al. (2022), and Cakici and Zaremba (2022) in international equity markets; and Kelly et al. (2022), Bali et al. (2022), and Bali et al. (2021) in bond and derivative markets. Recent empirical papers point out that trading strategies based on these factors and the literature on machine learning in asset pricing cited above involve large transaction costs in practice. This literature includes Li et al. (2020), Chen and Velikov (2021), and Detzel et al. (2021). Motivated by these papers, we develop a flexible portfolio optimization method that lends itself to ML while directly confronting the implementability challenge and explicitly incorporating transaction costs into the ML-based portfolio optimization problem.

The second related literature extends the frictionless paradigm of Markowitz (1952) to study portfolio choice in the presence of transaction costs. Constantinides (1986) and Davis and Norman (1990) analyze settings with a single security, where returns are not predictable. Balduzzi and Lynch (1999) and Lynch and Balduzzi (2000) numerically study single-asset trading with predictable returns and transaction costs. Gârleanu and Pedersen (2013) derive an explicit portfolio solution with multiple assets with predictable returns and transaction costs when returns are driven by a factor model. Gârleanu and Pedersen (2016) extend this to more general dynamics in continuous time and Collin-Dufresne et al. (2020) extend the model to include different liquidity regimes. Our contribution is to derive optimal portfolio rules based on stationary dynamics of returns (rather than dynamic programming with stationary price changes, as in the literature) and fully general functional forms for return predictability while incorporating an arbitrarily large set of predictors.

In summary, we provide a theoretical bridge between portfolio optimization and machine learning with powerful empirical results.

1 Model and the Implementable Efficient Frontier

We consider an economy with N securities traded in discrete time indexed by $t = ..., -2, -1, 0, 1, 2, ...$ The return of asset n from time t to $t+1$ is given by $r^f_{t+1} + r_{n,t+1}$, where r^f_{t+1} is risk-free rate and $r_{n,t+1}$ is the asset's excess return. The vector of all assets' excess returns

is denoted r_{t+1}.

An investor observes several characteristics (or signals) for each security, denoted $s_{n,t} \in \mathbb{R}^K$, for example, each asset's valuation ratio, momentum, size, and so on. The characteristics of all assets are collected in the matrix $s_t \in \mathbb{R}^{N\times K}$, and we assume that s_t and r_t are stationary and ergodic. The signals s_t fully characterize the investor's information about returns in a sense that

$$r_{t+1} = \mu(s_t) + \varepsilon_{t+1} \tag{1}$$

where the conditional mean $\mu(s_t) = E_t[r_{t+1}]$ and variance $\Sigma(s_t) = \text{Var}_t[r_{t+1}] = \text{Var}_t[\varepsilon_{t+1}]$ are bounded Borel-measurable functions of s_t with Σ being positive definite.

The investor can be seen as a professional asset manager, such as a hedge fund. The investor has wealth or assets under management (AUM) given by w_t at time t. The asset manager's AUM grows at a stochastic rate, g_t^w, so that $w_{t+1} = w_t(1 + g_{t+1}^w)$, which generally depends on performance and on how clients take money in and out of the fund, as specified in Section 1.2. The investor must choose how much capital, $\pi_{n,t}^{\$}$, to invest in each asset or, equivalently, choose the fraction of the capital invested in each asset, $\pi_{n,t} = \pi_{n,t}^{\$}/w_t$. This portfolio choice implies a dollar profit before transaction costs at time $t+1$ of

$$\text{dollar profit before t-costs}_{t+1} = (r_t^f + r_{t+1})'\pi_t^{\$} + (w_t - 1_N'\pi_t^{\$})r_t^f = w_t(r_t^f + r_{t+1}'\pi_t) \tag{2}$$

where $w_t - 1_N'\pi_t^{\$}$ is the amount of money in the risk-free money market account, and 1_N is a vector of ones. The corresponding return before trading costs, net of the risk-free rate, is

$$r_{t+1}^{\pi,gross} = \frac{\text{dollar profit before t-costs}_{t+1}}{w_t} - r_t^f = r_{t+1}'\pi_t \tag{3}$$

1.1 Trading Costs, Net Returns, and Portfolio Growth

The investor faces transaction costs due to her market impact. Specifically, suppose the investor chooses to trade dollar values given by $\tau_t \in \mathbb{R}^N$ at any time t. This trade leads to a market impact of $\frac{1}{2}\Lambda_t\tau_t$, where $\Lambda_t \in \mathbb{R}^{N\times N}$ is a multivariate version of "Kyle's Lambda," which is symmetric and positive semi-definite such that transaction costs are non-negative.

It may vary as a function of time and the state s_t of the market.[4] The resulting transaction cost is the product of the trade size and its market impact, that is,

$$\text{dollar t-costs}_t = \frac{1}{2}\tau_t' \Lambda_t \tau_t \,. \tag{4}$$

To determine the trade size, note that the dollar position $\pi^{\$}_{n,t-1}$ bought at time $t-1$ has grown in value to $\pi^{\$}_{n,t-1}(1 + r^f_t + r_{n,t})$. The old dollar position has grown due to the return on the asset (or, said differently, the price change). Hence, the vector of all dollar trade sizes is

$$\begin{aligned} \tau_t &= \pi^{\$}_t - \text{diag}(1 + r^f_t + r_t)\pi^{\$}_{t-1} \\ &= w_t \pi_t - w_{t-1}\,\text{diag}(1 + r^f_t + r_t)\pi_{t-1} \\ &= w_t \left(\pi_t - g_t \pi_{t-1}\right) \,, \end{aligned} \tag{5}$$

where $\text{diag}(v)$ is a diagonal matrix with the vector v in the diagonal and

$$g_t = \text{diag}\left(\frac{1 + r^f_t + r_t}{1 + g^w_t}\right) \tag{6}$$

is the growth of portfolio weights at time t. Combining equations (2)–(6), we see that the return as a fraction of wealth in excess of the risk-free rate and trading costs is

$$r^{\pi,net}_{t+1} = r^{\pi,gross}_{t+1} - TC^{\pi}_t = r'_{t+1}\pi_t - \frac{w_t}{2}\left(\pi_t - g_t\pi_{t-1}\right)' \Lambda_t \left(\pi_t - g_t \pi_{t-1}\right) \,. \tag{7}$$

where $TC^{\pi}_t = \frac{\text{dollar t-costs}_t}{w_t}$. The portfolio's return naturally depends on the portfolio weights, π, but it also depends on the wealth w_t even though the return is measured in percent of the wealth. This is because trading costs increase by the square of wealth, such that a larger wealth leads to lower portfolio returns after transaction costs. Said differently, a larger investor has a larger market impact (for the same portfolio weights π), thus receiving lower net returns.

[4] The symmetry of Λ is without loss of generality since, if we start with non-symmetric $\tilde{\Lambda}$, we can define $\Lambda = \frac{1}{2}(\tilde{\Lambda} + \tilde{\Lambda}')$ and note that $\tau'\Lambda\tau = \tau'\tilde{\Lambda}\tau$ for any τ.

1.2 Objective Function

The investor maximizes her expected mean-variance utility of portfolio excess returns with relative risk aversion γ_t:

$$\begin{aligned}\text{utility} &= E\left[E_t[r_{t+1}^{\pi,net}] - \text{Var}_t[r_{t+1}^{\pi,net}]\right] \\ &= E\left[\mu(s_t)'\pi_t - \frac{w_t}{2}\left(\pi_t - g_t\pi_{t-1}\right)'\Lambda_t\left(\pi_t - g_t\pi_{t-1}\right) - \frac{\gamma_t}{2}\pi_t'\Sigma(s_t)\pi_t\right].\end{aligned} \tag{8}$$

We make the following assumptions to keep the problem tractable and stationary. First, the investor has constant risk aversion $\gamma_t = \gamma$, and the risk Σ is constant over time. Second, the investor's wealth (or AUM) grows at an exogenous rate (controlled by how clients take money in and out), so the wealth remains a stationary part of the overall market. Specifically, $w_t\Lambda_t = w\Lambda$, such that the investor faces constant transaction costs relative to her wealth. Under these assumptions, the objective function simplifies as follows

$$\text{utility} = E\left[\mu(s_t)'\pi_t - \frac{w}{2}\left(\pi_t - g_t\pi_{t-1}\right)'\Lambda\left(\pi_t - g_t\pi_{t-1}\right) - \frac{\gamma}{2}\pi_t'\Sigma\pi_t\right]. \tag{9}$$

where the investor chooses π_t while s_t and g_t are exogenous. In summary, the investor's objective is to maximize utility by choosing her portfolio weights $\pi_t = \pi(\vec{s}_t)$ at each time t as a function of all the signals received up until that time, $\vec{s} = (..., s_{t-2}, s_{t-1}, s_t)$.

This setting is ideally suited for a flexible ML implementation for two reasons: First, expected returns are driven by a fully *general function*, μ. Second, the problem is specified in terms of stationary units, namely percentage returns and portfolio weights as fractions of wealth and a stationary objective function.

1.3 The Implementable Efficient Frontier

The utility function depends on risk and expected returns net of trading costs, which gives rise to the implementable efficient frontier as illustrated in Figure 1 in the introduction. Specifically, we defined the implementable efficient frontier as the combination of volatilities and expected net returns, $(\sigma, k(\sigma))_{\sigma\geq 0}$, such that the expected net return is as high as possible for that level of risk:

$$k(\sigma) = \max_{\pi_t\in\Pi} E\left[r_{t+1}^{\pi,net}\right] \quad \text{s.t. } E\left[\pi_t'\Sigma\pi_t\right] = \sigma^2 \tag{10}$$

We are mainly interested in the implementable efficient frontier when taking the maximum among all possible portfolios Π, but, as seen in Figure 1, we also consider the frontier among subsets such as standard portfolio sorts.

As an alternative way to compute the implementable efficient frontier, we can derive the optimal portfolio, π^γ, for any level of risk aversion, γ. Based on all these optimal portfolios, we then compute the corresponding combinations of risk and expected net return:

$$(\sqrt{E\left[(\pi_t^\gamma)'\Sigma\pi_t^\gamma\right]}, E[r_{t+1}^{\pi^\gamma,net}])_{\gamma>0} \tag{11}$$

This generates part of the same implementable efficient frontier, as we show in Appendix B.1. The only difference is that, while (10) can generate a downward-sloping curve as seen in Figure 1, (11) only produces a part of the frontier that ends before the downward sloping part, since an investor never wants the dominated portfolios on the downward-sloping part. The next result characterizes the frontier.

Proposition 1 (Implementable efficient frontier) *(i) The net Sharpe ratio, $k(\sigma)/\sigma$, is decreasing in σ along the implementable efficient frontier for any level of wealth, $w > 0$, when transaction costs are positive, $\Lambda > 0$; (ii) There exists a critical σ_* such that $k(\sigma)$ is increasing and concave for $\sigma < \sigma_*$; (iii) The part of the frontier $\sigma \in (0, \sigma_*)$ is generated by (11) as $\sqrt{E\left[(\pi_t^\gamma)'\Sigma\pi_t^\gamma\right]}$ decreases in γ and converges to σ_* when $\gamma \to 0$; (iv) If $w_1 < w_2$, then the implementable efficient frontier corresponding to a wealth of w_1 is above that of w_2.*

Interestingly, the implementable efficient frontier has a declining net Sharpe ratio – it is not a straight line with a constant Sharpe ratio as in the textbook frontier without trading costs. The declining net Sharpe ratio reflects that investors cannot freely leverage their portfolio to the desired risk in the presence of trading costs – because more leveraged positions are larger and incur more significant trading costs. Further, larger investors face larger trading costs, leading to a lower frontier. Propositions 2–5 characterize the implementable efficient frontier at a deeper level via the properties of the underlying portfolios.

1.4 Empirical Assessments of Portfolios with Trading Costs

The implementable efficient frontier can be computed in-sample or out-of-sample, where the latter provides a more realistic view of investors' experience, as discussed in our empirical

analysis. More broadly, the empirical counterpart of our utility function provides a useful way to evaluate the implementability and economic benefit of any trading strategy:

$$\text{utility(empirical)} = \frac{1}{T}\sum_{t=1}^{T}\left[r_{t+1}^{\pi,gross} - TC_t^{\pi} - \frac{\gamma}{2}\pi_t'\Sigma\pi_t\right]. \tag{12}$$

The first part of the sum, $r_t^{\pi,gross} = r_{t+1}'\pi_t$, is simply the average return of the strategy before trading costs. This is the standard metric by which most papers in the literature evaluate trading strategies.

However, real-world trading involves trading costs, so the second term computes the average trading cost over time, $TC_t^{\pi} = \frac{w}{2}\left(\pi_t - g_t\pi_{t-1}\right)'\Lambda\left(\pi_t - g_t\pi_{t-1}\right)$. This term is a bit more complex since it involves both the portfolio weights in the last period, π_{t-1}, and the portfolio in this period, π_t. Specifically, the trading cost is the cost of moving from the grown old portfolio, $g_t\pi_{t-1}$, to the new portfolio. So the first two terms together yield the average return net of trading costs.

Lastly, we need to consider that investors are risk-averse. In particular, two trading strategies that have delivered the same net returns are different if one did so at a much higher risk. Hence, the last term computes the average disutility of risk. Rather than looking at the ex-ante risk, we can also evaluate the ex-post realized risk, $\frac{1}{T}\sum_{t=1}\frac{\gamma}{2}(r_{t+1}^{\pi,net} - \bar{r}^{\pi,net})^2$, where $\bar{r}^{\pi,net}$ is the average net return. Therefore, our utility function suggests that the main object of interest is the average *return net of trading cost and risk*, which can be seen as the utility flow each period:

$$r_{t+1}^{\pi,util} = r_{t+1}^{\pi,gross} - TC_t^{\pi} - \frac{\gamma}{2}(r_{t+1}^{\pi,net} - \bar{r}^{\pi,net})^2 \tag{13}$$

So, when we evaluate trading strategies empirically, we start with each strategy's return gross of costs, $r_{t+1}^{\pi,gross}$, then compute its return net of trading costs, $r_{t+1}^{\pi,net} = r_{t+1}^{\pi,gross} - TC_t^{\pi}$, and finally focus on the return net of trading costs and risk, $r_{t+1}^{\pi,util}$.

Recall that the net Sharpe ratio declines along the implementable efficient frontier. This result means that an investor cannot just maximize her Sharpe ratio net of trading costs and then choose her risk level – as she could in the standard mean-variance analysis. Instead, she must directly maximize the return net of trading costs and risk, thus jointly considering risk, return, and trading costs. Hence, our framework provides useful tools to evaluate the

implementability of trading strategies in general – namely, the concepts of the implementable efficient frontier and the return net of trading cost and risk, $r_{t+1}^{\pi,util}$.

2 Solution

We seek to find the optimal portfolio π_t that maximizes average returns net of trading costs and risk (9) or its empirical counterpart (12). The problem is too complex and too high-dimensional to attack by brute force ML of a general function $\pi_t = \pi(\vec{s})$ since $\vec{s} = (..., s_{t-2}, s_{t-1}, s_t)$ is simply of too high dimension. So, we need help from economic theory before we turn to ML.

2.1 Optimal Dynamic Portfolio Choice

To solve for the optimal portfolio strategy, we use the "discount factor" m defined in the next lemma. To define this discount factor, we use the notation $\bar{g} = E[g_t]$ for the mean portfolio growth rate as defined in (6), and $G \in \mathbb{R}^{N\times N}$ for the second moments, $G_{ij} = E[g_{it}g_{jt}]$.

Lemma 1 *The fixed point equation*

$$\tilde{m} = \left(w^{-1}\Lambda^{-1/2}\gamma\Sigma\Lambda^{-1/2} + I + \Lambda^{-1/2}((\Lambda^{1/2}(I-\tilde{m})\Lambda^{1/2}) \circ G)\Lambda^{-1/2}\right)^{-1} \tag{14}$$

has a unique, symmetric, positive-definite solution $\tilde{m} \in \mathcal{S}(0,1)$.[5] *For this solution, all eigenvalues of $\Lambda^{-1/2}\tilde{m}\Lambda^{-1/2}\bar{g}\Lambda$ are between zero and one. Furthermore, $m = \Lambda^{-1/2}\tilde{m}\Lambda^{1/2}$ is such that all eigenvalues of $m\Lambda^{-1}\bar{g}\Lambda$ are between zero and one.*

We explain in Appendix A.1 how to calculate m, but for now, let us treat it as a known constant that depends on the exogenous parameters of the model. Based on this known constant, we can compute the optimal portfolio strategy. We start with a simpler case, namely where expected returns are constant. Even in this case, the solution is non-trivial, as is shown by the following proposition.

[5]The discount factor m is defined by an equation involving the symbol "$\circ$," which is an element-wise matrix product. The element-wise product is also called the Hadamard product, and, for any two matrices A and B, it is computed as the matrix $(A \circ B)_{i,j} = A_{i,j}B_{i,j}$. Further, $\mathcal{S}(0,1)$ is the set of positive-definite matrix-valued functions with eigenvalues between zero and one.

Proposition 2 (Optimal dynamic strategy: constant expected returns) *Let $\tilde{m}$ be the unique solution to (14) in $\mathcal{S}(0,1)$, and let $m = \Lambda^{-1/2}\tilde{m}\Lambda^{1/2}$. When expected returns, $\mu(s_t) = \bar{\mu} \in \mathbb{R}^N$, as well as g_t^w, r_t^f are constant, then the optimal portfolio is given by*

$$\pi_t \;=\; \sum_{\theta=0}^{\infty}\left(\prod_{\tau=1}^{\theta} m\, g_{t-\tau+1}\right)\frac{1}{w}(I - m\Lambda^{-1}\bar{g}\Lambda)^{-1}m\Lambda^{-1}\bar{\mu} \tag{15}$$

Furthermore, it is the unique L_2-solution to the stochastic difference equation

$$\pi_t \;=\; m\, g_t\pi_{t-1} + \frac{1}{w}(I - m\Lambda^{-1}\bar{g}\Lambda)^{-1}m\Lambda^{-1}\bar{\mu}\,. \tag{16}$$

To understand the intuition for this proposition, note that the optimal portfolio starts with the old grown position, $g_t\pi_{t-1}$, and then trades toward a fixed portfolio. To understand the direction of the trade, it is useful to write the optimal portfolio as

$$\pi_t \;=\; m\, g_t\pi_{t-1} + (I-m)A = g_t\pi_{t-1} + (I-m)(A - g_t\pi_{t-1})\,, \tag{17}$$

where A is the "aim" portfolio. The aim portfolio has the property that, if the investor holds this portfolio, then the investor optimally does not trade, and otherwise the investor trades in the direction of the aim with trading speed $I - m$. We see from Proposition 2 that the aim portfolio is

$$A \;=\; (I-m)^{-1}(I - m\Lambda^{-1}\bar{g}\Lambda)^{-1}\, c\, \underbrace{\frac{1}{\gamma}\Sigma^{-1}\bar{\mu}}_{Markowitz}\,, \tag{18}$$

where the matrix c is given by

$$c = \frac{\gamma}{w}m\Lambda^{-1}\Sigma \tag{19}$$

So we see that the aim portfolio is related to the Markowitz portfolio but adjusted to account for transaction costs.[6]

We next present a tractable approximation of the optimal portfolio when the expected

[6]We note that, under certain conditions, there is only a small adjustment in the sense that $(I-m)^{-1}(I - m\Lambda^{-1}\bar{g}\Lambda)^{-1}\, c$ is close to I. For example, this happens in the limit when G is close to $\mathbf{1}\mathbf{1}'$. As we show in the Appendix, when $G = \mathbf{1}\mathbf{1}'$, we have $\bar{g} = \text{diag}(\mathbf{1})$, and $c = (I-m)^2$.

returns are not too large. Specifically, we write expected returns as $\mu(s_t) = \epsilon\tilde{\mu}(s_t)$, where $\tilde{\mu}$ is a given function and ϵ is a small number that measures the magnitude of expected returns.

Proposition 3 (Optimal dynamic strategy) *Let $\tilde{m}$ be the unique solution to (14) in $\mathcal{S}(0,1)$ and let $m = \Lambda^{-1/2}\tilde{m}\Lambda^{1/2}$. With expected returns $\mu(s_t) = \epsilon\tilde{\mu}(s_t)$ and $g_t^w = g^w + O(\epsilon)$, $r_t^f = r^f + O(\epsilon)$, the optimal portfolio is*

$$\pi_t \;=\; m\, g_t \pi_{t-1} + (I-m)A_t \;+\; O(\epsilon^2)\,, \tag{20}$$

where the aim portfolio A_t at time t is

$$A_t \;=\; (I-m)^{-1}\sum_{\tau=0}^{\infty}(m\Lambda^{-1}\bar{g}\Lambda)^{\tau}\, cE_t\Big[\underbrace{\frac{1}{\gamma}\Sigma^{-1}\mu(s_{t+\tau})}_{Markowitz_{t+\tau}}\Big]\,. \tag{21}$$

This key theoretical result of the paper shows how to choose the optimal portfolio in two surprisingly simple and intuitive equations. The first equation (20) says that one should always start from the grown position inherited from the last period and then trade toward an aim portfolio.

The second equation (21) shows how the aim portfolio depends on the current and future Markowitz portfolios, thus providing an optimal risk-return tradeoff along the path where these stocks are expected to remain in the portfolio while simultaneously economizing transaction costs.[7] Proposition 3 generalizes the Gârleanu and Pedersen (2013) portfolio optimization principle, "aim in front of the target," when facing trading cost frictions. Unlike Gârleanu and Pedersen (2013), we do not require specific assumptions on return dynamics but, instead, allow a general function $\mu(\cdot)$ that predicts returns.

The proposition also leads to several economically intuitive properties, as shown next.

Proposition 4 (Trading speed) *The eigenvalues of the matrix m are monotone decreasing in G, Σ and γ and increasing in w, in the sense of positive semi-definite order.*[8]

To understand the intuition behind these results, recall from (20) that m is the persistence of the optimal portfolio, or, equivalently, $I-m$ is the trading speed toward the aim. At the

[7] For the convergence of the series (21), it is important that $m\bar{g}$ has all eigenvalues below one in absolute value. As we show in the Appendix, a stronger claim holds, and $\Lambda^{1/2}\bar{g}^{1/2}m\bar{g}^{1/2}\Lambda^{-1/2}$ is a symmetric, positive semi-definite matrix with all eigenvalues between zero and one.

