Supervised portfolios

Guillaume Chevalier[*] Guillaume Coqueret[†] Thomas Raffinot[‡]

July 21, 2022

Abstract

We propose an asset allocation strategy that engineers optimal weights *before* feeding them to a supervised learning algorithm. In contrast to the traditional approaches, the machine is able to learn risk measures, preferences and constraints beyond simple expected returns, within a flexible, forward-looking and non-linear framework. Our empirical analysis illustrates that predicting the optimal weights directly instead of the traditional two step approach leads to more stable portfolios with statistically better risk-adjusted performance measures. To foster reproducibility and future comparisons, our code is publicly available on Google Colab.

1 Introduction

Machine learning (ML) algorithms are now ubiquitous in the money management industry and have been extensively studied by scholars in financial economics. One of their common use cases is the prediction of expected returns, e.g., as in Gu et al. (2020) and Coqueret and Guida (2020). Endowed with ML-driven forecasts, the portfolio manager can craft allocations, given a set of preferences and trading constraints.

This splitting of tasks can be summarized as follows: the machine mines the data and finds (possibly spurious) relationships between predictors and returns, while the manager translates the automated predictions in portfolio weights (see for example Cenesizoglu and Timmermann (2012) and Gârleanu and Pedersen (2013) on this matter). This last stage is complex, because it relies on preferences or utility (e.g., the risk-aversion parameter in mean-variance portfolio), as well as on sets of constraints that are idiosyncratic to the manager (leverage, factor exposure, geography, sector biases, liquidity, etc.).

In this two step process (prediction first, followed by optimization), the ML arsenal is traditionally only deployed for the first task.[1] A natural question emerges: why not shift *all* the burden to the machine? As we show in this article, this is not entirely possible

[*]AXA IM, Tour Majunga La Défense, 6 Place de la Pyramide, 92800 Puteaux, FRANCE. E-mail: guillaume.chevalier@axa-im.com

[†]EMLYON Business School, 23 avenue Guy de Collongue, 69130 Ecully, FRANCE. E-mail: coqueret@em-lyon.com

[‡]AXA IM, Tour Majunga La Défense, 6 Place de la Pyramide, 92800 Puteaux, FRANCE. E-mail: thomas.raffinot@axa-im.com

[1]For the sake of completeness, we mention that López de Prado (2016) and Raffinot (2017) have introduced a new way to allocate capital based on unsupervised machine learning, namely *hierarchical clustering algorithms*.

because portfolio weights have to be computed at some point in the process (see Figure 1). We simply advocate to do so earlier than is usually done.

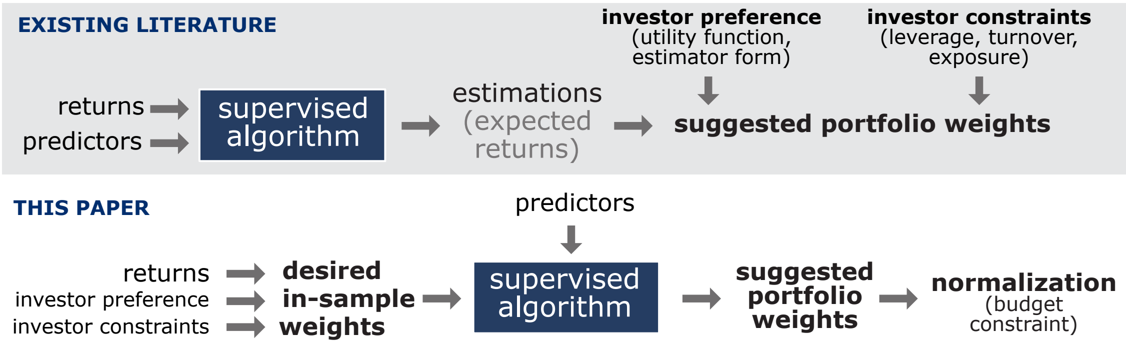

Figure 1: **Comparative diagram**. In the upper panel, this scheme shows the traditional process in the crafting of portfolio weights relying on machine learning methods. In the lower panel, it depicts the simple idea of the paper.