[8] Given two symmetric matrices A, B, we write $A \geq B$ in the sense of positive semi-definite order if $A-B$ is positive semi-definite.

same time, m also determines how much the aim portfolio weights near-time performance versus long-term returns, as seen in (21). So, consider what happens when we move to the right in the implementable efficient frontier in Figure 1 by decreasing risk aversion. This decreasing risk aversion means that m increases, thus reducing trading speed and making the aim more focused on persistent signals. In other words, trading costs increase as the investor takes more risk, but the investor compensates by trading more slowly toward a more stable aim. Likewise, an investor with larger wealth w has a lower trading speed because of more significant market impact costs, providing economic intuition for Figure 1.B.

The next proposition considers limiting portfolios with small are large wealth.

Proposition 5 (Small and large investors) *When wealth approaches zero such that transaction costs become negligible, $w \to 0$, the discount factor converges as $m \to 0$ and the optimal portfolio policy converges to the Markowitz portfolio, $\pi_t \to \frac{1}{\gamma}\Sigma^{-1}\mu_t$.*

When wealth grows large, $w \to \infty$, the optimal portfolio diminishes, $\pi_t \to 0$, but the discount factor m and rescaled portfolio, $w\pi_t$, and aim portfolio, wA_t, converge to finite limits if Λ is diagonal and $\bar{g}_i = \frac{1+r^f+\mu_i}{1+g^w} > 1$ for all i.

Naturally, a tiny investor holds a portfolio close to the Markowitz portfolio because of the low market impact costs. The limiting behavior as wealth goes to infinity is less obvious: As wealth grows infinite, the investor ultimately holds almost all wealth in the risk-free asset as trading a meaningful proportion of wealth in illiquid assets becomes too costly. However, this result does not mean that the portfolio in dollar terms is not large. Instead, what happens is that the portfolio, measured in dollar terms, grows toward a finite limit. In other words, a maximum amount of money can be made in the market, and as wealth increases, the investors ultimately hold this "maximum dollar portfolio."

2.2 Implementing the Solution with Machine Learning

From a machine learning perspective, Proposition 3 is a powerful result if we assume that s_t is Markovian. To see the power of this result, note that the proposition transfers the ML problem from looking for a general function π all *current and past* signals to a problem of looking for a function $A(s_t)$ of only the *current* signals. This enormous reduction in dimension vastly simplifies the problem. There are dual ways of using machine learning to implement Proposition 3, which we call "Multiperiod-ML" and "Portfolio-ML," respectively.

These approaches are designed to find the dynamic optimal portfolio while being aware of transaction costs in a theoretically consistent manner. We describe each approach in turn.

Multiperiod-ML: Machine Learning about Expected Returns across Horizons

The first approach to apply Proposition 3 empirically is to compute the aim portfolio using the expected returns across multiple future periods. To understand this approach, recall first that $\mu_t = E_t(r_{t+1})$ is the short-term expected return, so that $E_t[\mu_{t+\tau}] = E_t[E_{t+\tau}[r_{t+\tau+1}]] = E_t[r_{t+\tau+1}]$ is the current expectation about returns τ periods in the future. Using this identity, the aim portfolio can be written as

$$A(s_t) \;=\; (I-m)^{-1}\sum_{\tau=0}^{\infty}(m\bar{g})^{\tau}c\frac{1}{\gamma}\Sigma^{-1}E_t[r_{t+1+\tau}] \tag{22}$$

The aim portfolio depends on expected returns across all future time horizons.

One approach to apply Proposition 3 empirically is first to use standard ML techniques to predict returns, but do this for a range of forecasting horizons, thus producing proxies for $E_t[r_{t+1+\tau}]$ for all τ. Using these forecasts, the aim portfolio is given by (22). The resulting portfolio can be computed recursively using (20), which we call $\pi^{\text{Multiperiod-ML}}$ as it is based on expected returns over multiple periods.

While there exist many ML methods that can be used to forecast returns, we focus on a single method throughout the paper for its unique combination of flexibility and simplicity (the appendix contains robustness analysis). Specifically, we use the random features (RF) method of Rahimi and Recht (2007).[9] This method is based on the insight that any function can be approximated arbitrarily well by a linear combination of known auxiliary functions. In other words, we can write $E_t[r_{i,t+1+\tau}] = f(s_{i,t})b_\tau$, where f is a known family of functions of the signals and b_τ is a parameter to be estimated (Section 4.2 details our empirical methodology). The RF method is powerful in predicting returns and easily adaptable to our second method, discussed next.

Portfolio-ML: Machine Learning directly about Portfolio Weights

The disadvantage of Multiperiod-ML is that we need a return prediction model for all future return horizons, not just one period ahead. An alternative, and our preferred approach, is to

[9] Kelly et al. (2022) analyze i in the context of return prediction and portfolio choice.

learn *directly* about portfolio weights rather than the two-step procedure of first predicting returns and then constructing portfolios. We thus refer to this approach as "Portfolio-ML."

We use (12) directly as our ML objective, where the optimal portfolio π depends on the aim A, and then search for the function A that maximizes this objective. This method uses the insights that (i) we can focus on the aim portfolio A, which only depends on current signals, and (ii) the objective should penalize transaction costs.

To see how this works, note that Proposition 3 shows that the optimal portfolio is a weighted average of the inherited position and the current aim portfolio via (20). We can express this result as saying that the current optimal portfolio depends on the current and past aim portfolios and their growth over time:

$$\pi_t \;=\; \sum_{\theta=0}^{\infty} \left(\prod_{\tau=1}^{\theta} m\, g_{t-\tau+1} \right) (I-m) A(s_{t-\theta})\,. \tag{23}$$

So, we can replace π by this expression in the objective function (12), which leaves us the task of finding the best aim portfolio, $A(\cdot)$, based on an economic objective function.

In other words, we need to find a general function $A(s_t)$ that maximizes the expected utility. To do this, we again use the ML insight that any function can be approximated arbitrarily well by a linear combination of known auxiliary functions. Specifically, we write $A_t = f(s_t)\beta$, where f is a set of known functions (random features) of the signals and $\beta \in \mathbb{R}^p$ is an unknown vector of parameters. For example, if portfolio weights were linear in the signals, we could take f to be the identity such that $A_t = s_t\beta$. We take f as a set of random features, just like we did to predict returns (detailed in Section 4.2).

So we have boiled the portfolio choice problem down to finding the parameter β, and we next show how to do that in closed form. Plugging $A(s_t) = f(s_t)\beta$ into equation (23), we see that the optimal portfolio depends on known elements and the unknown parameter β:

$$\pi_t \;=\; \left[\sum_{\theta=0}^{\infty} \left(\prod_{\tau=1}^{\theta} m\, g_{t-\tau+1} \right) (I-m) f(s_{t-\theta}) \right] \beta \equiv \Pi_t \beta \tag{24}$$

where Π_t is defined by the last equation. Using this formulation for π_t in the objective (9),

we have

$$\begin{aligned}
\text{utility} &= \frac{1}{T}\sum_t \left[r'_{t+1}\pi_t - \frac{\gamma}{2}\pi'_t\Sigma\pi_t - \frac{w}{2}\left(\pi_t - g_t\pi_{t-1}\right)'\Lambda\left(\pi_t - g_t\pi_{t-1}\right)\right] \\
&= \frac{1}{T}\sum_t \left[r'_{t+1}\Pi_t\beta - \frac{\gamma}{2}\beta'\Pi'_t\Sigma\Pi_t\beta - \frac{w}{2}\left(\Pi_t\beta - g_t\Pi_{t-1}\beta\right)'\Lambda\left(\Pi_t\beta - g_t\Pi_{t-1}\beta\right)\right] \\
&= \frac{1}{T}\sum_t \left[\underbrace{r'_{t+1}\Pi_t}_{\equiv\tilde{r}'_{t+1}}\beta - \frac{1}{2}\beta'\underbrace{\left[\gamma\Pi'_t\Sigma\Pi_t + w(\Pi_t - g_t\Pi_{t-1})'\Lambda(\Pi_t - g_t\Pi_{t-1})\right]}_{\equiv\tilde{\Sigma}_t}\beta\right] \\
&\equiv E_T[\tilde{r}'_{t+1}]\beta - \frac{1}{2}\beta' E_T[\tilde{\Sigma}_t]\beta
\end{aligned} \tag{25}$$

So we can maximize utility by maximizing this quadratic equation in the unknown parameter β. To ensure a robust solution, we add ridge penalty $-\lambda\beta'\beta$, yielding the following solution:

Proposition 6 (Portfolio-ML) *The aim portfolio can be estimated as* $A(s_t) = f(s_t)\beta_T$, *and the corresponding optimal portfolio,* $\pi^{Portfolio\text{-}ML}$, *is given by* (24), *where*

$$\beta_T = (E_T[\tilde{\Sigma}_t] + \lambda I)^{-1} E_T[\tilde{r}_{t+1}] \tag{26}$$

Amazingly, this approach delivers a closed-form solution for the optimal dynamic portfolio in light of transaction costs. To find the optimal portfolio, we compute the two "expectations" on the right-hand side of (26) as their sample counterparts seen in (25). These sample counterparts depend only on data (r_{t+1}, s_t), known parameters, and the ridge parameter λ, which is chosen via ML validation as discussed in Section 4.2. With enough random features and enough time, the estimated portfolio in Proposition 6 asymptotically recovers the optimal portfolio, as discussed in more detail in the appendix (Proposition 9).

2.3 Economic Feature Importance

It is important to determine which characteristics are economically important. To address this issue, we consider the value function $V(s)$, the maximum utility for a given set of signals s. We define the importance of any feature n as

$$\iota_n = V(s) - V(s^{-n}) \tag{27}$$

where s^{-n} is the set of signals s, except that we drop feature n at all times. In other words, the importance of feature n is the drop in utility when the investor no longer has access to this information. The following results provide an intuitive characterization of the drivers of feature importance.

Proposition 7 *In the limit when Λ is small, the investor's steady-state optimal utility is $V(s) = v(s) - c(s) + O(\|\Lambda\|^2)$, where*

$$v(s) = \frac{1}{2} E[Markowitz_t' \, \Sigma \, Markowitz_t] \,, \tag{28}$$

is the value function without transaction costs, and c measures the cost of time-variation in the Markowitz portfolio:

$$c(s) = \frac{1}{2} E[(Markowitz_{t+1} - g_{t+1} Markowitz_t)' \, \Lambda \, (Markowitz_{t+1} - g_{t+1} Markowitz_t)] \,. \tag{29}$$

The importance, ι_n, of feature n is

$$\iota_n = \underbrace{v(s) - v(s^{-n})}_{\textit{efficiency loss}} - \underbrace{(c(s) - c(s^{-n}))}_{\textit{cost reduction}} + O(\|\Lambda\|^2) \,. \tag{30}$$

We see that a feature is more important if it is an important contributor to the Markowitz portfolio (the first term in (30)) and if it is a persistent signal such that it reduces the turnover and, hence, the transaction costs (the second term in (30)).

This result provides intuition on economic feature importance based on an approximation. Empirically, we do not rely on this approximation but, instead, use ML tools to characterize the economic feature importance (27) as described in Section 5.3.

3 Benchmarks based on Standard Approaches

3.1 Standard Approach: Predicting Returns without T-Costs

The standard approach in the literature is to assume away transaction costs, that is, setting $\Lambda = 0$. In this case, the portfolio problem (9) becomes static in the sense that we can choose the optimal portfolio π_t at time t without regard to what happens at other time periods. Hence, the standard approach is focused on finding methods to predict returns

and then using these return predictions to form a portfolio. For example, we can write the standard ML prediction problem as seeking to find a function f of stock characteristics $s_{n,t}$ that minimizes the mean-squared forecast errors for future 1-period (say, 1-month) excess returns $r_{n,t+1}$:

$$\min_{f:\mathbb{R}^K\to\mathbb{R}} \frac{1}{TN}\sum_{n,t}[r_{n,t+1} - f(s_{n,t})]^2 \,. \tag{31}$$

This standard approach generates a function that approximates the conditional mean, $f(s_{i,t}) \cong E[r_{i,t+1}|s_{i,t}]$, when returns are stationary across time and assets, and the number of observations is large. The standard approach to turn such predictions into portfolio weights is to make factor, $\pi^{\text{factor-ML}}$, by going long a value-weighted average of the top 10% of the assets with the highest predicted returns $f(s_{i,t})$ while shorting the bottom 10% of the assets.

This simple factor approach ignores risk and transaction costs, but a more sophisticated method maximizes (9) while assuming zero transaction costs. Using the vector of expected excess returns $\mu(s_t) = (f(s_{1,t}), ..., f(s_{N,t}))'$, the solution to (9) without transaction costs is

$$\pi_t^{\text{Markowitz-ML}} \;=\; \frac{1}{\gamma}\Sigma^{-1}\mu(s_t)\,. \tag{32}$$

which is an ML-based version of the Markowitz portfolio.

3.2 Static Transaction Cost Optimization

A more sophisticated method is first to estimate the vector of expected excess returns $\mu(s_t)$ via (31) and then account for transaction costs in a second step. While this two-step procedure does not fully account for the dynamic nature of the problem, it serves as an interesting benchmark for our fully dynamic method. To see how this works, consider the problem of choosing an optimal portfolio π_t given the existing portfolio $g_t\pi_{t-1}$:

$$\max_{\pi_t\in\mathbb{R}^N}\left\{\pi_t'\mu_t - \frac{\gamma}{2}\pi_t'\Sigma\pi_t - \frac{w}{2\phi}(\pi_t - g_t\pi_{t-1})'\Lambda(\pi_t - g_t\pi_{t-1})\right\}. \tag{33}$$

Here, the transaction costs are divided by a "transaction-cost amortization parameter" ϕ to account for the static nature of the problem in an ad-hoc manner. A naive choice of this parameter is $\phi = 1$, which would mean that the objective (33) compares the returns

earned over the next period (a "flow" variable) with the transaction costs (a "stock" variable) paid today. This comparison is problematic if the portfolio is expected to be held for many periods, so having the fudge factor ϕ is a simple way to address this problem. In particular, if the portfolio is expected to be held for $\phi = 6$ periods, we can amortize the trading cost over these six time periods, thus dividing the current transaction cost by six.

The solution to the static objective (33) is:

$$\begin{aligned}\pi_t^{\text{static-ML}} &= (\gamma\Sigma + \frac{w}{\phi}\Lambda)^{-1}(\mu_t + \frac{w}{\phi}\Lambda g_t \pi_{t-1}) \\ &= m^{\text{static}}\, g_t \pi_{t-1} + (I - m^{\text{static}})\, \pi_t^{\text{Markowitz-ML}}\end{aligned} \tag{34}$$

where $m^{\text{static}} = (\gamma\Sigma + \frac{w}{\phi}\Lambda)^{-1}\frac{w}{\phi}\Lambda$. So we see that this strategy is a weighted average of the inherited grown position, $g_t\pi_{t-1}$, and the current Markowitz portfolio. Said differently, this strategy always trades in the direction of the current Markowitz portfolio — so Markowitz is the "aim portfolio" in this static trading cost formulation. This solution is similar to our optimal portfolio with two exceptions. First, the aim portfolio in the fully dynamic model is more forward looking, distinguishing persistent signals from those with fast alpha decay. Second, the dynamic solution uses the utility optimal discount factor, m, rather than m^{static}.

4 Data and Empirical Methodology

4.1 Data and Inputs to the Portfolio Choice

Returns and Investment Universe

We use the dataset from Jensen et al. (2022), a publicly available dataset and replication code of stock returns and characteristics, with the underlying return data sourced from CRSP and accounting data from Compustat.[10] We restrict our sample to US common stocks (`shrcd`: 10, 11, and 12) traded on AMEX, NASDAQ, or NYSE (`exchcd`: 1, 2, or 3) with a market cap above the 50$^{\text{th}}$ percentile of NYSE stocks (denoted as large-cap stocks). For example, the group of large-cap stocks consists of the largest 1204 stocks at the end of 2020. This sample is deliberately conservative; the effects of trading cost optimization will be magnified among small, micro, and nano-cap stocks subject to notably larger trading costs. Our sample runs

[10]The data, code, and documentation are available at https://github.com/bkelly-lab/ReplicationCrisis/tree/master/GlobalFactors.

from 1952 to 2020, where the first part of the sample is used only for estimation, and our out-of-sample backtests run from 1981 to 2020.

Signals

To predict returns, covariances, and portfolio weights, we use 115 stocks characteristics (or features) studied in Jensen et al. (2022).[11] We standardize each feature in each month by mapping the cross-sectional rank into the [0,1] interval. We set missing values to 0.5 but require at least 57 non-missing features and non-missing market equity at the beginning of the month.

Investor Wealth and Optimization Methods

We assume that the investor wealth grows according to the realized market return, $w_t = w_{t-1}(1 + R_{m,t})$, such that the size of the investor is a stable share of the market. This assumption means that the investor withdraws money when the portfolio has outperformed the market and vice versa when the portfolio has underperformed. For interpretability, we label each investor's size by the corresponding wealth level by the end of 2020. In our baseline specification, the investor's wealth evolves with the market return, so the final wealth by 2020 is \$10 billion.

We assume that the investor optimizes the portfolio each month using either Portfolio-ML, Multiperiod-ML, Static-ML, portfolio sort, or Markowitz-ML, as described below in Sections 4.2–4.3. These portfolio choice methods depend on trading costs and risks, which we estimate each month as described next. While we re-estimate trading costs and risks each month, the investor behaves as if trading costs and risks are constant over time. This assumption simplifies the ML problem, and while it may hurt out-of-sample performance that trading costs and risk do change over time, we find that the methods perform well nevertheless.

Trading Cost Matrix

Trading cost measured in dollars are given $\text{TC}_t = \frac{1}{2}\tau_t'\Lambda\tau_t$ for any vector of dollar trades, τ_t. In our empirical analysis, we let the trading cost matrix be diagonal and calibrate it

[11] Jensen et al. (2022) studies 153 features. However, here we exclude features with poor coverage early in the sample. Table C.1 shows an overview of the features.

based on the estimates in Frazzini et al. (2018). Specifically, we assume that the (expected and realized) market impact, $\frac{1}{2}\Lambda\tau_t$, is 0.1% when trading 1% of the daily dollar volume in a stock. This assumption means that the i^{th} diagonal entry in Λ_t, denoted $\Lambda_{i,t}$, satisfies $0.001 = \frac{1}{2}\Lambda_{i,t}0.01V_{i,t}$, which means that

$$\Lambda_{i,t} = \frac{0.2}{V_{i,t}}, \tag{35}$$

where $V_{i,t}$ is the expected daily dollar volume of stock i at time t. For example, trading \$5 million over a day in a stock with a daily volume of \$500 million moves the price by $\frac{1}{2}\frac{0.2}{\$500\text{m}} \times \$5\text{m} = 0.1\%$, leading to a transaction cost of $\frac{1}{2}\frac{0.2}{\$500\text{m}} \times (\$5\text{m})^2 = \$5000$. We follow Frazzini et al. (2018) and assume that the expected daily volume is equal to the average daily dollar volume over the last six months.

Variance-Covariance Matrix

We need to estimate the variance-covariance matrix, $\Sigma_t = \text{Var}_t(r_{t+1})$, at each time in a way that guarantees it to be positive definite and is broadly consistent with our estimates of expected returns. We use a factor model similar to the MSCI Barra risk model to accomplish these goals. Specifically, security characteristics are used as observable factor loadings, and latent factor returns are estimated via a simple regression (MSCI Barra, 2007).[12] Specifically, each trading day, we estimate a cross-sectional regression of stock returns on stock characteristics

$$r_{i,t+1} = S'_{i,t}\hat{f}_{t+1} + \epsilon_{i,t+1}, \tag{36}$$

and the regression coefficients, $\hat{f}_{t+1}$, are the estimated factor returns. Here, the observed characteristics, $S_{i,t}$, consist of a constant (the number one) and the 13 cluster characteristics from Jensen et al. (2022). These cluster factors capture the main features of return predicting factors but do so in a simplified way to have a tractable variance-covariance matrix (simplified by reducing more than 100 factors to 13 clusters and by using a linear factor model rather than ML). Specifically, each stock's cluster characteristic, $S_{i,t}$, is its average rank of the characteristics in the cluster, standardized by subtracting the mean and dividing by the

[12]The procedure of fixing factor loadings and estimating factor returns differs from models such as Fama and French (1993), that fix factor returns and estimate loadings.

standard deviation each month. This standardization implies that the associated factors are long-short and dollar neutral, and, at the same time, the constant corresponds to an equal-weighted market factor. This structure means that the variance-covariance matrix is:

$$\hat{\Sigma}_t = S_t \text{Var}_t(\hat{f}_{t+1}) S_t' + \text{diag}\left(\text{Var}_t(\hat{\varepsilon}_{i,t+1})\right). \tag{37}$$

Here, $\text{Var}_t(\hat{f}_{t+1})$ is estimated as the exponentially-weighted sample covariance matrix of factor returns over the past ten years of daily observations. We weight observations with exponential decays to put more weight on recent observations and, since correlations move slower than variances, we use a half-life of 378 days for correlations and 126 days for variances.[13]

Lastly, each stock's idiosyncratic variance, $\text{Var}_t(\hat{\varepsilon}_{i,t+1})$, is estimated using an exponentially-weighted moving average of squared residuals, $\epsilon_{i,s}$, from (36) with a half-life of 126 days. We require at least 200 non-missing observations within the last 252 trading days. We use the median idiosyncratic variance within size groups to impute missing observations for stocks with less than 200 valid observations.

4.2 Machine Learning Methodology

Machine Learning via Random Fourier Features

We use the machine learning method called random feature (RF) regression from Rahimi and Recht (2007).[14] To understand the intuition behind this method, note that any function $f(s_t)$ can be approximated as

$$f(s_{i,t}) \approx RF(s_{i,t})\beta, \tag{38}$$

where $\beta \in \mathbb{R}^p$ is a vector of parameters and RF consists of random features. The RF method transforms the original features using random weights and a non-linear activation function. There are several ways to generate random features. We use so-called random Fourier features, which essentially approximate a function via its Fourier transformation.[15]

[13] Specifically, observations j days from t gets a weight of $w_{t-j} = c0.5^{j/\text{half-life}}$ where c is a constant ensuring that the sum of the weights is one.

[14] See Kelly et al. (2022) for a detailed analysis of the theoretical properties of this RF methodology in the context of return prediction.

[15] The approach we use to generate random features is motivated by Sutherland and Schneider (2015), who find that it is preferable to alternative schemes with Gaussian weights.

While this may sound complicated, it is straightforward to do in practice. We first simply draw some random Normal vectors, $w^j \in \mathbb{R}^{115} \sim iidN(0, \eta^2 I)$ for $j = 1, ..., p/2$. Then, for each j, we create a pair of new features, $\sin(s'_{i,t} w^j)$ and $\cos(s'_{i,t} w^j)$, where the sine and cosine functions can capture non-linearities. We finally collect all these p random features:

$$RF(s_{i,t}) = [\sin(s'_{i,t} w^1), \cos(s'_{i,t} w^1), \ldots, \sin(s'_{i,t} w^{p/2}), \cos(s'_{i,t} w^{p/2})]'.$$

The RF method thus involves a vector of parameters β, estimated via a ridge regression, and two hyper-parameters, namely the number of random features p and the standard deviation of the random weights, η. We describe below how we choose these hyper-parameters via tuning.

Machine Learning about Expected Returns: Multiperiod-ML

We estimate expected returns using a ridge regression on the RF-transformed features. The resulting model can be viewed as a two-layer neural network with non-optimized weights in the first layer (the random features) and optimized weights in the final layer (the betas). We predict returns over three different horizons. The first model predicts excess returns over month $t+1$, the second model predicts the average excess return over month $t+2$ to $t+6$, and the third model predicts the average excess return over month $t+7$ to $t+12$.

Machine Learning Directly about the Optimal Portfolio: Portfolio-ML

Our Portfolio-ML learns about the aim portfolio via the relation

$$A_t = f(s_t)\beta = \text{diag}\left(\frac{1}{\sigma_{i,t}}\right) RF(s_t)\beta, \tag{39}$$

where $\beta \in \mathbb{R}^K$ is a parameter, we scale each asset's position by its volatility, $\sigma_{i,t} = \sqrt{\Sigma_{t,ii}}$, and RF consists of random Fourier features as described above. Note the objective for estimation is no longer return prediction but utility maximization, and the solution is given in Proposition 6.

Table 1: Hyper-Parameters

Hyper-parameter	Method		
First tuning layer, h	Portfolio-ML	Multiperiod-ML	Static-ML
Ridge penalty, λ	$\{0, e^4, e^5, \dots e^8\}$	$\{0, e^{-10}, e^{-9.8}, \dots e^{10}\}$	$\{0, e^{-10}, e^{-9.8}, \dots e^{10}\}$
#random features, p	$\{2^6, 2^7, 2^8, 2^9\}$	$\{2^1, 2^2, \dots, 2^{10}\}$	$\{2^1, 2^2, \dots, 2^{10}\}$
Std of weights, η	$\{e^{-3}, e^{-2}\}$	$\{e^{-4}, e^{-3}, e^{-2}, e^{-1}\}$	$\{e^{-4}, e^{-3}, e^{-2}, e^{-1}\}$
Second tuning layer, h^*		Multiperiod-ML*	Static-ML*
Adjustment to mean, u		$\{0.25, 0.50, 1.00\}$	$\{0.25, 0.50, 1.00\}$
Adjustment to variance, v		$\{1, 2, 3\}$	$\{1, 2, 3\}$
Adjustment to t-cost, k		$\{1, 2, 3\}$	$\{\frac{1}{1}, \frac{1}{3}, \frac{1}{5}\}$

Note: The table shows the hyper-parameter space we use for portfolio tuning. For Portfolio-ML, λ is a ridge penalty, p is the number of random features, and η is the standard deviation of random weights. For Multiperiod-ML* and Static-ML* we add a second tuning layer; u shrinks the expected return vectors as $E_t[r_{t+\tau}]^* = uE_t[r_{t+\tau}]$, v increases stock variances as $\Sigma_t^* = \Sigma_t + v\,\text{diag}(\sigma_t)$, and k controls trading cost as $\Lambda_t^* = k\Lambda_t$.

4.3 Portfolio Tuning

The empirical implementation relies on several hyper-parameters as summarized in Table 1. Consider first how we tune our Portfolio-ML method. This method runs a ridge regression on RF-transformed features, so we need to find the ridge parameter λ, the number of random features p, and the standard deviation of random weights η, collected in $h = (\lambda, p, \eta)$.

We tune h as follows. For each h, we compute a "validation backtest" in each year starting in 1971. Specifically, in each year $y \geq 1971$, we compute the optimal beta (26) for that h using monthly data from 1952 to $y - 1$. Using this beta, we compute the optimal portfolio for each month in year y and repeat this process each year until the end of our sample. This process creates – for each h – a backtest from 1971 onwards. These validation backtests are out-of-sample with respect to beta, but we still need to pick h.

Our "actual backtest" starts in 1981. Each year from 1981 onwards, we pick the hyper-parameter h with the highest realized utility in the validation backtest up until now (i.e., from 1971 until the previous year). Using this h and the corresponding beta, we compute the optimal portfolio over the next year, which is, therefore, truly out-of-sample with respect to both h and beta.

For the other methods (Multiperiod-ML, Static-ML, Markowitz-ML, portfolio sort), we first fit a model that predicts returns and then compute the optimal portfolio. We predict

returns using a similar ML method based on random features using the tuning parameters h shown in Table 1.

We show in the next section that Portfolio-ML outperforms Multiperiod-ML and Static-ML. In fact, the latter methods, in fact, deliver negative utility to the investor out of sample. This disappointing performance happens even though the ML model to predict returns works reasonably *well* in terms of how it ranks stocks. The problem is that the resulting portfolios tend to be poorly scaled because out-of-sample returns and risks for optimized portfolios do not match the scale of their ex-ante expected versions. So, this finding already shows the power of the Portfolio-ML method, namely its focus on the economic objective and on directly choosing portfolio weights, which immediately leads to an appropriate scaling of the portfolio with strong performance.

Nevertheless, we want to give the other methods a chance to compete with the Portfolio-ML method. To improve these alternative methods, we add an additional layer of portfolio tuning to Multiperiod-ML and Static-ML, where we add three additional tuning parameters: u, v, k. In particular, u shrinks the expected return vector towards zero, v increases the diagonal of the covariance matrix, and k increases the trading cost matrix:

$$\begin{aligned} E_t^*[r_{t+\tau}] &= uE_t[r_{t+\tau}], \\ \Sigma_t^* &= \Sigma_t + v\,\mathrm{diag}(\sigma_t), \\ \Lambda_t^* &= k\Lambda_t. \end{aligned} \tag{40}$$

To estimate two layers of hyper-parameters, we proceed in the following way (which is rather involved, but, again, Portfolio-ML avoids this complexity). In the first layer, we produce a time series of out-of-sample expected returns based on $h = (\lambda, p, \eta)$ and, in the second layer, we produce optimal portfolios based on $h^* = (u, v, k)$. For the first layer, we update the RF models based on h each decade using the past 30 years of data. We estimate the random features models with each set of hyper-parameters over the first 20 years and pick the ones that lead to the lowest mean squared error over the last ten years. After finding the optimal hyper-parameters h, we re-train the model using all 30 years of data.

In the second layer of portfolio tuning, we update the hyperparameters h^* each year starting in 1981 by choosing the hyper-parameters that led to the highest utility since 1971. This two-layer approach is based on some experimentation to make these methods work,

which gives these methods an advantage. Again, our main finding is that Portfolio-ML nevertheless performs even better. Figure C.1 in the appendix shows the optimal parameters over time.

5 Empirical Results

This section reports our empirical results for each of our methods of portfolio choice. In our baseline specification, we consider an investor with a wealth of $10 billion by the end of 2020 and a relative risk aversion of 10. We also consider other levels of wealth for comparison.

5.1 Out-of-Sample Portfolio Performance

Table 2 shows the out-of-sample performance for each method from 1981 to 2020. Judged by the performance before trading cost, the Markowitz-ML method is the clear winner with an impressive gross Sharpe ratio of 2.00. This finding shows that our methods for predicting risk and return perform well out-of-sample. The gross performance of the trading cost-aware portfolio choice methods (Portfolio-ML, Multiperiod-ML, Static-ML, Multiperiod-ML*, Static-ML*) is substantially lower than that of Markowitz-ML because these methods exploit fewer and less extreme trading opportunities to save on trading costs. Interestingly, several of these methods nevertheless realize a higher gross Sharpe ratio than the standard portfolio sort, presumably because they utilize information about risk and return.

After accounting for trading costs, the net return (and net Sharpe ratio) of the Markowitz-ML and portfolio sort methods are highly negative. Both methods trade too aggressively and are infeasible for the investor we consider. In contrast, the trading cost-aware methods still deliver positive net Sharpe ratios, reaching 1.38 for Portfolio-ML, an impressive performance given that we report out-of-sample results and account for the trading cost of a large investor with $10 billion invested.

Turning to our main objective, which is to maximize the realized utility (12), Table 2 shows that the Portfolio-ML approach delivers the highest realized utility. In contrast, the methods in the top panel deliver negative realized utility. This top panel shows our results when all methods are fitted with a single layer of tuning (estimating portfolio weights or expected returns via random feature ML). While Porfolio-ML performs well with a single layer of tuning, the other methods deliver negative utility.

Table 2: Out-of-Sample Performance Statistics

Method	R	Vol.	SR_{gross}	TC	R-TC	SR_{net}	Utility	Turnover	Lev.
One tuning layer									
Portfolio-ML	0.20	0.14	1.43	0.008	0.19	1.38	0.095	0.32	3.60
Multiperiod-ML	0.32	0.34	0.95	0.182	0.14	0.41	-0.437	1.47	12.70
Static-ML	0.28	0.27	1.06	0.033	0.25	0.94	-0.106	0.76	11.21
Portfolio Sort	0.17	0.15	1.10	1.972	-1.81	-11.87	-1.921	2.60	2.00
Markowitz-ML	3.12	1.56	2.00	+	-	-	-	56.33	53.15
Two tuning layers									
Multiperiod-ML*	0.11	0.08	1.33	0.014	0.09	1.16	0.060	0.40	2.50
Static-ML*	0.13	0.10	1.36	0.024	0.11	1.11	0.060	0.61	3.22

Note: The table shows the out-of-sample performance of the various portfolio choice methods, rebalanced monthly from 1981–2020. Here, R is excess return; Vol. is volatility, SR_{gross} is the Sharpe ratio before trading cost; TC is trading cost, R-TC is excess return minus trading cost; SR_{net} is the Sharpe ratio after trading cost; Utility is the realized utility computed as the excess return after trading cost minus one-half times the assumed risk aversion of 10 times the realized portfolio variance; Turnover is the monthly average of the sum of absolute changes in portfolio weights, and Lev. is the portfolio leverage computed as the monthly average of the sum of absolute portfolio weights. All items except turnover and leverage are annualized. Portfolio-ML and Multiperiod-ML are the two dynamic trading cost optimization methods motivated by Proposition 3, Static-ML is the static trading cost optimization method from (34), Portfolio sort goes long/short the 10% of stocks with the highest/lowest 1-month expected return, and Markowitz-ML is the optimal portfolio absent trading cost from (32). For methods with one tuning layers, we search for the optimal hyper-parameters for a ridge regression implemented on RF-transformed features. For Multiperiod-ML* and Static-ML*, we add a second tuning layer to modify expected return, covariance, and trading cost inputs. An entry of "+" or "−" reflects an extremely high or low value.

It is instructive to consider why Static-ML with a single layer of tuning delivers a negative utility despite its positive net Sharpe ratios. Figure 1.A shows that the indifference curve corresponding to Static-ML goes below the origin, thus yielding a negative utility. This happens because this method realizes too high a risk relative to its ex-ante risk estimate. This is seen in Figure 1.A from the fact that the indifference curve crosses the frontier rather than being tangent (we note that the Static-ML frontier is not drawn but has a similar shape as that of Static-ML*). In other words, two factors determine the realized utility (i.e., the return net of trading costs and risk), out of sample: (i) how good the implementable efficient frontier is, and (ii) whether the method places the investor correctly on the frontier based on the investor's risk aversion. While Static-ML produces a frontier that could deliver positive utility, it places the investor too far to the right on the frontier, thus realizing a negative utility.

To test Portfolio-ML even further, we also compare its performance to versions of the other methods where we give these other methods an extra "advantage" via a second layer of tuning as described in Section 4.3. This second layer of tuning is designed to improve the

scaling of the portfolio, thus helping these methods to place the investor more correctly on the implementable efficient frontier.

Table 2 shows that the two-layer versions outperform the one-layer versions across all performance statistics. This result highlight that the second tuning layer is crucial for these methods based on ML about expected returns. Nevertheless, our Portfolio-ML continues to outperform these methods. This outperformance of Portfolio-ML relative to the two-layer methods shows a benefit of learning directly about portfolio weights, namely that the ML algorithm immediately searches for a well-scaled portfolio that delivers high utility – so no additional tuning layer is needed.

As seen from the notation in Table 2, we add a superscript "$*$" to the implementations with two tuning layers. In the remainder of this section, we focus on the comparison between Portfolio-ML and the two-layers alternatives, Multiperiod-ML* and Static-ML*, studying their performance over time and the statistical significance of their performance differences.

Figure 2 shows that the outperformance of Portfolio-ML in terms of net returns and realized utility is consistent over time. The outperformance of Portfolio-ML is all the more remarkable when considering that the other methods were given the advantage of a second level of tuning to make them perform better. Figure 2 also shows some interesting time-series patterns in performance. For example, we see that several of these methods had a relatively lower performance during the dot-com bubble in 2000, the global financial crisis in 2008, and the COVID-19 crash in 2020.

One of the reasons behind the outperformance of Portfolio-ML is that this method keeps trading costs lower. This lower trading cost is achieved via a lower monthly turnover of 32% relative to 40% and 61% for Multiperiod-ML* and Static-ML*, respectively, as seen from Table 2. Figure 3 shows how the ex-ante volatility, leverage, and turnover evolve over time, again showing that Portfolio-ML tends to have a lower turnover.

To visualize an example of some specific portfolio weights over time, Figure 4 depicts how the portfolio weights for Apply and Xerox stocks evolve for each method. Portfolio-ML adjusts its positions more slowly than the other methods, especially for the less liquid stock (Xerox).

Table 3 reports the statistical significance of the relative performance differences across portfolio choice methods. Specifically, the table reports the Bayesian probability that each method outperforms any of the other methods. To compute these pairwise probabilities of

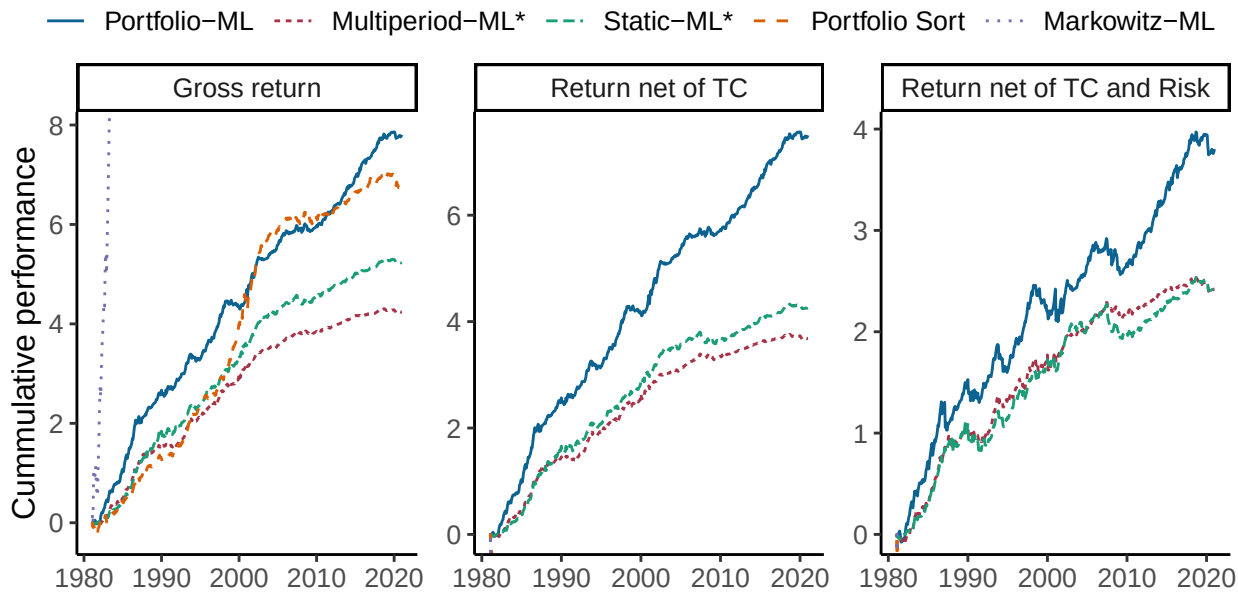

Figure 2: Performance over Time

Note: The left panel shows the cumulative sum of returns before trading cost, $r_{t+1}^{\pi,gross}$, for each portfolio method. The middle panel shows the cumulative sum of returns net of trading cost, $r_{t+1}^{\pi,net}$. The right panel shows the cumulative return net of trading cost (TC) and net of disutility from risk, computed as $r_{t+1}^{\pi,util} = r_{t+1}^{\pi,gross} - TC_t^{\pi} - \frac{\gamma}{2}(r_{t+1}^{\pi,net} - \bar{r}_{t+1}^{\pi,net})^2$, corresponding to the realized utility. We assume that the investors has a relative risk aversion of 10 and invested wealth of \$10 billion by the end of 2020.

one method outperforming another, we first compute the utility flow (return net of trading costs and risk) of each method π at time $t+1$ as $r_{t+1}^{\pi,util} = r_{t+1}^{\pi,gross} - TC_t^{\pi} - \frac{\gamma}{2}(r_{t+1}^{\pi,net} - \bar{r}_{t+1}^{\pi,net})^2$, where the relative risk aversion is $\gamma = 10$ as before. We then compute the utility difference between any two methods, say π and $\tilde{\pi}$, as $d_{\pi,\tilde{\pi},t+1} = r_{t+1}^{\pi,util} - r_{t+1}^{\tilde{\pi},util}$. The posterior of the true utility difference is then normally distributed with mean $\bar{d}_{\pi,\tilde{\pi}} = \frac{1}{T}\sum_{t=1}^{T} d_{\pi,\tilde{\pi},t}$ and variance $\frac{1}{T-1}\sum_{t=1}^{T}(d_{\pi,\tilde{\pi},t} - \bar{d}_{\pi,\tilde{\pi}})^2$ assuming that the difference is normally distributed with a non-informative prior about the mean and a known variance. Based on these calculations, Table 3 reports the posterior probability that $d_{\pi,\tilde{\pi}} > 0$, that is, the posterior probability that the first portfolio choice method, π, delivers a higher average utility than the second method, $\tilde{\pi}$.

Table 3 shows that the probability that Portfolio-ML delivers a higher expected utility than Multiperiod-ML*, Static-ML*, Portfolio sort, and Markowitz-ML are, respectively, 95%, 96%, 100%, and 100%, suggesting that the superiority of Portfolio-ML is not just random noise.

Alternatively, we can think of the probabilities in Table 3 as being approximately the p-value of a one-sided test that the realized utility of i is greater than j. Hence, we see that

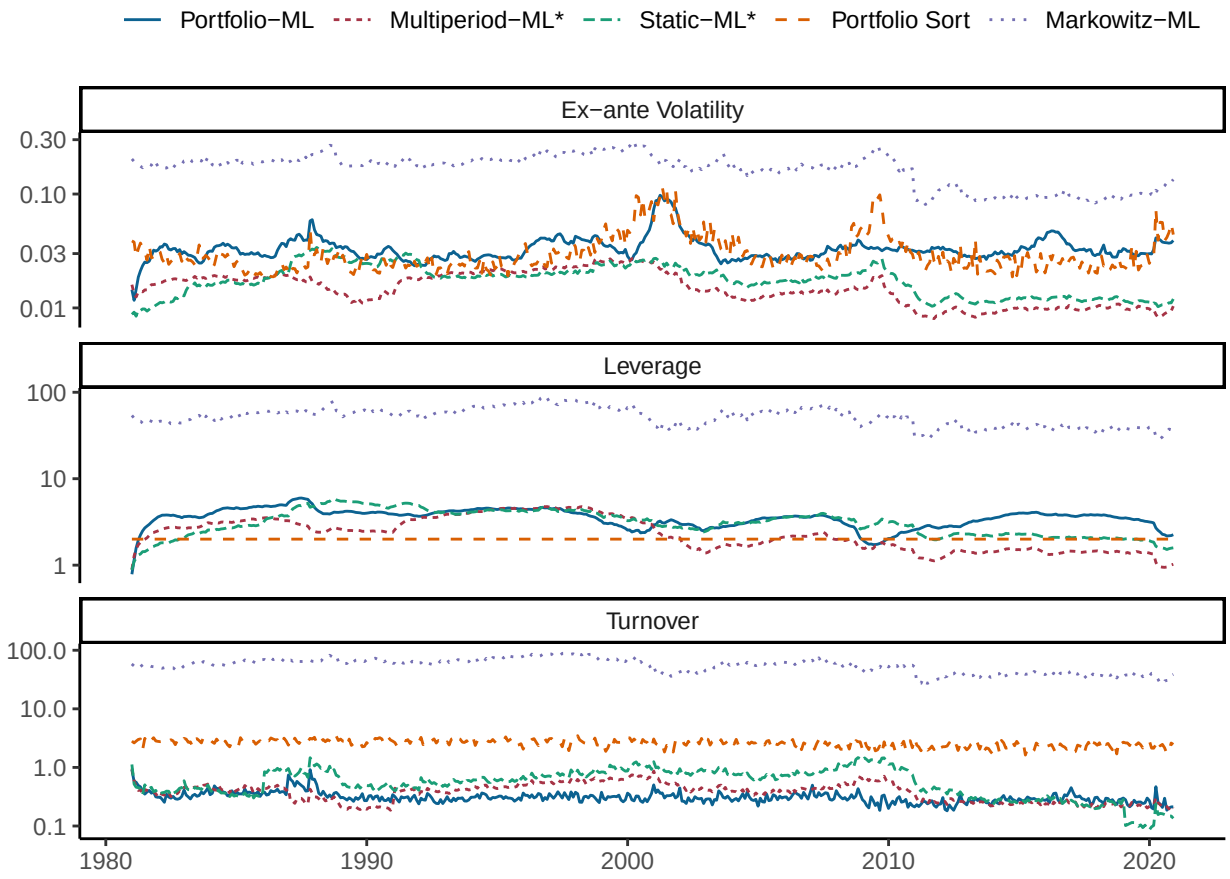

Figure 3: Portfolio Statistics over Time

Note: The top panel shows the ex-ante volatility of each portfolio method based on a monthly updated covariance matrix. The middle panel shows portfolio leverage defined as the sum of absolute portfolio weights. The lower panel shows the monthly portfolio turnover defined as $\sum_i |(\pi_t - g_t \pi_{t-1})_i|$, the sum of absolute differences between the current portfolio weight, π_t, and the grown portfolio weight from last month, $g_t \pi_{t-1}$. We use a logarithmic scale for the y-axis because of large differences across methods.

we can reject that Portfolio-ML delivers a lower realized utility than the other methods at conventional levels of significance.