Given a training dataset of past observations, the method we propose starts by computing optimal weights for all relevant dates in the sample. For instance, this is done by considering the realized returns as the *expected* returns in standard portfolio choice optimizations. This stage may require other estimations, typically for covariance matrices, and they can be performed with some other subsets of the training sample. In addition, constraints can of course be added in order to satisfy targets and policies. In the case of mean-variance-based allocation, the risk aversion parameter is naturally a key input and we document its importance in our results.

The construction of optimal weights in the training sample can thus be interpreted as a pre-processing stage. Instead of trying to forecast the returns directly, we transform them into the in-sample weights that would have maximized a given utility function *if these returns had been known in advance*. It it then these optimal weights which we try to forecast with the chosen set of predictors. It is important to note that this pre-processing stage allows the construction to be completely forward-looking, which is a major advantage when market regimes shift. Indeed, by lagging the data, we can use the in-sample future realized returns to compute all estimates (mean vector and covariance matrix). This allows to be forward-looking in the training sample, while at the same time avoiding any look-ahead bias.

Our method is flexible and works for any tractable utility function and set of constraints (e.g., leverage preferences towards long-only or long-short portfolios). It can accommodate any asset class, as long as there are suitable predictors for its returns. As we show, it also works *across* asset classes.

The performance of our approach is evaluated across three datasets, which differ in terms of number of assets and composition of the universe. This highlights the robustness of the methodology we propose.

We start by computing the target weights based on a constrained Markowitz mean-variance optimization. Next, we build the predictors, which fall into two categories. First, those that pertain to momentum-style variables (past returns) and those that are risk-based (realized volatilities). Momentum predictors have been shown to perform well out-of-sample in the literature (Gu et al. (2020) and Coqueret (2021)). Risk measures are also important because they will be useful to learn the risk tolerance or aversion that are coded within the weights (e.g., via the risk aversion parameter). In fact, both momentum and volatility are the main drivers of returns, according to the recent work of He et al. (2021).

The second category of predictors relates to macro-economic series, such as the yield curve, the TED spread, the VIX and the US credit spreads. These variables are important because they allow to introduce some conditionality in the model so as to adapt to changing environments.

The findings of the paper can be summarized as follows. In a traditional mean-variance optimization with constraints, predicting optimal weights directly instead of expected returns generates substantial gains with respect to trading costs as long as the investor is not too risk averse. In addition, this generates higher average returns as well as slightly higher realized risk. Nevertheless, the net aggregate effect is that Sharpe ratios are improved when switching to the direct estimation of weights. Our results are robust to sub-period analyses and still hold when replacing boosted trees by simple regressions.

The remainder of the paper is structured as follows. Section 2 surveys the literature that is adjacent to the themes of the article. We present the theoretical framing of our approach in Section 3, along with a simple illustration. The empirical protocol is outlined in detail in Section 4 and the corresponding results are gathered in Section 5. Several robustness checks are carried out in Section 6. Finally, Section 7 concludes.

2 Related literature

The idea of optimizing portfolio weights based on predictors directly is not new. The parametric portfolio policies of Brandt et al. (2009), Hjalmarsson and Manchev (2012) and Ammann et al. (2016) are such examples and they seek to build enhanced benchmarks based on firms' characteristics.

Moreover, our contribution relates mostly to the streams of the literature that propose non-standard portfolios allocations. Recently, Zhang et al. (2020a) seek to optimize Sharpe ratios via gradient ascent in a neural network architecture where portfolio weights are derived from a given set of predictors. Langlois (2020) models efficient weights as functions of auto-regressive latent variables. The system learns the dynamics from past returns and is then able to make predictions about expected returns.

Optimal weights are also a central ingredient in reinforcement learning because they are progressively learned via the actions (investment decisions) of the agent and the subsequent rewards (see Moody et al. (1998) and Zhang et al. (2020b), among others). However, due to dimension issues, applications often include a very limited number of assets and position types. In André and Coqueret (2020), the authors bypass this technical hurdle by considering Dirichlet distributions as the driver of the allocation process.