Lastly, Table 4 reports the return correlations of the various portfolio choice methods. We see that all methods are positively correlated, but the correlations tend to be modest in size. In addition to showing the relative connection across these methods of portfolio choice, these findings may also be informative about asset pricing more generally. Indeed, the Markowitz-ML portfolio return can be viewed as an estimate of the minimum-variance stochastic discount factor (SDF) in a frictionless market as shown by Hansen and Jagannathan (1991). Since risk adjustments depend on covariance with the SDF, a natural ques-

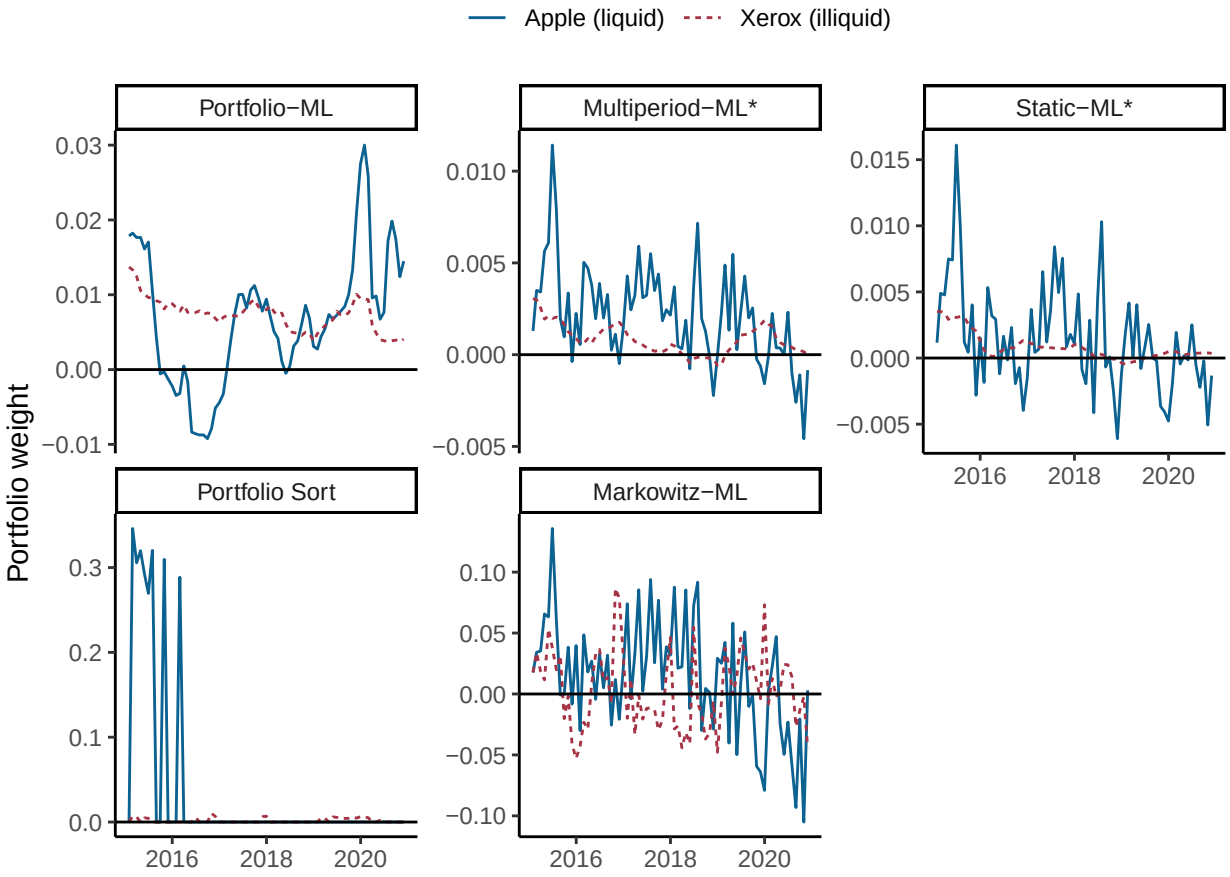

Figure 4: Portfolio Weights: Apple vs. Xerox

Note: The figure shows the portfolio weights of Apple and Xerox for each of the five portfolio choice methods, 2015–2020. Apple is chosen as an example of a relatively liquid stock and Xerox as a relatively illiquid stock over this time period. By the end of 2020, the average daily dollar volume over the past six months was \$16.39B for Apple and \$0.06B for Xerox.

tion is how closely SDF aligns with the corresponding measure designed for a market with frictions. The correlation between Portfolio-ML and Markowitz-ML is only 0.17, indicating that the marginal utility of an investor with \$10 billion using Portfolio-ML could be very different from risk adjustments in a frictionless market.

In summary, this section shows that Portfolio-ML outperforms the other methods, delivering a high net Sharpe ratio and high utility. While this strong performance suggests that Portfolio-ML works well, a few words of warning are in order. First, while the net performance is extremely good in our simulation, real-world investors seeking to achieve this performance must often pay fees to an asset manager (e.g., a hedge fund) running such strategies and face other real-world complications, potentially reducing performance. Sec-

Table 3: Relative Probability of Outperformance

	Portfolio-ML	Multiperiod-ML*	Static-ML*	Portfolio Sort	Markowitz-ML
Portfolio-ML		95%	96%	100%	100%
Multiperiod-ML*	5%		51%	100%	100%
Static-ML*	4%	49%		100%	100%
Portfolio Sort	0%	0%	0%		100%
Markowitz-ML	0%	0%	0%	0%	

Note: The table shows the probability of the row method having a higher average utility than the column method. The probability is computed via an uninformative prior, assuming that the difference in utility is normally distributed. The utility flow of any method π at time $t+1$ is $r^{\pi,util}_{t+1} = r^{\pi,gross}_{t+1} - TC^{\pi}_{t} - \frac{\gamma}{2}(r^{\pi,net}_{t+1} - \bar{r}^{\pi,net}_{t+1})^2$. One can also think of each number as the p-value in the test of whether the average utility of the portfolio choice method in the row is greater than the average utility of the method in the column.

ond, investors might not have been able to realize this performance in real-time due to more limited computing power and a less developed ML methodology in the early sample. In any case, this warning applies to any simulation, and the statistically significant outperformance of Portfolio-ML relative to other methods is an encouraging apples-to-apples test.

5.2 Evidence on the Implementable Efficient Frontier

Textbooks and real-world investors often depict their investment opportunities in terms of the achievable combinations of risk and expected return. This illustration highlights that investors seek a portfolio on the efficient frontier with the highest expected return for any level of risk. The textbook version of the efficient frontier – without trading costs – is a straight-line tangent to the hyperbola of risky investments. However, we propose that investors should focus on what we call the implementable efficient frontier, namely the efficient frontier net of trading costs.

Figure 1 illustrates our estimated implementable efficient frontier, out-of-sample. To understand how we have generated this plot, we start by describing the two benchmarks

Table 4: Portfolio Correlations

	Portfolio-ML	Multiperiod-ML*	Static-ML*	Portfolio Sort
Multiperiod-ML*	0.51			
Static-ML*	0.55	0.80		
Portfolio Sort	0.24	0.46	0.53	
Markowitz-ML	0.17	0.50	0.59	0.56

Note: The table shows the time-series correlation of the returns before trading costs for the various portfolio choice methods, 1981-2020.

for a world without trading costs. The hyperbola is a mean-variance frontier of risky assets inspired by Markowitz (1952, 1959). We generate the points on the frontier by minimizing variance for a given mean and requiring that portfolio weights sum to 1:

$$\begin{aligned} \min_{\pi_t \in \mathbb{R}^N} \quad & \pi_t' \Sigma_t \pi_t, \\ \text{s.t.} \quad & \pi_t' \mu_t = k, \\ & \pi_t' 1_N = 1, \end{aligned}$$

where k is the required mean return, and 1_N is a vector of ones. The solution is given by

$$\pi_t = \frac{c_t k - b_t}{d_t} \Sigma_t^{-1} \mu_t + \frac{(a_t - b_t k)}{d_t} \Sigma_t^{-1} 1_N, \tag{41}$$

where $a_t = \mu_t' \Sigma_t^{-1} \mu_t$, $b_t = 1_N' \Sigma_t^{-1} \mu_t$, $c_t = 1_N' \Sigma_t^{-1} 1_N$, and $d_t = a_t c_t - b_t^2$ are constants. Implementing this solution for a range of k's generates the hyperbola. A standard presentation of the frontier uses one cross-section of stocks (i.e., one μ_t and one Σ_t) and presents the ex-ante expected frontier. In contrast, we show the realized frontier out-of-sample. Specifically, for each k, we update portfolio weights each month using (41). We then record the realized return and volatility before trading cost over the sample, 1981-2020. In contrast to the standard textbook presentation, the efficient frontier of risky assets in Figure 1 also accounts for out-of-sample performance decay. As such, it gives a more realistic picture of what investors could achieve absent trading costs.

The second benchmark for the case without trading cost is the Markowitz-ML portfolio. In a standard presentation, the line from this portfolio would be tangent to the hyperbola. However, because our analysis is out-of-sample, this outcome is not ensured. In fact, the Markowitz-ML portfolio lies above the hyperbola.[16]

Next, we shift the attention from the frictionless benchmarks to our main focus, namely

[16]This result might be surprising since the Markowitz-ML portfolio in a given period is proportional to the tangency portfolio of risky assets. Specifically, the tangency portfolio is $\pi^{\text{TPF}} = \frac{1}{b_t} \Sigma_t^{-1} \mu_t$, while the Markowitz-ML portfolio is $\pi_t^{\text{Markowitz-ML}} = \frac{1}{\gamma} \Sigma_t^{-1} \mu_t$. However, the scaling constant for Markowitz-ML, γ^{-1}, is fixed across time periods, while the scaling constant for the tangency portfolio, b_t^{-1}, varies significantly over time and even turns negative during a minority of periods. So the Markowitz portfolio can dominate the hyperbola for two reasons. First, when b_t is negative, the Markowitz portfolio shorts the tangency portfolio (which is on the lower branch of the hyperbola during such times), and the efficient frontier lies strictly above the hyperbola. Second, the differing time-varying scales of $\pi_t^{\text{Markowitz-ML}}$ and π^{TPF} (i.e., the different timing of the common underlying portfolio) turn out to work in favor of the Markowitz portfolio.

the implementable efficient frontier. These are computed via equation (11). Figure 1.A shows the efficient frontier after trading cost for an investor using different portfolio methods. We implement each method assuming an investor with a wealth that reaches \$10b by 2020.

We illustrate the performance of Markowitz-ML and portfolio sort with trading cost by scaling each portfolio to ex-post volatilities ranging from 0 to 10% in increments of 1%. We see their returns after trading costs are negative, except at very low volatilities. Hence, implementable efficient frontiers of these standard methods show that an investor maximizes utility by putting almost all wealth into the risk-free asset, thus choosing to hardly trade on these standard methods.

Turning to the trading-cost-aware methods, Portfolio-ML and Static-ML*, we see that their performance is much better. Instead of varying the ex-post volatilities, we implement the methods under five different relative risk aversions, $\gamma \in \{1, 5, 10, 20, 100\}$, and interpolate between their realized performance to plot an efficient frontier. Comparing the two implied frontiers, we see that Portfolio-ML leads to a higher achievable return for the same volatility. More generally, the figure shows that it is feasible for a large investor to generate an attractive implementable efficient frontier, even net of trading costs.

Panel B of Figure 1 shows how the implementable efficient frontier varies by investor size. Specifically, we implement the Portfolio-ML method for the same relative risk aversions as above, but now we also vary the investor wealth, $w_{2020} \in \{0, 10^9, 10^{10}, 10^{11}\}$. Such a plot is not interesting without trading costs since the efficient frontier is the same regardless of investor size. With trading costs, this is no longer the case. Price impact is increasing in trade size, so a larger investor must trade more slowly and focus more on liquid stocks. Naturally, these effects imply that larger investors have a worse risk-return tradeoff. The results in Figure 1.B quantifies how much worse. The figure shows the substantial cost of being a large investor. For example, an investor with \$10B and a relative risk aversion of 10 gets a net excess return of 19% at 14% volatility. If the same investor had \$1B, a 14% volatility would provide a net return of 22%. In summary, once we introduce trading costs, we no longer have a unique, efficient frontier. Instead, the implementable efficient frontier depends on the investor's size and portfolio choice method.

Finally, Figure 5 shows how the efficient frontier depends on access to certain features. We use a methodology known as permutation feature importance to assess this dependence. We provide a detailed description of this methodology in Section 5.3. Briefly, for a specific theme,

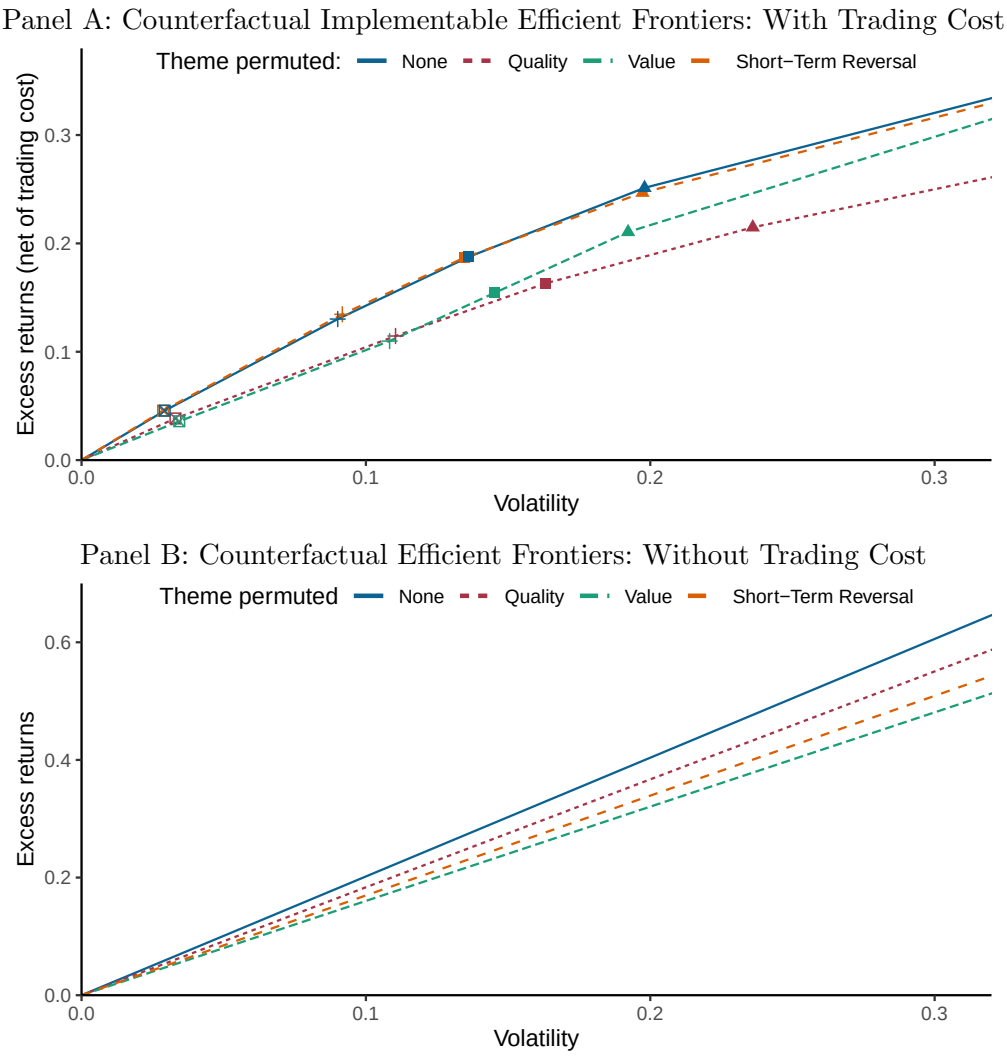

Figure 5: Feature Importance: Counterfactual Implementable Efficient Frontiers

Note: Panel A shows the implementable efficient frontier with trading costs for an investor with a wealth of \$10B by 2020. We implement the Portfolio-ML method using a counterfactual data set, where we permute all feature values related to either quality, value, or short-term reversal. The solid blue line shows the frontier using the actual data. Panel B shows the same analysis without trading cost, now using the Markowitz-ML method. In both panels, the relative risk aversions are 1 (circle), 5 (triangle), 10 (square), 20 (plus), and 100 (boxed cross) and the sample period is 1981-2020.

say quality, we randomly shuffle (permute) all features related to this theme, effectively breaking their informational content. Using this counterfactual data, we then implement Portfolio-ML (Panel A) or Markowitz-ML (Panel B). If the theme is important, breaking its informational content should lead to a worse risk-return tradeoff. In other words, destroying important features leads to a less desirable efficient frontier. We build separate counterfactual data sets by permuting quality, value, and short-term reversal signals. The solid blue line

shows the efficient frontier with the original data.

Panel A shows the impact on the efficient frontier generated by Portfolio-ML, after trading cost, for an investor with $10B in wealth. We see that quality and value signals are crucial for the implementable frontier. In contrast, destroying the informational content in short-term reversal signals barely changes the achievable frontier because these trading-cost-aware portfolio choice methods hardly use this signal anyway.

Panel B shows the impact on the efficient frontier generated by Markowitz-ML before trading cost. Here, all three themes are important. Interestingly, while short-term reversal has a minor effect on the implementable frontier with trading costs in Panel A, it greatly impacts the frontier without trading costs in Panel B.

In summary, looking at the efficient frontier without trading cost suggest that value, quality, and short-term reversal signals are all important for the efficient frontier. However, looking at the implementable efficient frontier of a large investor, value and quality remain important while short-term reversal does not. These results highlight that feature importance can change drastically when accounting for trading costs, and we explore this finding further in the next section.

5.3 Economic Feature Importance

Following the theoretical discussion in Section 2.3, we define economic feature importance as the drop in realized utility when excluding a feature from the information set of the investor. To implement this idea, we use the concept of permutation feature importance, introduced by Breiman (2001), which is a standard model-agnostic method for assessing feature importance of machine learning models (Molnar, 2022). The basic idea behind permutation feature importance is to permute features randomly and assess the decline in a user-defined value function. As such, we do not exclude the feature from the information set but destroy any predictive relationship between the feature and the outcome variable.

To compute economic feature importance for any portfolio choice method i, say Portfolio-ML, we first compute the baseline realized utility, utility$[\pi^i(s_{\text{orig}})]$, of the portfolio, π^i, computed based on the original features, s_{orig}. Next, for each feature j, we randomly permute its associated values at each given time while keeping all other features at their actual values. We then implement the portfolio method using the same parameters as in the original specification, but now with the permuted features, $s_{\text{perm},j}$, as inputs. Fi-

nally, we compute the economic feature importance as the resulting drop in utility, $\text{FI}_j^i = \text{utility}[\pi^i(s_{\text{orig}})] - \text{utility}[\pi^i(s_{\text{perm},j})]$. In other words, a feature j is economically important for method i if destroying its informational content leads to a large drop in realized utility, that is, a large FI_j^i.

A potential issue with permuting each feature separately is that substitution effects can distort the inference. For example, we include many different value features, so the effect of permuting a specific feature, such as book-to-market, is muted because the method can rely on other value features, such as assets-to-market or earnings-to-price. To handle substitution effects, we permute *all* features within a specific theme (or cluster) and record feature importance at the theme level. We use the 13 themes from Jensen et al. (2022), shown in Table C.1 in the appendix.

Figure 6 shows the features' importance for three different methods. The left and middle panels show feature importance after trading costs for a large investor with a wealth of \$10b by 2020 for, respectively, Portfolio-ML and Multiperiod-ML*. The right panel shows feature importance without trading costs for Markowitz-ML, where we ignore trading costs because this method does not work after trading costs, making it meaningless to discuss net-of-cost feature importance. As such, the right panel serves as the benchmark of a frictionless market.

Looking at the frictionless benchmark in the right panel of Figure 6, we see that the important feature themes before trading costs are value, short-term reversal, and low risk. Turning to the net-of-cost feature importance in the left and middle panels, we see that value remains important for a large investor. In contrast, short-term and low-risk are far less important due to the high turnover of many factors within these themes (see figure C.2). This finding is consistent with the theoretical results of Section 2.3, namely that high-frequency features are less important in the presence of trading costs.

For example, the short-term reversal theme includes the short-term reversal factor, which has a monthly autocorrelation of -0.04. This autocorrelation is not just low but actually negative, giving rise to large portfolio turnover.[17] For a large investor, the predictive ability of short-term reversal is not enough to overcome the cost of trading it. Similarly, the low-risk theme includes features such as the past-month volatility or the maximum return over the past month, where the predictive ability is short-lived. (We note that some of the factors in

[17]The autocorrelation of a feature is computed as the average autocorrelation across all stocks with at least five years of monthly observations.

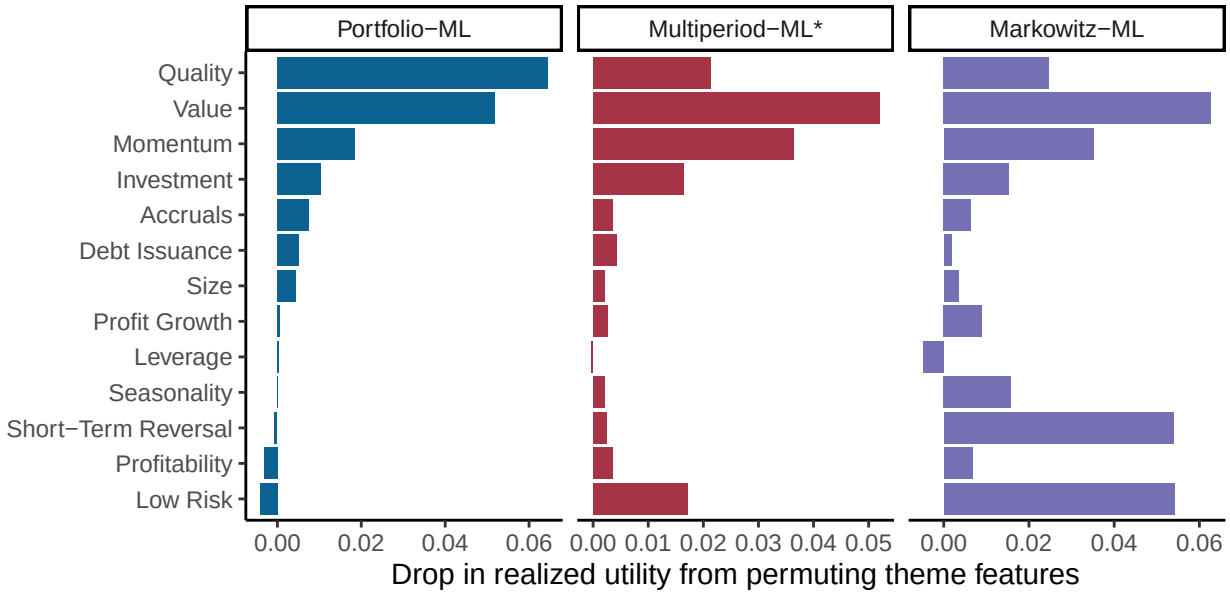

Figure 6: Economic Feature Importance

Note: The figure shows a utility-based feature importance measure for Portfolio-ML and Multiperiod-ML*, the two trading cost-aware methods motivated by Proposition 3, and Markowitz-ML, which is the optimal solution absent trading cost from (32). We randomly shuffle the associated features for each theme while keeping all other features at their actual value. We then implement each method based on this counterfactual data and measure feature importance as the difference in realized utility relative to the implementation that uses the actual data. For Markowitz-ML, we assume that the investor can trade without incurring the trading cost.

these themes have low turnover, e.g., market beta, and may individually have a meaningful feature importance, but our results indicate that the overall themes are less important net of trading costs.) In comparison, book-to-market has a monthly autocorrelation of 0.94, indicating that it is a highly persistent feature, thus economizing on trading costs.

For Portfolio-ML, quality emerges as the most important theme. The median monthly autocorrelation of quality features is 0.93, so the result is again consistent with the theoretical findings. Perhaps surprisingly, momentum, generally considered a "fast" signal, is one of the most important themes for a large investor. However, several momentum features actually do exhibit meaningful persistence; for example, 12-month return momentum has a monthly autocorrelation of 0.87. Furthermore, momentum and value are negatively correlated, which leads to less trading because the two signals offset each other.

In summary, value, quality, and momentum are the economically important feature themes for a large investor facing trading costs. Before trading costs, high-frequency signals such as short-term reversal are the most important.

6 Conclusion

We develop a bridge between ML and portfolio choice with trading costs. To accomplish this bridge, we solve the optimal portfolio problem with transaction costs when returns are predictable via an *arbitrary* function of security characteristics and then show how the solution can be computed in a tractable way via machine learning directly about portfolio weights.

To evaluate the usefulness of our method – and, in fact, any method of portfolio choice – we propose that investors should focus on the implementable efficient frontier, not the standard cost-agnostic efficient frontier. We show empirically that our method expands the implementable efficient frontier relative to other methods of portfolio choice. In other words, we find significant out-of-sample gains from our method even relative to sophisticated and more highly parameterized alternatives. We also consider several comparative statics, showing how the implementable efficient frontier contracts for larger investors facing higher market impact costs.

Finally, the method implies a novel view of which securities are important. Indeed, while standard methods that ignore transaction costs focus on transient features that work well on paper for small stocks, our method naturally selects persistent features of economic importance.

References

Avramov, D., S. Cheng, and L. Metzker (2021). Machine learning versus economic restrictions: Evidence from stock return predictability. *Management Science, forthcoming*.

Balduzzi, P. and A. W. Lynch (1999). Transaction costs and predictability: some utility cost calculations. *Journal of Financial Economics 52*(1), 47–78.

Bali, T. G., H. Beckmeyer, M. Moerke, and F. Weigert (2021). Option return predictability with machine learning and big data. *Available at SSRN 3895984*.

Bali, T. G., A. Goyal, D. Huang, F. Jiang, and Q. Wen (2022). Predicting corporate bond returns: Merton meets machine learning. *Georgetown McDonough School of Business Research Paper* (3686164), 20–110.

Brandt, M. W., P. Santa-Clara, and R. Valkanov (2009). Parametric portfolio policies: Exploiting characteristics in the cross-section of equity returns. *The Review of Financial Studies 22*(9), 3411–3447.

Breiman, L. (2001). Random forests. *Machine learning 45*(1), 5–32.

Cakici, N. and A. Zaremba (2022). Empirical asset pricing via machine learning: The global edition. *Available at SSRN 4028525*.

Chen, A. Y. and M. Velikov (2021). Zeroing in on the expected returns of anomalies. *Working paper, Federal Reserve Board*.

Chen, L., M. Pelger, and J. Zhu (2021). Deep learning in asset pricing. *Available at SSRN 3350138*.

Choi, D., W. Jiang, and C. Zhang (2021). Alpha go everywhere: Machine learning and international stock returns. *Available at SSRN 3489679*.

Collin-Dufresne, P., K. Daniel, and M. Sağlam (2020). Liquidity regimes and optimal dynamic asset allocation. *Journal of Financial Economics 136*(2), 379–406.

Constantinides, G. M. (1986). Capital market equilibrium with transaction costs. *Journal of Political Economy 94*, 842–862.

Davis, M. and A. Norman (1990). Portfolio selection with transaction costs. *Mathematics of Operations Research 15*, 676–713.

Detzel, A. L., R. Novy-Marx, and M. Velikov (2021). Model selection with transaction costs. *Available at SSRN 3805379*.

Fama, E. F. and K. R. French (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics 33*(1), 3–56.

Frazzini, A., R. Israel, and T. J. Moskowitz (2018). Trading costs. *Available at SSRN 3229719*.

Freyberger, J., A. Neuhierl, and M. Weber (2020). Dissecting characteristics nonparametrically. *The Review of Financial Studies 33*(5), 2326–2377.

Gârleanu, N. and L. H. Pedersen (2013). Dynamic trading with predictable returns and transaction costs. *The Journal of Finance 68*(6), 2309–2340.

Gârleanu, N. and L. H. Pedersen (2016). Dynamic portfolio choice with frictions. *Journal of Economic Theory 165*, 487–516.

Gu, S., B. Kelly, and D. Xiu (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies 33*(5), 2223–2273.

Gu, S., B. Kelly, and D. Xiu (2021). Autoencoder asset pricing models. *Journal of Econometrics 222*(1), 429–450.

Han, Y., A. He, D. Rapach, and G. Zhou (2021). Expected stock returns and firm characteristics: E-lasso, assessment, and implications. *Assessment, and Implications (September 10, 2021)*.

Hansen, L. P. and R. Jagannathan (1991). Implications of security market data for models of dynamic economies. *Journal of political economy 99*(2), 225–262.

Jensen, T. I., B. T. Kelly, and L. H. Pedersen (2022). Is there a replication crisis in finance? *Journal of Finance, forthcoming*.

Kelly, B. T., S. Malamud, and K. Zhou (2022). The virtue of complexity in return prediction.

Kelly, B. T., D. Palhares, and S. Pruitt (2022). Modeling corporate bond returns. *Journal of Finance*.

Kelly, B. T., S. Pruitt, and Y. Su (2019). Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics 134*(3), 501–524.

Koijen, R. S. and M. Yogo (2019). A demand system approach to asset pricing. *Journal of Political Economy 127*(4), 1475–1515.

Krein, M. G. and M. Rutman (1950). *Linear operators leaving invariant a cone in a Banach space*. Number 26. American Mathematical Society.

Leippold, M., Q. Wang, and W. Zhou (2022). Machine learning in the chinese stock market. *Journal of Financial Economics 145*(2), 64–82.

Li, S. A., V. DeMiguel, and A. Martin-Utrera (2020). Which factors with price-impact costs? *Victor and Martin-Utrera, Alberto, Which Factors with Price-Impact Costs*.

Lynch, A. W. and P. Balduzzi (2000). Predictability and transaction costs: the impact on rebalancing rules and behavior. *Journal of Finance 55*, 2285–2309.

Markowitz, H. M. (1952). Portfolio selection. *The Journal of Finance 7*(1), 77–91.

Markowitz, H. M. (1959). *Portfolio Selection: Efficient Diversification of Investments*. Yale University Press.

Molnar, C. (2022). *Interpretable Machine Learning* (2 ed.).

MSCI Barra (2007). Barra Risk Model Handbook.

Rahimi, A. and B. Recht (2007). Random features for large-scale kernel machines. *Advances in neural information processing systems 20*.

Sutherland, D. J. and J. Schneider (2015). On the error of random fourier features. *arXiv preprint arXiv:1506.02785*.

Van Binsbergen, J. H. and C. C. Opp (2019). Real anomalies. *The Journal of finance 74*(4), 1659–1706.

Appendix

The appendix is organized as follows. Appendix A presents implementation details, including how to compute the discount factor m (section A.1) and details on Multiperiod-ML (A.2).

Appendix B contains proofs, including a key technical lemma for verifying optimality of policies (B.2), properties of m used in the proofs (B.3), proofs of Propositions 2 and 3 (B.4), proofs of Propositions 4 and 5 (B.5), the optimality of Portfolio ML (B.6), and economic feature importance (B.7).

Appendix C contains further empirical information, including an overview of the security characteristics used empirically (C.1), the estimated hyper-parameters over time (C.2), and the autocorrelation of the features and their importance for different return prediction horizons (C.3).

A Implementation Details

A.1 Computing the Discount Factor m

Lemma 2 *Suppose that Λ and Σ are both diagonal. Let $\Lambda^{-1/2}\Sigma\Lambda^{-1/2} = \text{diag}(q_{i,i})$ is diagonal, then there exists a unique diagonal solution $\tilde{m}$ to (B.58) such that $\Lambda^{-1/2}\tilde{m}\Lambda^{1/2}\bar{g}$ has all eigenvalues below one in absolute value. It is given by*

$$m_{i,i} \;=\; \frac{2}{w^{-1}\gamma q_{i,i} + G_{i,i} + 1 + \sqrt{(w^{-1}\gamma q_{i,i} + G_{i,i} + 1)^2 - 4G_{i,i}}} \tag{A.1}$$

Proof of Lemma 2. The proof follows by direct calculation. □

Another special case with a closed-form solution is when G has a rank of one, which is not a realistic case, but turns out to be a useful approximation. Specifically, we have that $G = \bar{g}\bar{g}' + \frac{1}{(1+r^f+\bar{\mu})^2}\Sigma \cong \bar{g}\bar{g}'$, where[18] $\bar{g} \;=\; (1+r^f+\bar{\mu})^{-1}(1+r^f+E[\mu])$ and the approximation is based on the idea that $\bar{g}$ is a vector of numbers close to one, whereas Σ is much smaller with numbers of the order of 0.10^2 when monthly volatility is around 10%.

Lemma 3 *Suppose that Λ is diagonal. In the case when $G = \xi\xi'$ for some vector $\xi > 0$, then the unique solution m to (B.46) such that $m\,\text{diag}(\xi)$ has all eigenvalues below one in absolute value and $\tilde{m} \in \mathcal{S}(0,1)$ given by $m = \Lambda^{-1/2}\tilde{m}\Lambda^{1/2}$ where*

$$\tilde{m} \;=\; \text{diag}(\xi)^{-1/2}\; 0.5(\hat{\Sigma} - \left(\hat{\Sigma}^2 - 4I\right)^{1/2})\,\text{diag}(\xi)^{-1/2} \tag{A.2}$$

and $\hat{\Sigma} \;=\; \text{diag}(\xi)^{-1/2}(w^{-1}\Lambda^{-1/2}\gamma\Sigma\Lambda^{-1/2} + \text{diag}((\xi_i^2+1)))\,\text{diag}(\xi)^{-1/2}$ and $\left(\hat{\Sigma}^2 - 4I\right)^{1/2}$ is the unique positive-definite square root.

[18] We abuse the notation and use $\bar{g}$ to denote both the vector and the diagonal matrix.

Proof of Lemma 3. We have that (B.58) takes the form

$$\tilde{m} \;=\; \Big(\Lambda^{-1/2}\Sigma\Lambda^{-1/2} + I + \Lambda^{-1/2}\,\mathrm{diag}(\xi)(\Lambda^{1/2}(I-\tilde{m})\Lambda^{1/2})\,\mathrm{diag}(\xi)\Lambda^{-1/2}\Big)^{-1}. \tag{A.3}$$

and the assumption of a diagonal Λ implies

$$\tilde{m} \;=\; \Big(\Lambda^{-1/2}\Sigma\Lambda^{-1/2} + I + \mathrm{diag}(\xi)(I-\tilde{m})\,\mathrm{diag}(\xi)\Big)^{-1}. \tag{A.4}$$

We abuse the notation and use ξ to denote $\mathrm{diag}(\xi)$. Let $\tilde{\Sigma} \;=\; \Lambda^{-1/2}\Sigma\Lambda^{-1/2} + I + \xi^2$. Then, (A.4) takes the form

$$\tilde{m} \;=\; \Big(\tilde{\Sigma} - \xi\tilde{m}\xi\Big)^{-1}. \tag{A.5}$$

Define

$$\hat{\Sigma} \;=\; \xi^{-1/2}\tilde{\Sigma}\xi^{-1/2} \;=\; \xi^{-1/2}\Lambda^{-1/2}\Sigma\Lambda^{-1/2}\xi^{-1/2} + \xi^{-1} + \xi \;>\; 2I\,,$$

where the last inequality follows because $\xi + \xi^{-1} \geq 2$ for any positive number ξ. Let also $\hat{m} = \xi^{1/2}\tilde{m}\xi^{1/2}$. Then, we get

$$\hat{m}^2 - \hat{\Sigma}\hat{m} \;+\; I \;=\; 0 \tag{A.6}$$

which has 2^N solutions. The smallest solution is given by

$$\hat{m} \;=\; 0.5(\hat{\Sigma} - (\hat{\Sigma}^2 - 4I)^{1/2})\,. \tag{A.7}$$

The function $f(x) = 0.5(x - (x^2-4)^{1/2}) = 2/(x + (x^2-4)^{1/2}) < 1$ for all $x > 2$, and the claim follows. □

Starting with this approximation, we can compute the exact m stepwise, as follows. Note first that the set $\mathcal{S}$ of symmetric, positive definite matrices is a partially-ordered set with respect to the positive semi-definite order: We say that $m_1 \leq m_2$ if $m_2 - m_1$ is positive semi-definite. Further, we let $\mathcal{S}(0,1)$ be the set of positive semi-definite matrices with eigenvalues between zero and one.

Suppose for simplicity that Λ is diagonal. Since the optimum is unique, the proof of Proposition 2 implies that (B.58) has a unique solution $\tilde{m} \in \mathcal{S}(0,1)$. Remarkably, this solution automatically satisfies the transversality condition $\bar{g}^{1/2}\tilde{m}\bar{g}^{1/2} \in \mathcal{S}(0,1)$. The following lemma shows how to construct this unique solution.

Lemma 4 *The unique solution $\tilde{m} \in \mathcal{S}(0,1)$ to (B.58) can be computed as follows: $m = \Lambda^{-1/2}\tilde{m}\Lambda^{1/2}$, where $\tilde{m}$ can be found by iterating the mapping F:*

$$F(\tilde{m}) \;=\; \Big(w^{-1}\Lambda^{-1/2}\gamma\Sigma\Lambda^{-1/2} + I + \Lambda^{-1/2}((\Lambda^{1/2}(I-\tilde{m})\Lambda^{1/2}) \circ G)\Lambda^{-1/2}\Big)^{-1}. \tag{A.8}$$

Indeed, F maps $\mathcal{S}(0,1)$ into itself, is monotonic with respect to the positive semi-definite order. Furthermore, it has a unique fixed point $\tilde{m}_* \in \mathcal{S}(0,1)$, and its iterations converge to this unique fixed point from any starting point m_0 in $\mathcal{S}(0,1)$ satisfying either $m_0 \leq F(m_0)$ or

$m_0 \geq F(m_0)$. *In particular, it converges upward from the smallest starting point* $0 \in \mathcal{S}(0,1)$:

$$F(0) \leq F(F(0)) \;\leq\; \cdots \;\leq\; F(\cdots(F(0))) \;\to\; \tilde{m}_*$$

Furthermore, the map F *is monotone decreasing in the matrix* G *in the sense of positive semi-definite order:* $\tilde{m}(G_1) \;\leq\; \tilde{m}(G_2)$ *whenever* $G_1 \geq G_2$. *In particular, if* $G = \bar{g}\bar{g}' + \frac{1}{(1+r^f+\bar{\mu})^2}\Sigma$ *with* $\bar{g} \;=\; (1+r^f+\bar{\mu})^{-1}(1+r^f+E[\mu])$, *and if* Λ *is diagonal, then it also converges downward from the starting point* $\tilde{m}(\bar{g})$ *from Lemma 3:*

$$F(\tilde{m}(\bar{g})) \geq F(F(\tilde{m}(\bar{g}))) \;\geq\; \cdots \;\geq\; F(\cdots(F(\tilde{m}(\bar{g})))) \;\to\; \tilde{m}_* \tag{A.9}$$

so the first iterations of these provide lower and upper bounds:

$$\begin{aligned} &\left(w^{-1}\Lambda^{-1/2}\gamma\Sigma\Lambda^{-1/2} + \mathrm{diag}((G_{i,i}+1))\right)^{-1} \\ &\leq \tilde{m}_* \;\leq\; \left(w^{-1}\Lambda^{-1/2}\gamma\Sigma\Lambda^{-1/2} + I + ((I-\tilde{m}(\bar{g}))\circ G)\right)^{-1}. \end{aligned} \tag{A.10}$$

Proof. The only claim that requires proof is the fact that $\tilde{m}(\bar{g}) \;\geq\; F(\tilde{m}(\bar{g}); G)$. Indeed, by the monotonicity of F in G we have

$$\tilde{m}(\bar{g}) \;=\; F(\tilde{m}(\bar{g}); \bar{g}\bar{g}') \;\geq\; F(\tilde{m}(\bar{g}); G)\,,$$

and the claim follows. This inequality, combined with monotonicity, implies the required sequence of inequalities (A.9). □

A.2 Implementation Details for Multiperiod-ML

Suppose that Λ is diagonal. While formula (21) requires an infinite sum, in our numerical implementation, we use the approximation

$$\begin{aligned} A_t \;&=\; (I-m)^{-1}(I-(m\,\mathrm{diag}(\bar{g})))^2(I-(m\,\mathrm{diag}(\bar{g})))^{-2}\sum_{\tau=0}^{\infty}(m\,\mathrm{diag}(\bar{g}))^\tau c\Sigma^{-1}E_t[r_{t+1+\tau}] \\ &=\; (I-m)^{-1}(I-(m\,\mathrm{diag}(\bar{g})))^2\sum_{\tau=0}^{\infty}(m\,\mathrm{diag}(\bar{g}))^\tau(I-(m\,\mathrm{diag}(\bar{g})))^{-2}c\Sigma^{-1}E_t[r_{t+1+\tau}] \\ &=\; (I-m)^{-1}(I-(m\,\mathrm{diag}(\bar{g})))(I-(m\,\mathrm{diag}(\bar{g})))\sum_{\tau=0}^{\infty}(m\,\mathrm{diag}(\bar{g}))^\tau\,\widetilde{c}\,\Sigma^{-1}E_t[r_{t+1+\tau}] \\ &\approx\; (I-m)^{-1}(I-(m\,\mathrm{diag}(\bar{g})))^2(I-(m\,\mathrm{diag}(\bar{g}))^{k+1})^{-1}\sum_{\tau=0}^{k}(m\,\mathrm{diag}(\bar{g}))^\tau\,\widetilde{c}\,\Sigma^{-1}E_t[r_{t+1+\tau}] \end{aligned} \tag{A.11}$$

where we have defined

$$\widetilde{c} \;=\; (I-(m\,\mathrm{diag}(\bar{g})))^{-2}c\,. \tag{A.12}$$

B Proofs

B.1 Properties of the Implementable Efficient Frontier

We note that the textbook efficient frontier is usually defined in terms of a risk-minimization problem rather than the return-maximization in (10). Hence, we could consider the corresponding definition of the implementable efficient frontier as the combination of volatilities and expected net returns, $(\sigma(k), k)_{k\geq 0}$, such that risk is minimal for that level of net return:

$$\sigma(k)^2 = \min_{\pi_t} E\left[\pi_t' \Sigma \pi_t\right] \quad \text{s.t. } E\left[r_{t+1}^{\pi,net}\right] = k \tag{B.1}$$

However, this definition is less helpful for two reasons. First, no solution exists for large enough k. Second, this definition cannot produce the downward-sloping portion of the frontier seen in Figure 1.