The contributions that are the closest to our paper are the recent studies of Zhang et al. (2020a), Uysal et al. (2021), Butler and Kwon (2021) and Simon et al. (2022).[2] In the former, it is proposed to maximize the Sharpe ratio via gradient ascent in a neural network structure where the portfolio weights serve as inputs. Their approach is generalized to

[2]Outside financial applications, Elmachtoub and Grigas (2022) provide theoretical results on various "*predict, then optimize*" tasks.

other performance metrics in Uysal et al. (2021), with a view towards risk parity. Other black-box approaches are advocated in Cong et al. (2022), Costa and Iyengar (2022) and Simon et al. (2022).

In Butler and Kwon (2021), the authors also seek to optimize portfolio weights directly. They show, theoretically, that if predictions stem from a linear regression, it is possible to model the entire process as a neural network with two intermediate layers: one for the prediction stage and one for the optimization. Nevertheless, if nonlinear models (e.g., tree methods, as in the present paper) are used for the prediction stage, then such architectures cannot be implemented. Using simpler forecasting techniques (AR(FI)MA), Golosnoy and Gribisch (2022) seek predict the realized weights of global minimum variance portfolios. The present paper aims to propose a general framework for the direct estimation of portfolio weights, and to pursue the discussion on the out-of-sample efficiency of this approach.

3 Model

This section first lays out a rigorous formulation of the portfolio problem tackled in this paper. In order to illustrate our idea, we then propose a simple toy model.

3.1 Theoretical framework

At any given (discrete) time t, the agent receives a financial dataset $\mathbb{D}_t$ which comprises information on N assets (stocks, stock portfolios, or broad asset class indices in this paper), as well as on the economy more generally. More precisely, it must contain at least the first two of the following elements:

- **prices** of assets, and possibly **dividends**, which can be used to derive price returns, total returns and dividend yields;
- from **returns**, risk measures like realized volatility, Value-at-Risk, CVaR (i.e., Expected Shortfall) can be assessed;
- traditional **accounting values** or ratios which are often mentioned in the asset pricing literature: market capitalization (Banz (1981), Fama and French (1992); Fama and Kenneth (1993), Van Dijk (2011)), book-to-maket ratios (Fama and French (1992); Fama and Kenneth (1993), Asness et al. (2013)), asset growth and profitability margins (Fama and French (2015)), etc.;[3]
- **volumes**, **bid-ask spreads** and other **liquidity proxies** that can help build precise models of transaction costs;
- so-called **alternative data** fields, such as sentiment (both asset-specific, or region and industry aggregates) and ESG data;
- **macroeconomic** variables (GDP growth, inflation, credit spread, term spread, etc.).

With these variables, the agents computes a set of estimations $\mathcal{E}_t$. Standard examples of estimates include the first moments of expected returns. In portfolio choice problems, the most important one is the mean vector of expected returns. If $\boldsymbol{r}_t$ denotes the $N \times 1$ vector

[3]Previous attributes are also mentioned in asset pricing, like past returns (Jegadeesh and Titman (1993), Asness et al. (2013), or realized volatility (Baker et al. (2011), Li et al. (2016))

of all assets' time-t returns, then $\hat{\boldsymbol{\mu}}_t = \mathbb{E}_t[\boldsymbol{r}_{t+1}] = \mathbb{E}[\boldsymbol{r}_{t+1}|\mathcal{F}_t]$ gives the agent's estimate of the one step ahead average returns, conditional on the information set available at time t, $\mathcal{F}_t$. In our framework, $\mathcal{F}_t = \cup_{s\le t}\mathbb{D}_s$. For instance, $\hat{\boldsymbol{\mu}}_t$ can be estimated using supervised learning algorithms, as is done in Gu et al. (2020):

$$\boldsymbol{r}_{s+1} = f_b(\mathbb{D}_s) + \boldsymbol{e}_{s+1,b}, \quad s < t, \tag{1}$$

with the final estimate being $\hat{\boldsymbol{\mu}}_t = \hat{f}_b(\mathbb{D}_t)$ where $\hat{f}_b$ is the estimated model and t the current date. The b subscript stands for the *benchmark* learning approach.