Proof of Proposition 1. (i) We first show that the net Sharpe ratio is decreasing along the frontier. To see that, consider two risk levels, $\sigma_1 < \sigma_2$. Let π^2 be the frontier portfolio corresponding to σ_2. This portfolio has the highest expected net return for this risk level and the highest net Sharpe ratio. If we scale down this portfolio to $\pi^1 = \frac{\sigma_1}{\sigma_2}\pi^2$ (putting the rest of the money in the risk-free asset), then the risk becomes $\frac{\sigma_1}{\sigma_2}\sigma^2 = \sigma_1$. Then we have:

$$\max_{\pi_t \in \Pi \text{ s.t. } E[\pi_t'\Sigma\pi_t]=\sigma_1^2} E\left[r_{t+1}^{\pi,net}\right] \geq E\left[r_{t+1}^{\pi^1,net}\right] > \frac{\sigma_1}{\sigma_2} E\left[r_{t+1}^{\pi^2,net}\right] \tag{B.2}$$

Here, the first inequality follows from the definition of the frontier as the maximum. The second inequality follows from the fact that gross returns are linear, but transaction costs are quadratic and $\frac{\sigma_1}{\sigma_2} < 1$. Dividing both sides of (B.7) by σ_1, we see that the net Sharpe ratio is decreasing in σ.

(vi) Consider the implementable efficient frontier corresponding to a wealth of w_1 and w_2, where $w_1 < w_2$. Take a point on the frontier of w_2 corresponding to the risk of σ and a portfolio π^2. Then with wealth w_1, the portfolio π^2 delivers a higher net return with the same risk (and there may exist another portfolio with an even higher net return for this level of wealth), so the frontier of w_1 must be above that of w_2.

(ii)–(iii) To prove the concavity of the implementable efficient frontier, consider two risk levels, $\sigma_1,\ \sigma_2$. Let $\pi(\sigma)$ be the frontier portfolio corresponding to σ, and let

$$R(\sigma) \ = \ \max_{\pi_t \in \Pi \text{ s.t. } E[\pi_t'\Sigma\pi_t]=\sigma} E\left[r_{t+1}^{\pi,net}\right] \tag{B.3}$$

Define

$$\tilde{\pi} \ = \ 0.5(\pi(\sigma_1) + \pi(\sigma_2))\,. \tag{B.4}$$

Then,

$$\begin{aligned} E[\tilde{\pi}'\Sigma\tilde{\pi}] \ &= \ E[0.25((\pi(\sigma_1)'\Sigma\pi(\sigma_1) + (\pi(\sigma_2)'\Sigma\pi(\sigma_2) + 2(\pi(\sigma_1)'\Sigma\pi(\sigma_2))] \\ &\leq \ 0.25(\sigma_1^2 + \sigma_2^2 + 2\sigma_1\sigma_2) \ = \ (0.5(\sigma_1+\sigma_2))^2\,, \end{aligned} \tag{B.5}$$

where we have used a modified Cauchy-Schwarz inequality

$$\begin{aligned} &E[(\pi(\sigma_1)'\Sigma\pi(\sigma_2)] \;\leq\; E[((\pi(\sigma_1)'\Sigma\pi(\sigma_1))^{1/2}((\pi(\sigma_2)'\Sigma\pi(\sigma_2))^{1/2}] \\ &\leq\; E[((\pi(\sigma_1)'\Sigma\pi(\sigma_1))]^{1/2}E[((\pi(\sigma_2)'\Sigma\pi(\sigma_2))]^{1/2} \;\leq \sigma_1\sigma_2 \end{aligned} \tag{B.6}$$

Therefore,

$$\begin{aligned} &0.5(R(\sigma_1)+R(\sigma_2)) \;=\; 0.5(E\left[r_{t+1}^{\pi(\sigma_1),net}\right] \;+\; E\left[r_{t+1}^{\pi(\sigma_1),net}\right]) \\ &=\; E\left[r'_{t+1}\tilde{\pi} \;-\; 0.5(TC^{\pi(\sigma_1)}+TC^{\pi(\sigma_2)})\right]\,. \end{aligned} \tag{B.7}$$

Since transaction costs are convex in π_t, we have

$$-0.5(TC^{\pi(\sigma_1)}+TC^{\pi(\sigma_2)}) \;\leq\; -TC^{\tilde{\pi}}\,.$$

Thus,

$$E\left[r_{t+1}^{\tilde{\pi},net}\right] \;\geq\; 0.5(R(\sigma_1)+R(\sigma_2))\,,$$

while

$$\sigma(\tilde{\pi}) \;=\; (E[\tilde{\pi}'\Sigma\tilde{\pi}])^{1/2} \;\leq\; 0.5(\sigma_1+\sigma_2)\,.$$

Thus, we get $R(\sigma(\tilde{\pi})) \geq 0.5(R(\sigma_1)+R(\sigma_2))$, and hence, the required concavity follows if $R(\sigma)$ is increasing on $[0.5(\sigma_1+\sigma_2), \sigma(\tilde{\pi})]$.

Now, pick a $\gamma > 0$. Then, clearly, π^γ belongs to the efficient frontier, corresponding to some $\sigma(\gamma)$: Otherwise, we could increase net expected return keeping the variance fixed. We also note that for large γ, the effect of transaction costs is negligible and, hence, the efficient frontier for $\sigma \approx 0$ approximately coincides with the frictionless one, and hence $k(\sigma)$ is monotone increasing for $\sigma \approx 0$.

The set of eligible portfolios (adapted, square-integrable processes) is a Hilbert space H we can define operators A, B and a vector $x \in H$ so that $E[r^{\pi,net}] \;=\; \langle x,\pi\rangle - 0.5\langle Ax, x\rangle$ and $E[\pi'\Sigma\pi] = \langle \pi B, \pi\rangle$, where $\langle\cdot,\cdot\rangle$ is the inner product in the Hilbert space. We consider finite-dimensional approximations of the quadratic problem and thus assume that A, B are finite-dimensional matrices. Then, the first order condition is

$$x \;-\; A\pi \;=\; \lambda\, B\pi \tag{B.8}$$

where λ is the Lagrange multiplier of the constraint $\langle \pi, B\pi\rangle = \sigma^2$. Now,

$$\pi \;=\; (A+\lambda B)^{-1}x\,, \tag{B.9}$$

and we need to solve the equation

$$\langle (A+\lambda B)^{-1}B(A+\lambda B)^{-1}x, x\rangle = \sigma^2 \tag{B.10}$$

For the increasing part of the frontier, the constraint $\langle \pi, B\pi\rangle \leq \sigma^2$ is binding and $\lambda > 0$. This defines $\sigma_*^2 = \langle A^{-1}BA^{-1}x, x\rangle$.

Beyond that we need to use the eigen-decomposition of $B^{-1/2}AB^{-1/2}$ and define $\tilde{x} = B^{-1/2}x$. Then, if ν_i are the eigenvalues of $BB^{-1/2}AB^{-1/2}$ and $\tilde{x}_i$ are the coordinates of $\tilde{x}$ in

the eigen-basis, we get that we need to maximize

$$\begin{aligned} & \langle x, (A+\lambda B)^{-1}x\rangle - 0.5\langle (A+\lambda B)^{-1}(A+\lambda B-\lambda B)x, (A+\lambda B)^{-1}x\rangle \\ &= 0.5\langle x, (A+\lambda B)^{-1}x\rangle + 0.5\lambda\sigma^2 \\ &= 0.5\sum_i \tilde{x}_i^2(1+\lambda\nu_i)^{-1} + 0.5\lambda\sigma^2 \end{aligned} \tag{B.11}$$

under the constraint

$$\sum_i \tilde{x}_i^2(1+\lambda\nu_i)^{-2} \;=\; \sigma^2\,. \tag{B.12}$$

This function (as a function of λ) explodes for $\lambda = -1/\nu_i$.

Now, for the mean-variance optimization problem, the solution is

$$\pi \;=\; (A+\gamma B)^{-1}x \tag{B.13}$$

and the variance

$$\langle (A+\gamma B)^{-1}x, B(A+\gamma B)^{-1}x\rangle \tag{B.14}$$

is monotone decreasing in γ and converges to σ_*^2 when $\gamma \to 0$. At the same time,

$$R(\gamma) \;=\; \langle (A+\gamma B)^{-1}x, x\rangle \tag{B.15}$$

is also monotone decreasing in γ. □

B.2 Verification Lemma

Our proofs are based on the following auxiliary result.

Lemma 5 *For simplicity, we normalize $\gamma/w = 1$. Let $\bar{\Lambda}_t \;=\; E_t[g_{t+1}\Lambda g_{t+1}]$. For any solution m_t to*

$$m_t \;=\; (\Sigma+\Lambda+\bar{\Lambda}_t)^{-1}\left(E_t[g_{t+1}\Lambda m_{t+1}g_{t+1}]m_t + \Lambda\right), \tag{B.16}$$

define

$$N_{t,t+\tau} \;=\; \prod_{\tau=1}^{\theta} m_{t-\tau+1}\, g_{t-\tau+1} \tag{B.17}$$

and

$$\tilde{N}_{t,t+\tau} \;=\; \prod_{\tau=1}^{\theta} m_{t-\tau+1}\, \Lambda^{-1} g_{t-\tau+1}\Lambda\,. \tag{B.18}$$

Suppose that

$$\sum_{\tau=1}^{\infty} \|E_t[N'_{t,t+\tau} N_{t,t+\tau}]\|^{1/2} < \infty \tag{B.19}$$

and

$$\sum_{\tau=1}^{\infty} \|E_t[\tilde{N}'_{t,t+\tau} \tilde{N}_{t,t+\tau}]\|^{1/2} < \infty \,. \tag{B.20}$$

Define

$$c_t \;=\; m_t \Lambda^{-1} \Sigma \tag{B.21}$$

and

$$Q_t \;=\; E_t \left[\sum_{\tau=0}^{\infty} \tilde{N}_{t,t+\tau} c_{t+\tau} Markowitz_{t+\tau} \right] \tag{B.22}$$

and let

$$\pi_t \;=\; \pi(s_t, s_{t-}) \;=\; \sum_{\theta=0}^{\infty} N_{t-\theta,t} Q(s_{t-\theta}) \,. \tag{B.23}$$

Then, all series converge in L_2 and π_t is optimal among all bounded stationary processes π_t. Furthermore, it satisfies the recursive relationship

$$\pi(s_t, s_{t-}) \;=\; Q(s_t) \;+\; m_t g_t \pi(s_{t-1}, s_{t-1-}) \tag{B.24}$$

Proof of Lemma 5. Due to the strict convexity of the objective, it suffices to verify the first order conditions. Let

$$\mathcal{O}(\pi) = \min_{\pi \in L_2} E\left[-2\mu(s_t)' \pi_t + \pi_t' \Sigma \pi_t + (\pi_t - g_t \pi_{t-1})' \Lambda (\pi_t - g_t \pi_{t-1}) \right] \tag{B.25}$$

Standard convexity arguments imply that it suffices to derive and verify the first order conditions for our candidate solution.

Let s_{t-} denote the history of s_t and $\pi_t = \pi(s_t, s_{t-})$ be a candidate optimal policy and Then, by direct calculation, using the ergodicity property, we get while the law of iterated expectations implies that

$$E[\pi'_{t-1} g_t \Lambda g_t \pi_{t-1}] \;=\; E[\pi'_t g_{t+1} \Lambda g_{t+1} \pi_t] \;=\; E[\pi'_t \hat{\Lambda}_t \pi_t] \,,$$

where

$$\bar{\Lambda}_t \;=\; E_t[g_{t+1} \Lambda g_{t+1}] \,. \tag{B.26}$$

Therefore,

$$\begin{aligned}
&\mathcal{O}(\pi) \\
&= E\Big[-2\,\mu(s_t)'\pi(s_t,s_{t-}) \;+\; \pi(s_t,s_{t-})'(\Sigma+\Lambda+\bar{\Lambda}_t)\pi(s_t,s_{t-})\Big] \\
&-2E[\pi(s_t,s_{t-})'g_{t+1}\Lambda\pi(s_{t+1},s_{t+1-})]\,,
\end{aligned} \tag{B.27}$$

In order to compute the first order conditions, we need to calculate the Frechet derivative of (B.27) with respect to π. To this end, we consider a small perturbation $\pi \to \pi + \varepsilon Y$ and calculate the first order term in ε, so that

$$\mathcal{O}(\pi_t + \varepsilon Y_t) \;=\; \mathcal{O}(\pi_t) \;+\; \varepsilon\, E[\mathcal{D}(\pi_t)'\, Y_t] \;+\; O(\varepsilon^2) \tag{B.28}$$

and $\mathcal{D}(\pi)$ is the Frechet derivative. To this end, we compute

$$\begin{aligned}
&E[\mu(s_t)'(\pi(s_t,s_{t-})+\varepsilon Y_t)] \;=\; E[\mu(s_t)'\pi(s_t,s_{t-})] + \varepsilon E[\mu(s_t)'Y_t] \\
&E[(\pi(s_t,s_{t-})+\varepsilon Y_t)'(\Sigma+\Lambda+\bar{\Lambda}_t)(\pi(s_t,s_{t-})+\varepsilon Y_t)] \\
&=\; E[\pi(s_t,s_{t-})'(\Sigma+\Lambda+\bar{\Lambda}_t)\pi(s_t,s_{t-})] \\
&+\; 2\varepsilon\, E[\pi(s_t,s_{t-})'(\Sigma+\Lambda+\bar{\Lambda})Y_t] \;+\; O(\varepsilon^2) \\
&E[(\pi(s_t,s_{t-})+Y_t)'g_{t+1}\Lambda(\pi(s_{t+1},s_{t+1-})+Y_{t+1})] \\
&=\; E[\pi(s_t,s_{t-})'g_{t+1}\Lambda\pi(s_{t+1},s_{t+1-})] \\
&+\; \varepsilon\Big(E[Y_t'E_t[g_{t+1}\Lambda\pi(s_{t+1},s_{t+1-})]] \;+\; E[\pi(s_t,s_{t-})'g_{t+1}\Lambda Y_{t+1}]\Big)
\end{aligned} \tag{B.29}$$

Furthermore, by stationarity,

$$E[\pi(s_t,s_{t-})Y_{t+1}] \;=\; E[\pi(s_{t-1},s_{t-1-})Y_t]\,.$$

We conclude that the Frechet derivative is given by

$$\begin{aligned}
\mathcal{D}(\pi) \;&=\; -2\mu(s_t) \;+\; 2(\Sigma+\Lambda+\bar{\Lambda})\pi(s_t,s_{t-}) \\
&-\; 2E_t[g_{t+1}\Lambda\pi(s_{t+1},s_{t+1-})] \;-\; 2\Lambda g_t\pi(s_{t-1},s_{t-1-})\,,
\end{aligned} \tag{B.30}$$

where $\bar{\Lambda}(s_t) = E_t[\Lambda(s_{t+1})]$, implying that a bounded π is optimal if it satisfies the *integral equation*

$$\pi(s_t,s_{t-}) \;=\; (\Sigma+\Lambda+\bar{\Lambda}_t)^{-1}\Big(\mu(s_t) \;+\; E_t[g_{t+1}\Lambda\pi(s_{t+1},s_{t+1-})] \;+\; \Lambda g_t\pi(s_{t-1},s_{t-1-})\Big)\,. \tag{B.31}$$

Substituting the Ansatz

$$\pi_t \;=\; Q_t \;+\; m g_t \pi_{t-1} \tag{B.32}$$

into this equation, we get

$$\begin{aligned} & Q_t \;+\; m_t g_t \pi_{t-1} \\ &= \; (\Sigma + \Lambda + \bar{\Lambda}_t)^{-1} \Bigg(\mu(s_t) \\ &+ \; E_t[g_{t+1}\Lambda(Q_{t+1} + m_{t+1}g_{t+1}Q_t + m_{t+1}g_{t+1}m_t g_t \pi_{t-1})] \;+\; \Lambda g_t \pi_{t-1} \Bigg) . \end{aligned} \tag{B.33}$$

Equating the coefficients on π_{t-1} gives an integral equation for m_t :

$$m_t g_t \;=\; (\Sigma + \Lambda + \bar{\Lambda}_t)^{-1} \Bigg(E_t[g_{t+1}\Lambda m_{t+1} g_{t+1} m_t g_t] + \Lambda g_t \Bigg) , \tag{B.34}$$

and (B.33) turns into an integral equation for Q_t :

$$Q_t \;=\; (\Sigma + \Lambda + \bar{\Lambda}_t)^{-1} \Bigg(\mu(s_t) \;+\; E_t[g_{t+1}\Lambda(Q_{t+1} + m_{t+1}g_{t+1}Q_t)] \Bigg) . \tag{B.35}$$

Dividing by g_t, we get that (B.34) turns into the required equation (B.16). Furthermore, we can rewrite (B.34) as

$$(I - (\Sigma + \Lambda + \bar{\Lambda}_t)^{-1} E_t[g_{t+1}\Lambda m_{t+1} g_{t+1}]) m_t \;=\; (\Sigma + \Lambda + \bar{\Lambda}_t)^{-1}\Lambda \, , \tag{B.36}$$

which implies

$$(I - (\Sigma + \Lambda + \bar{\Lambda}_t)^{-1} E_t[g_{t+1}\Lambda m_{t+1} g_{t+1}]) \;=\; (\Sigma + \Lambda + \bar{\Lambda}_t)^{-1}\Lambda m_t^{-1} \, . \tag{B.37}$$

After a few algebraic transformations, we get that (B.35) is equivalent to

$$(I - (\Sigma + \Lambda + \bar{\Lambda}_t)^{-1} E_t[g_{t+1}\Lambda m_{t+1} g_{t+1}]) Q_t \;=\; (\Sigma + \Lambda + \bar{\Lambda}_t)^{-1} \Bigg(\mu(s_t) \;+\; E_t[g_{t+1}\Lambda Q_{t+1}] \Bigg) . \tag{B.38}$$

Substituting from (B.37), we can rewrite (B.38) as

$$Q_t \;=\; m_t \Lambda^{-1} \Bigg(\mu(s_t) \;+\; E_t[g_{t+1}\Lambda Q_{t+1}] \Bigg) . \tag{B.39}$$

Defining

$$c_t \;=\; M_t \Lambda_t^{-1} \Sigma_t , \tag{B.40}$$

we get

$$Q_t \;=\; c_t \, Markowitz_t \;+\; E_t[m_t \Lambda^{-1} g_{t+1} \Lambda Q_{t+1}] \, . \tag{B.41}$$

Iterating this equation, we get the required up to the convergence statement. Convergence

in L_2 follows directly from the made assumptions. Indeed,

$$E[E_t[X]^2] \;\leq\; E[X^2]\,, \tag{B.42}$$

and hence we can ignore $E_t[]$ when proving convergence. Furthermore, by the made uniform boundedness assumptions and the uniform positive-definiteness of Σ_t, we have

$$\|q_{t+\tau}\| \;=\; \|c_{t+\tau} Markowitz_{t+\tau}\| \;\leq\; K$$

for some $K > 0$, almost surely. Since

$$\|Q_t\| \;=\; \|E_t\left[\sum_{\tau=0}^{\infty} N_{t,t+\tau} c_{t+\tau} Markowitz_{t+\tau}\right]\| \;\leq\; \sum_{\tau} \|N_{t,t+\tau} q_{t+\tau}\|\,, \tag{B.43}$$

to prove the convergence of Q_t it suffices to show that

$$\sum_{\tau} E[q'_{t+\tau} N'_{t,t+\tau} N_{t,t+\tau} q_{t+\tau}]^{1/2} \;<\; \infty \tag{B.44}$$

which follows from the made assumptions. Convergence of the series representation for π_t also follows from the made assumptions. □

Recall that g_t is the diagonal matrix of $vec(g_t)$ on the diagonal. For the case when μ is constant, we have that

$$G_t = E_t[vac(g_{t+1})vec(g_{t+1})'] = (1+g_t^w)^{-2}(\Sigma + (1+r_t^f+\mu_t)(1+r_t^f+\mu_t)')\,. \tag{B.45}$$

When $g_t^w,\ r_t^f$, and μ_t are all constant, we get that $G_t = G$ is also constant, and we arrive at the following result, which is a direct consequence of Lemma 5.

Lemma 6 *For simplicity, we normalize $\gamma/w = 1$. Suppose that $\mu_t = \mu$, g_t^w, r_t^f are constant. Let $\bar{\Lambda} \;=\; \Lambda \circ G$. For any solution m to*

$$m \;=\; (\Sigma + \Lambda + \bar{\Lambda})^{-1}\left(((\Lambda m) \circ G)m + \Lambda\right), \tag{B.46}$$

define

$$N_{t,t+\tau} \;=\; \prod_{\tau=1}^{\theta} m\, g_{t-\tau+1} \tag{B.47}$$

and

$$\tilde{N}_{t,t+\tau} \;=\; \prod_{\tau=1}^{\theta} m\, \Lambda^{-1} g_{t-\tau+1} \Lambda\,. \tag{B.48}$$

Suppose that

$$\sum_{\tau=1}^{\infty} \|E_t[N'_{t,t+\tau} N_{t,t+\tau}]\|^{1/2} < \infty \tag{B.49}$$

and

$$\sum_{\tau=1}^{\infty} \|E_t[\tilde{N}'_{t,t+\tau}\tilde{N}_{t,t+\tau}]\|^{1/2} < \infty\,. \tag{B.50}$$

Define

$$c \;=\; m\Lambda^{-1}\Sigma \tag{B.51}$$

and

$$Q_t \;=\; (I - m\Lambda^{-1}\bar{g}\Lambda)^{-1} cMarkowitz \tag{B.52}$$

and let

$$\pi_t \;=\; \pi(s_t, s_{t-}) \;=\; \sum_{\theta=0}^{\infty} N_{t-\theta,t} Q(s_{t-\theta})\,. \tag{B.53}$$

Then, all series converge in L_2, and π_t is optimal among all bounded stationary processes π_t. Furthermore, it satisfies the recursive relationship

$$\pi(s_t, s_{t-}) \;=\; Q(s_t) \;+\; m g_t \pi(s_{t-1}, s_{t-1-})\,. \tag{B.54}$$

B.3 Properties the Discount Factor m

This section shows some useful properties of the discount factor m solving (B.46). We start with the following observation

Lemma 7 *For simplicity, we normalize $\gamma/w = 1$. A matrix-valued function $m(s_t) = m_t$ solves*

$$m_t \;=\; (\Sigma + \Lambda + \bar{\Lambda}_t)^{-1}\left(E_t[g_{t+1}\Lambda m_{t+1} g_{t+1}]m_t + \Lambda\right) \tag{B.55}$$

if and only if $\tilde{m}_t \;=\; \Lambda^{1/2} m_t \Lambda^{-1/2}$ solves

$$\tilde{m}_t \;=\; \left(\Lambda^{-1/2}\Sigma\Lambda^{-1/2} + I + \Lambda^{-1/2}E_t[g_{t+1}\Lambda^{1/2}(I - \tilde{m}_{t+1})\Lambda^{1/2} g_{t+1}]\Lambda^{-1/2}\right)^{-1}. \tag{B.56}$$

In particular, if G_t is constant, then a matrix m solves

$$m \;=\; (\Sigma + \Lambda + \bar{\Lambda})^{-1}\left(((\Lambda m) \circ G)m + \Lambda\right) \tag{B.57}$$

with $\bar{\Lambda} = \Lambda \circ G$ if and only if the matrix $\tilde{m} = \Lambda^{1/2} m \Lambda^{-1/2}$ solves

$$\tilde{m} \;=\; \left(\Lambda^{-1/2}\Sigma\Lambda^{-1/2} + I + \Lambda^{-1/2}(G \circ (\Lambda^{1/2}(I - \tilde{m})\Lambda^{1/2}))\Lambda^{-1/2}\right)^{-1}. \tag{B.58}$$

Proof of Lemma 7. We have

$$(\Sigma + \Lambda + \bar{\Lambda}_t)m_t \;=\; \left(E_t[g_{t+1}\Lambda m_{t+1} g_{t+1}]m_t + \Lambda \right). \tag{B.59}$$

Writing $m_t \;=\; \Lambda^{-1/2}\tilde{m}_t\Lambda^{1/2}$, we get

$$(\Sigma + \Lambda + \bar{\Lambda}_t)\Lambda^{-1/2}\tilde{m}_t\Lambda^{1/2} \;=\; \left(E_t[g_{t+1}\Lambda\Lambda^{-1/2}\tilde{m}_{t+1}\Lambda^{1/2} g_{t+1}]\Lambda^{-1/2}\tilde{m}_t\Lambda^{1/2} + \Lambda \right). \tag{B.60}$$

Multiplying by $\Lambda^{-1/2}\tilde{m}_t^{-1}$ from the right and by $\Lambda^{-1/2}$ from the left, we get

$$\Lambda^{-1/2}(\Sigma + \Lambda + \bar{\Lambda}_t)\Lambda^{-1/2} \;=\; \left(\Lambda^{-1/2} E_t[g_{t+1}\Lambda^{1/2}\tilde{m}_{t+1}\Lambda^{1/2} g_{t+1}]\Lambda^{-1/2} + \tilde{m}_t^{-1} \right). \tag{B.61}$$

This is equivalent to

$$\tilde{m}_t^{-1} \;=\; \Lambda^{-1/2}(\Sigma + \Lambda + \bar{\Lambda}_t - E_t[g_{t+1}\Lambda^{1/2}\tilde{m}_{t+1}\Lambda^{1/2} g_{t+1}])\Lambda^{-1/2}\,, \tag{B.62}$$

which is, in turn, equivalent to

$$\tilde{m}_t \;=\; \left(\Lambda^{-1/2}\Sigma\Lambda^{-1/2} + I + \Lambda^{-1/2} E_t[g_{t+1}\Lambda^{1/2}(I - \tilde{m}_{t+1})\Lambda^{1/2} g_{t+1}]\Lambda^{-1/2} \right)^{-1}. \tag{B.63}$$

In the case when $m_t = m$ is constant and $G_t = G$ is constant, we get

$$E_t[g_{t+1}\Lambda^{1/2}(I - \tilde{m})\Lambda^{1/2} g_{t+1}] \;=\; G \circ (\Lambda^{1/2}(I - \tilde{m})\Lambda^{1/2})\,,$$

and we get the required. □

Recall that $\mathcal{S}(0,1)$ is the set of symmetric, positive semi-definite matrices with eigenvalues in $(0,1)$.

Proposition 8 *Suppose that there exists a solution $\tilde{m} \in \mathcal{S}(0,1)$ to (B.58). Let $q_* < 1$ be the largest eigenvalue of $\tilde{m}$. Let $m = \Lambda^{-1/2}\tilde{m}\Lambda^{1/2}$ and*

$$\Pi_{t-\theta,t} \;=\; \left(\prod_{\tau=1}^{\theta} m\, g_{t-\tau+1} \right). \tag{B.64}$$

Then,

$$\lim_{\theta\to\infty} q_*^{-\theta} E[\|\Pi_{t-\theta,t}(\nu)\|^2] \;=\; 0\,. \tag{B.65}$$

Similarly, if we define

$$\hat{\Pi}_{t,t+\theta} \;=\; \prod_{\tau=1}^{\theta} (m\Lambda^{-1} g_{t+\tau-1}\Lambda)\,.$$

Then,

$$\lim_{\theta\to\infty} q_*^{-\theta} E[\|\hat{\Pi}_{t,t+\theta}\|^2] \;=\; 0\,. \tag{B.66}$$

Proof of Proposition 8. For simplicity, we normalize $\gamma/w = 1$. Recall that $\mathcal{S}$ is the set of symmetric matrices, and $\mathcal{S}(a,b)$ is the set of positive semi-definite matrices with eigenvalues between a and b.

Equation (B.58) can be rewritten as

$$\tilde{m}\Big(\Lambda^{-1/2}\Sigma\Lambda^{-1/2} + I + \Lambda^{-1/2}E_t[g_{t+1}\Lambda^{1/2}(I-\tilde{m}_{t+1})\Lambda^{1/2}g_{t+1}]\Lambda^{-1/2}\Big)\tilde{m} \;=\; \tilde{m}\,. \tag{B.67}$$

Define the map

$$\Xi(Z) \;=\; \tilde{m}\Lambda^{-1/2}E_t[g_{t+1}\Lambda^{1/2}Z\Lambda^{1/2}g_{t+1}]\Lambda^{-1/2}\tilde{m} \tag{B.68}$$

mapping the cone of positive semi-definite matrices into itself. Then, (B.67) implies that

$$\Xi(I-\tilde{m}) \;=\; \tilde{m} - \tilde{m}^2 - \tilde{m}\Lambda^{-1/2}\Sigma\Lambda^{-1/2}\tilde{m} \;<\; \tilde{m}(I-\tilde{m}) \;\le\; q_*(I-\tilde{m})\,. \tag{B.69}$$

Now, the map Ξ leaves the proper cone $\mathcal{S}(0,+\infty)$ invariant, and hence, by the Krein and Rutman (1950) theorem, its spectral radius corresponds to a strictly positive eigenvalue $\lambda_* > 0$. Let $Z \in \mathcal{S}(0,+\infty)$ be the corresponding eigenvector. Then,

$$\Xi(Z) \;=\; \lambda_*\, Z\,.$$

Since $I-\tilde{m}$ is strictly positive definite, there exists a constant $a_* > 0$ such that $aZ \le I-\tilde{m}$ if and only if $a \le a_*$. Then,

$$\lambda a_* Z = \Xi(a_* Z) \;\le\; \Xi(I-\tilde{m}) \;<\; q_*\,(I-\tilde{m})$$

Thus, $(\lambda a_*/q_*)Z \;\le\; I-\tilde{m}$ implying that, by the definition of a_*, we must have $\lambda/q_* < 1$.

Note also that the transformation

$$\hat{\Xi}(Z) \;=\; A^{-1}\tilde{m}E_t[\Lambda^{-1/2}g_{t+1}\Lambda^{1/2}AZA'\Lambda^{1/2}g_{t+1}\Lambda^{-1/2}]\tilde{m}(A')^{-1}$$

is similar to Ξ for any invertible matrix A. Hence, Ξ and $\hat{\Xi}$ have the same spectral radius. Pick $A = \tilde{m}\Lambda^{-1/2}$. Then,

$$\hat{\Xi}(Z) \;=\; E_t[g_{t+1}\Lambda^{1/2}\tilde{m}\Lambda^{-1/2}Z\Lambda^{-1/2}\tilde{m}\Lambda^{1/2}g_{t+1}] \;=\; E_t[g_{t+1}m'Zmg_{t+1}]\,.$$

By direct calculation,

$$E[\Pi'_{t-\theta,t}\Pi_{t-\theta,t}] \;=\; \hat{\Xi}^\theta(I)\,,$$

and

$$E[\hat{\Pi}_{t,t+\theta}\hat{\Pi}'_{t,t+\theta}] \;=\; \Xi^\theta(I)\,,$$

and the claim follows. □

B.4 Proofs of Propositions 2 and 3

Lemma 8 *For simplicity, we normalize $\gamma/w = 1$. Consider the map F mapping the convex set of $\mathcal{S}(0,1)$-valued matrix functions into itself and defined via*

$$F(\tilde{m}_t) \;=\; \left(\Lambda^{-1/2}\Sigma\Lambda^{-1/2} + I + \Lambda^{-1/2}E_t[g_{t+1}\Lambda^{1/2}(I-\tilde{m}_{t+1})\Lambda^{1/2}g_{t+1}]\Lambda^{-1/2}\right)^{-1}. \tag{B.70}$$

This map is strictly monotone increasing in the positive semi-definite order and hence has at least one fixed point in $\mathcal{S}(0,1)$. The set of fixed points has a unique maximal and a unique minimal element. The minimal element is obtained by iterating F on 0. The maximal element is obtained by iterating F on I.

Proof of Lemma 8. The proof follows directly from the fact that the map $A \to A^{-1}$ is monotone decreasing in the positive semi-definite order, and the same is true for the map $\tilde{m}_{t+1} \to E_t[g_{t+1}\Lambda^{1/2}(I-\tilde{m}_{t+1})\Lambda^{1/2}g_{t+1}]$. □

When μ is stochastic, things are a bit more tricky, as shown by the following lemma.

Lemma 9 *Suppose that $\mu(s_t) \;=\; \epsilon\tilde{\mu}(s_t)$, $g_t^w = g^w + O(\varepsilon)$, $r_t^f = r^f + O(\varepsilon)$. Then,*

$$G_t \;=\; G \;+\; O(\epsilon) \tag{B.71}$$

where $G = E[vec(g_t)vec(g_t)']$ and hence, for every solution $\tilde{m} \in \mathcal{S}(0,1)$ to (B.58) and any sufficiently small $\epsilon > 0$ there exists a unique solution $\tilde{m}_t$ to (B.56) satisfying

$$\tilde{m}_t \;=\; \tilde{m} \;+\; O(\varepsilon)\,. \tag{B.72}$$

Proof of Lemma 9. The proof follows directly from the implicit function theorem and the fact that the map F from Lemma 8 is strictly monotone on $\mathcal{S}(0,1)$ and (by direct calculation) has a non-degenerate Jacobian. □

Proof of Proposition 2. By Lemma 8, there exists a $\tilde{m} \in \mathcal{S}(0,1)$ solving (B.58). By Proposition 8, the technical conditions (B.49) and (B.50) are satisfied. Lemma 7 implies that m solves (B.46) and hence Lemma 6 implies that the policy π is optimal. Its uniqueness follows from the strict concavity of the objective. □

Proof of Proposition 3. By Lemma 8, there exists a $\tilde{m} \in \mathcal{S}(0,1)$ solving (B.58). By Lemma 9, there exists a $\tilde{m}_t$ solving (B.56) satisfying $m_t = \tilde{m} + O(\varepsilon)$. By a small modification of the proof of Lemma 8, the technical conditions (B.19) and (B.20) are satisfied. Lemma 7 implies that m_t solves (B.55) and hence Lemma 5 implies that the policy

$$\pi_t^* \;=\; \pi(s_t, s_{t-}) \;=\; \sum_{\theta=0}^{\infty} N_{t-\theta,t}Q(s_{t-\theta})\,. \tag{B.73}$$

is optimal with

$$Q_t \;=\; E_t\left[\sum_{\tau=0}^{\infty}\tilde{N}_{t,t+\tau}c_{t+\tau}Markowitz_{t+\tau}\right] \tag{B.74}$$

and

$$N_{t,t+\tau} \;=\; \prod_{\tau=1}^{\theta} m_{t-\tau+1}\, g_{t-\tau+1} \tag{B.75}$$

and

$$\tilde{N}_{t,t+\tau} \;=\; \prod_{\tau=1}^{\theta} m_{t-\tau+1}\, \Lambda^{-1} g_{t-\tau+1} \Lambda\,. \tag{B.76}$$

and $c_t = m_t \Lambda^{-1} \Sigma$. Its uniqueness follows from the strict concavity of the objective. Now, substituting $m_t = m + O(\varepsilon)$ into these equations, we get that π_t^* differs from (20) by $O(\epsilon)$ (technical conditions (B.19)-(B.20) ensure that the infinite sum also is $O(\epsilon)$.) The proof is complete. □

B.5 Proofs of Propositions 4 and 5

We have $m = \Lambda^{-1/2} \tilde{m} \Lambda^{1/2}$ and hence m and $\tilde{m}$ have identical eigenvalues, and the claim of Proposition 4 follows from Lemma 4 and the monotonicity of the map F.

The convergence of m to zero when $w \to 0$ follows directly from (B.58). When $w \to \infty$, to prove convergence, we need to show that the technical conditions (B.49) and (B.50) hold uniformly when $w \to \infty$. By Proposition 8, to this end we need to show that q_*, the maximal eigenvalue of $\tilde{m}$ stays uniformly bounded away from 1. This follows from Lemma 4: Since $\tilde{m} \leq \tilde{m}(\bar{g})$, it suffices to establish this fact for $\tilde{m}(\bar{g})$. From the proof of Lemma 3, we have

$$\tilde{m}(\xi) \;=\; \xi^{-1/2} \hat{m} \xi^{-1/2}$$

where

$$\hat{m} \;=\; 0.5(\hat{\Sigma} - (\hat{\Sigma}^2 - 4I)^{1/2}) \;=\; 2(\hat{\Sigma} + (\hat{\Sigma}^2 - 4I)^{1/2})^{-1} \tag{B.77}$$

and

$$\hat{\Sigma} \;=\; \xi^{-1/2} \tilde{\Sigma} \xi^{-1/2} \;=\; \gamma w^{-1} \xi^{-1/2} \Lambda^{-1/2} \Sigma \Lambda^{-1/2} \xi^{-1/2} + \xi^{-1} + \xi\,.$$

and hence $\hat{\Sigma} \to \xi^{-1} + \xi$ when $w \to \infty$. Thus,

$$\tilde{m} \;\to\; 0.5 \xi^{-1} (\xi^{-1} + \xi - |\xi - \xi^{-1}|)$$

Suppose that $\xi > 1$. Then, we get $\tilde{m} = \xi^{-2} < 1$. If $\xi < 1$, then we get $\tilde{m} \to 1$.

B.6 On the Optimality of Portfolio-ML

Given a function $A(\cdot)$, denote

$$\Pi_t^*(A(\cdot),\ \vec{s}) \;=\; \sum_{\theta=0}^{\infty} \left(\prod_{\tau=1}^{\theta} m\, g_{t-\tau+1} \right) (I - m) A(s_{t-\theta})\,. \tag{B.78}$$

We formalize the above observations in the following lemma.

Lemma 10 *Suppose that $\mu(s_t) = \varepsilon\tilde{\mu}(s_t)$ where ε is a small number. Then, the solution to the aim optimization problem*

$$\begin{aligned}\max_{A(\cdot)} E\Bigg[&\mu(s_t)'\Pi_t^*(A(\cdot),\ \vec{s})\\ &-\frac{w}{2}\Big(\Pi_t^*(A(\cdot),\ \vec{s}_t) - g_t\Pi_t^*(A(\cdot),\ \vec{s}_{t-1})\Big)'\Lambda\Big(\Pi_t^*(A(\cdot),\ \vec{s}_t) - g_t\Pi_t^*(A(\cdot),\ \vec{s}_{t-1})\Big)\\ &-\frac{\gamma}{2}\left(\Pi_t^*(A(\cdot),\ \vec{s}_t)\right)'\Sigma\left(\Pi_t^*(A(\cdot),\ \vec{s}_t)\right)\Bigg]\end{aligned} \tag{B.79}$$

coincides with (21) up to terms of the order ε^2.

Lemma 10 reduces the infeasible (due to infinite history dependence) portfolio optimization problem (9) to a feasible aim optimization problem (B.79), where only the function $A(s_t)$ of the current state needs to be optimized. As we now show, it is possible to use machine learning methods to further reduce (B.79) to a linear-quadratic problem that can be solved analytically. We will need the following assumption.

Assumption 1 *Let s_{-i} denote the vector of signals for all stocks except i. There exists a function $a(s_i, s_{-i})$ such that $(A(s_t))_i = a_i(s_{i,t},\ s_{-i,t})$.*

Assumption 1 is not restrictive and naturally holds whenever $\Lambda_{i,j}$ and $\Sigma_{i,j}$ only depend on s_i, s_j for any pair of stocks i, j. It ensures that the dependence of the aim on signals is the same for all stocks.

Assumption 2 *Suppose that a family of functions $\{f_k(s)\}_{k\geq 1}$ has the universal approximation property: For any $\epsilon > 0$ there exists a $P > 0$ and a vector $\beta \in \mathbb{R}^P$ such that*

$$\|a(s) - \sum_k \beta_k f_k(s)\|_2 \ \leq\ \epsilon\,. \tag{B.80}$$

Let now $F(s) \ = \ (f_k(s_i, s_{-i}))_{i,k=1}^{n,P}$. Then, Assumptions 1 and 2 imply the existence of a vector β such that

$$\|A(s) - F(s)\beta\|_2 \ \leq\ \epsilon\,, \tag{B.81}$$

and hence we can rewrite (B.78) as

$$\pi_t \ = \ X_t\beta \ + \ O(\epsilon)\,, \tag{B.82}$$

where

$$X_t \ \equiv \ \left[\sum_{\theta=0}^{\infty}\left(\prod_{\tau=1}^{\theta} m\, g_{t-\tau+1}\right)(I-m)f(s_{t-\theta})\right]\,. \tag{B.83}$$

Using this formulation for π_t in the objective (B.79), we can rewrite it as

$$\begin{aligned}
&E\left[r'_{t+1}X_t\beta - \frac{\gamma}{2}\beta' X_t'\Sigma X_t\beta - \frac{w}{2}\left(X_t\beta - g_t X_{t-1}\beta\right)'\Lambda\left(X_t\beta - g_t X_{t-1}\beta\right)\right] \\
&= \frac{1}{T}E\left[\underbrace{r'_{t+1}X_t}_{\equiv\tilde{r}'_{t+1}}\beta - \frac{1}{2}\beta'\underbrace{\left[\gamma X_t'\Sigma X_t + w(X_t - g_t X_{t-1})'\Lambda(X_t - g_t X_{t-1})\right]}_{\equiv\tilde{\Sigma}_t}\beta\right] \\
&\equiv E[\tilde{r}'_{t+1}]\beta - \frac{1}{2}\beta' E[\tilde{\Sigma}_t]\beta\,.
\end{aligned} \tag{B.84}$$

and the optimal β is given by

$$\beta^* \;=\; E[\tilde{\Sigma}_t]^{-1}E[\tilde{r}_{t+1}] \tag{B.85}$$

Uniform boundedness of all coefficients implies that the solution to the approximate optimization problem achieves approximate optimum in (B.79). Thus, we can maximize utility by maximizing this quadratic equation in the unknown parameter vector β.

Proposition 9 (Portfolio-ML) *Suppose that $\mu(s_t) = \varepsilon\tilde{\mu}(s_t)$ where ε is a small number. Let β_T be a finite sample counter-part of (B.85). Then, in the limit as $T \to \infty$, β_T converges to β^*. Furthermore, the optimal portfolio (24) achieves the optimal utility (9) up to an error of the order $\varepsilon^2 + \epsilon$, where ϵ is defined in (B.80).*

B.7 Proofs related to Economic Feature Importance

The optimal portfolio admits a simpler analytical expression when transaction costs are small as seen in the following result.

Proposition 10 *Suppose that $\|\Lambda\|$ is small and Λ is diagonal. Then, the aim portfolio A_t is given by*

$$\begin{aligned}
A_t \;&=\; Markowitz_t \\
&+\; \Sigma^{-1}E_t[\Lambda g_{t+1}Markowitz_{t+1} - g_{t+1}\Lambda g_{t+1}Markowitz_t)] \;+\; O(\|\Lambda\|^2)\,,
\end{aligned} \tag{B.86}$$

while investor's utility is given by the same expression as in Proposition 7.

Proof of Proposition 7 and Proposition 10. Under the made assumption, we have

$$m_t \;=\; \Sigma^{-1}\Lambda - \Sigma^{-1}(\Lambda + E_t[g_{t+1}\Lambda g_{t+1}])\Sigma^{-1}\Lambda \;+\; O(\varepsilon^3)$$

and therefore

$$c_t = m_t\Lambda^{-1}\Sigma_t. \;=\; I - \Sigma^{-1}(\Lambda + E_t[g_{t+1}\Lambda g_{t+1}]) \;+\; O(\varepsilon^2)\,.$$

Let $\nu_t = Markowitz_t$. Then,

$$\begin{aligned}
A_t &= (I - m_t)^{-1} \sum_{\tau=0}^{\infty} E_t \left[M_{t,t+\tau}\, c_{t+\tau}\, Markowitz_{t+\tau} \right] \\
&= (I - m_t)^{-1}(c_t \nu_t + E_t[m_t \Lambda^{-1} \Lambda g_{t+1} c_{t+1} \nu_{t+1}]) \; + \; O(\varepsilon^2) \\
&= (I + \Sigma^{-1}\Lambda)\Big(c_t \nu_t \; + \; E_t[\Sigma^{-1}\Lambda\Lambda^{-1}\Lambda g_{t+1} c_{t+1} \nu_{t+1}]\Big) \; + \; O(\varepsilon^2) \\
&= (I + \Sigma^{-1}\Lambda)\Big(\Big(I - \Sigma^{-1}(\Lambda + E_t[g_{t+1}\Lambda g_{t+1}])\Big)\nu_t \\
&+ \; E_t[\Sigma^{-1}\Lambda\Lambda^{-1}\Lambda g_{t+1}\Big(I - \Sigma^{-1}(\Lambda + E_t[g_{t+1}\Lambda g_{t+1}])\Big)\nu_{t+1}]\Big) \; + \; O(\varepsilon^2) \\
&= (I - \Sigma^{-1} E_t[g_{t+1}\Lambda g_{t+1}])\nu_t \; + \; E_t[\Sigma^{-1}\Lambda g_{t+1}\nu_{t+1}] + \; O(\varepsilon^2) \\
&= \nu_t \; + \; \Sigma^{-1} E_t[\Lambda g_{t+1}\nu_{t+1} - g_{t+1}\Lambda g_{t+1}\nu_t)] \; + \; O(\varepsilon^2)\,.
\end{aligned} \tag{B.87}$$

and

$$\begin{aligned}
\pi_t &= m_t g_t \pi_{t-1} \; + \; (I - m_t) A_t \\
&= m_t g_t \nu_{t-1} + (I - m_t)(\nu_t \; + \; \Sigma^{-1} E_t[\Lambda g_{t+1}(\nu_{t+1} - g_{t+1}\nu_t)]) \; + \; O(\varepsilon^2) \\
&= \nu_t \; + \; \Sigma^{-1} E_t[\Lambda g_{t+1}(\nu_{t+1} - g_{t+1}\nu_t)] + \Sigma^{-1}\Lambda(g_t \nu_{t-1} - \nu_t) \; + \; O(\varepsilon^2) \\
&= \nu_t \; + \; \xi_t \; + \; O(\varepsilon^2)\,.
\end{aligned} \tag{B.88}$$

Hence

$$\begin{aligned}
&E[\mu(x_t)'\pi_t - 0.5\pi_t'\Sigma\pi_t - 0.5(\pi_t - g_t\,\pi_{t-1})'\Lambda(\pi_t - \pi_{t-1})] \\
&= \; E[\mu(x_t)'\pi_t - 0.5\pi_t'(\Sigma + \Lambda + E_t[g_{t+1}\Lambda g_{t+1}])\pi_t + \pi_{t-1}' g_t \Lambda \pi_t] \\
&= \; E[\mu_t'(\nu_t + \xi_t) \\
&- 0.5(\nu_t + \xi_t)'(\Sigma + \Lambda + E_t[g_{t+1}\Lambda g_{t+1}])(\nu_t + \xi_t) \\
&+ (\nu_{t-1} + \xi_{t-1})' g_t \Lambda (\nu_t + \xi_t)] \\
&= \; 0.5E[\mu_t'\Sigma^{-1}\mu_t] \; + \; E[\mu_t'(\Sigma^{-1}E_t[\Lambda g_{t+1}(\nu_{t+1} - g_{t+1}\nu_t)] + \Sigma^{-1}\Lambda(g_t\nu_{t-1} - \nu_t))] \\
&- E[\nu_t'\Sigma\xi_t] \; - \; 0.5E[\nu_t'(\Lambda + E_t[g_{t+1}\Lambda g_{t+1}]))\nu_t] \\
&+ E[\nu_{t-1}' g_t \Lambda \nu_{t-1}] \\
&= \; 0.5E[\mu_t'\Sigma^{-1}\mu_t] \; - \; 0.5E[\nu_t'(\Lambda + E_t[g_{t+1}\Lambda g_{t+1}]))\nu_t] \; + E[\nu_t' g_{t+1}\Lambda \nu_{t+1}] \\
&= \; 0.5E[\mu_t'\Sigma^{-1}\mu_t] \; - 0.5E[(\nu_t - g_t\nu_{t-1})'\Lambda(\nu_t - g_t\nu_{t-1})] + O(\varepsilon^2)\,.
\end{aligned}$$

□

C Data and Empirical Results

C.1 Information of Stock Characteristics (Features)

Table C.1 shows the security characteristics that we use as features for all portfolio methods. The features are a subset of the 153 characteristics used in Jensen et al. (2022) plus 1-year

trailing volatility (rvol_252d),[19] where the subset is chosen to have sufficient coverage in the early parts of our sample. Specifically, we select all features with a non-missing value for at least 70% of our sample by the end of 1952. The cluster assignments in Table C.1 are also from Jensen et al. (2022) except rvol_252d, which we assign to the low-risk cluster.

Table C.1: Feature Information

	Characteristic	Cluster		Characteristic	Cluster		Characteristic	Cluster
1	cowc_gr1a	accruals	40	ivol_capm_252d	low_risk	79	opex_at	quality
2	oaccruals_at	accruals	41	ivol_ff3_21d	low_risk	80	qmj_prof	quality
3	oaccruals_ni	accruals	42	rmax1_21d	low_risk	81	qmj_safety	quality
4	taccruals_at	accruals	43	rmax5_21d	low_risk	82	sale_bev	quality
5	taccruals_ni	accruals	44	rvol_21d	low_risk	83	corr_1260d	seasonality
6	fnl_gr1a	debt_issuance	45	turnover_126d	low_risk	84	coskew_21d	seasonality
7	ncol_gr1a	debt_issuance	46	zero_trades_126d	low_risk	85	dbnetis_at	seasonality
8	nfna_gr1a	debt_issuance	47	zero_trades_21d	low_risk	86	kz_index	seasonality
9	noa_at	debt_issuance	48	zero_trades_252d	low_risk	87	lti_gr1a	seasonality
10	aliq_at	investment	49	rvol_252d	low_risk	88	pi_nix	seasonality
11	at_gr1	investment	50	prc_highprc_252d	momentum	89	seas_11_15an	seasonality
12	be_gr1a	investment	51	ret_12_1	momentum	90	seas_11_15na	seasonality
13	capx_gr1	investment	52	ret_3_1	momentum	91	seas_2_5an	seasonality
14	coa_gr1a	investment	53	ret_6_1	momentum	92	seas_6_10an	seasonality
15	col_gr1a	investment	54	ret_9_1	momentum	93	ami_126d	size
16	emp_gr1	investment	55	seas_1_1na	momentum	94	dolvol_126d	size
17	inv_gr1	investment	56	ocf_at_chg1	profit_growth	95	market_equity	size
18	inv_gr1a	investment	57	ret_12_7	profit_growth	96	prc	size
19	lnoa_gr1a	investment	58	sale_emp_gr1	profit_growth	97	iskew_capm_21d	short-term reversal
20	mispricing_mgmt	investment	59	seas_1_1an	profit_growth	98	iskew_ff3_21d	short-term reversal
21	ncoa_gr1a	investment	60	tax_gr1a	profit_growth	99	ret_1_0	short-term reversal
22	nncoa_gr1a	investment	61	dolvol_var_126d	profitability	100	rmax5_rvol_21d	short-term reversal
23	noa_gr1a	investment	62	ebit_bev	profitability	101	rskew_21d	short-term reversal
24	ppeinv_gr1a	investment	63	ebit_sale	profitability	102	at_me	value
25	ret_60_12	investment	64	intrinsic_value	profitability	103	be_me	value
26	sale_gr1	investment	65	ni_be	profitability	104	bev_mev	value
27	seas_2_5na	investment	66	o_score	profitability	105	chcsho_12m	value
28	age	leverage	67	ocf_at	profitability	106	debt_me	value
29	aliq_mat	leverage	68	ope_be	profitability	107	div12m_me	value
30	at_be	leverage	69	ope_bel1	profitability	108	ebitda_mev	value
31	bidaskhl_21d	leverage	70	turnover_var_126d	profitability	109	eq_dur	value
32	cash_at	leverage	71	at_turnover	quality	110	eqnpo_12m	value
33	netdebt_me	leverage	72	cop_at	quality	111	fcf_me	value
34	tangibility	leverage	73	cop_atl1	quality	112	ni_me	value
35	beta_60m	low_risk	74	gp_at	quality	113	ocf_me	value
36	beta_dimson_21d	low_risk	75	gp_atl1	quality	114	sale_me	value
37	betabab_1260d	low_risk	76	mispricing_perf	quality	115	seas_6_10na	value
38	betadown_252d	low_risk	77	op_at	quality			
39	ivol_capm_21d	low_risk	78	op_atl1	quality			

Note: The table shows the security characteristics we use as features for the portfolio methods. The characteristics are from Jensen et al. (2022), and we refer to this paper for details about the construction methodology.

C.2 Portfolio Tuning

Panel A shows the optimal hyper-parameters for the RF method that predicts expected returns. The 1 month model is used by all methods except Portfolio-ML, while the expected return over 2-6 and 7-12 months is only used by Multiperiod-ML and Multiperiod-ML*. Panel

[19] We add 1-year trailing volatility because of its close connection to the covariance matrix. For example, suppose volatility is unrelated to expected returns. In that case, the optimal portfolio should have a higher allocation to low volatility assets, all else equal.

B shows the optimal hyper-parameters for Portfolio-ML and the second layer of portfolio tuning used by Multiperiod-ML* and Static-ML*. Section 4.3 describes how we choose hyper-parameters and table 1 the range of possible hyper-parameters.

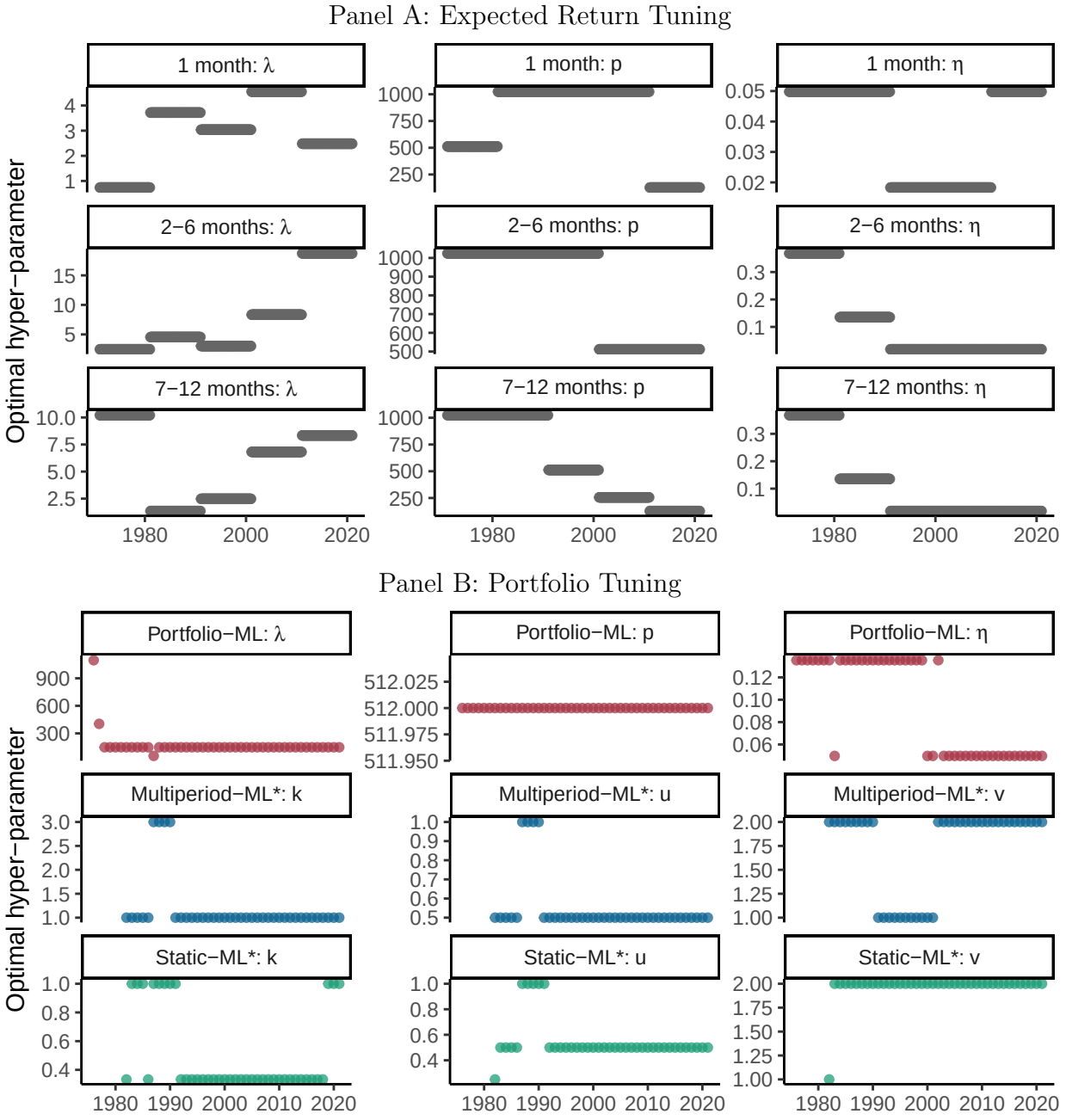

Figure C.1: Optimal Portfolio Hyper-parameters

Note: Panel A shows the optimal hyper-parameters used for predicting expected returns via. Ridge regression of RF transformed features. Panel B shows the optimal hyper-parameters for selecting portfolio weights for Portfolio-ML, Multiperiod-ML*, and Static-ML*. We show the range of possible hyper-parameters in table 1.

C.3 Feature Persistence and Importance across Return Horizons

Figure C.2 shows the monthly autocorrelation of all prediction features. Features grouped into themes following Jensen et al. (2022). We see that most features are highly persistent from month to month but that we also include a substantial amount of fast-moving predictors. These high-frequency predictors are particularly present in the low-risk, seasonality, and short-term reversal themes. Figure C.3 shows a measure of feature importance for each of the three models that predicts future returns. The short-term model that predicts returns one month ahead is used by all portfolio methods except Portfolio-ML, while only Multiperiod-ML uses the two other longer-term models. Notably, feature importance for the short-term model differs from the others by distributing importance more evenly across themes. In contrast, the two longer-term models mainly use value and momentum features.

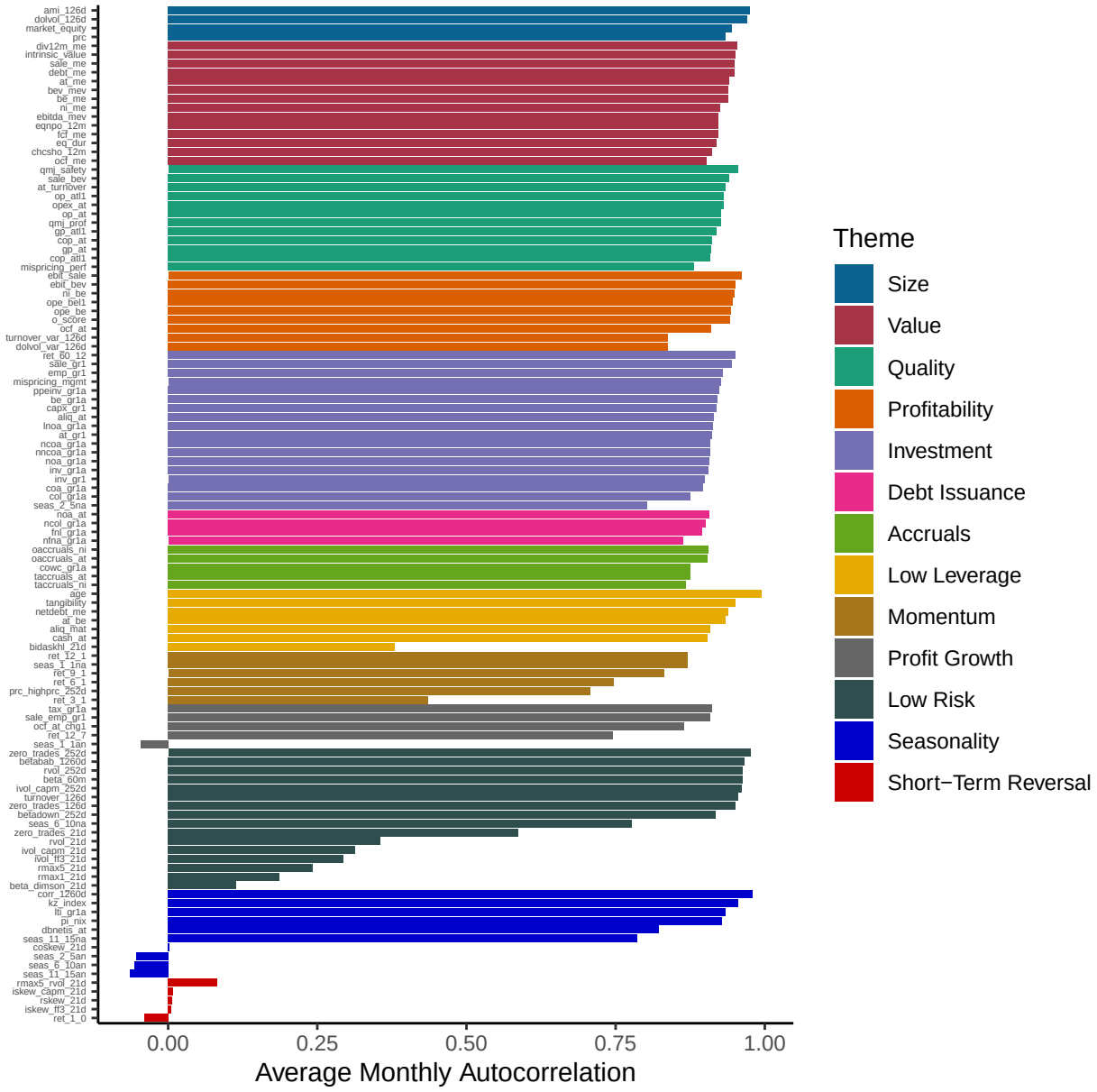

Figure C.2: Feature Autocorrelation

Note: The figure shows the monthly autocorrelation for each feature in our sample. We first compute each feature's monthly autocorrelation for all stocks with at least five years of monthly data. Next, we average the stock-level autocorrelations to arrive at the final estimate. The features are grouped by theme and sorted by average theme autocorrelation.

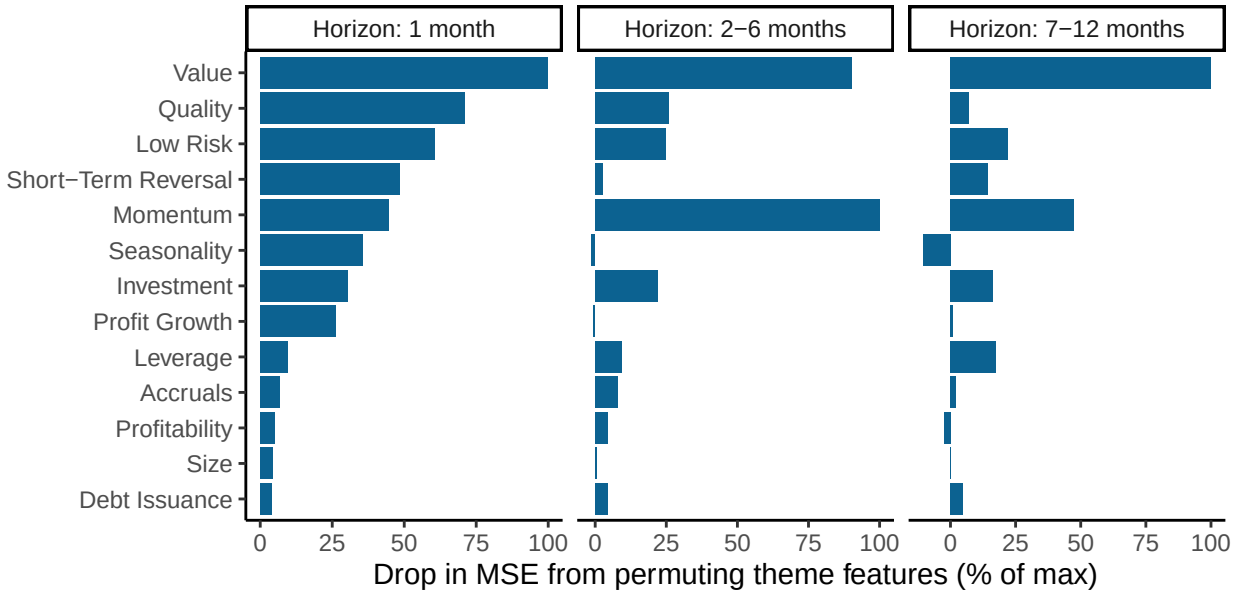

Figure C.3: Feature Importance across Return Horizons

Note: The figure shows feature importance for the three random feature-based models that predict returns in month $t+1$, the average return over month $t+2$ to $t+6$, and the average return over month $t+7$ to $t+12$. For each model, we randomly permute the associated features for each theme while keeping all other features at their actual value. We then implement each method based on this counterfactual data and measure feature importance as the difference in the mean-squared error relative to the implementation that uses the actual data. For comparability, we re-scale the difference by scaling all differences by the largest difference. Hence, feature importance is measured relative to the best feature theme within a specific horizon.