Likewise, the agent may construct an estimator $\hat{\boldsymbol{\Sigma}}_t = \mathbb{E}_t[(\boldsymbol{r}_{t+1} - \hat{\boldsymbol{\mu}})(\boldsymbol{r}_{t+1} - \hat{\boldsymbol{\mu}})']$ for the covariance matrix of returns. We use the convention that $\boldsymbol{v}'$ denotes the transpose of the vector $\boldsymbol{v}$. In addition, $\mathbf{1}_N$ and $\boldsymbol{I}_N$ will denote the unit N-dimensional vector and the corresponding identity matrix. Thus, the mean-variance investing framework à la Markowitz (1952) can be interpreted as a case where $\mathcal{E}_t = \{\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\Sigma}}_t\}$. Moments of higher order are sometimes used (see, e.g., Harvey et al. (2010)), but they usually require algebra with 3 or 4 dimensional arrays (tensors) which are harder to handle.

The investor seeks to maximize some expected utility over future returns[4] and under a given set of constraints on the weights of her portfolio $\boldsymbol{w}$, which we assume can be written as $\mathcal{C}(\boldsymbol{w}) \succeq \mathbf{0}$, where $\mathbf{0}$ is a zero vector of arbitrary size. This latter curvy inequality is understood as term-by-term. For instance, a long-only portfolio is equivalent to (or requires) $\boldsymbol{w} \succeq \mathbf{0}_N$, where N is the number of assets. For some matrix $\boldsymbol{A}$, linear constraints such as $\boldsymbol{A}\boldsymbol{w} \succeq \mathbf{0}_N$ are well suited to impose geographic, factor, or sector constraints.

3.2 The benchmark approach

We posit that this optimization can be formulated and solved as a function of the estimates $\mathcal{E}_t$, i.e., that there exists a function g_b such that

$$\boldsymbol{w}_b^* = \underset{\boldsymbol{w}}{\text{argmax}} \ \left\{\mathbb{E}_t\left[u(\boldsymbol{w}'\boldsymbol{r}_{t+1})\right], \text{ s.t. } \mathcal{C}(\boldsymbol{w}) \succeq \mathbf{0}\right\} = g_b(\mathcal{E}_t), \tag{2}$$

where the benchmark function g_b naturally operates on the set of estimators used by the agent. Again, the subscript b stands for benchmark. For instance, in the case of a mean variance investor with a typical budget constraint $\boldsymbol{w}'\mathbf{1}_N = 1$ and a diversification target $\boldsymbol{w}'\boldsymbol{w} \le \kappa$,[5] the maximization of $\boldsymbol{w}'\hat{\boldsymbol{\mu}}_t - \frac{\gamma}{2}\boldsymbol{w}'\hat{\boldsymbol{\Sigma}}_t\boldsymbol{w}$ yields a solution of the form

$$\boldsymbol{w}_b^* = \gamma^{-1}(\hat{\boldsymbol{\Sigma}}_t + \delta\boldsymbol{I}_N)^{-1}(\hat{\boldsymbol{\mu}}_t + \eta\mathbf{1}_N), \tag{3}$$

where η is chosen to satisfy the budget constraint and δ is adjusted to match the diversification goal. The latter is very useful to reduce the leverage of unconstrained mean-variance portfolios, which is one of their notorious weaknesses.

[4] In economics, the agent often maximize expected utility over terminal *wealth*. In finance, it is customary to work on *returns* instead, see Campbell and Viceira (2002)

[5] We use the L^2 norm of weights to measure diversification, as in Goetzmann and Kumar (2008). The integration of diversification constraints in standard portfolio optimization is discussed in Coqueret (2015). In practice, the minimum feasible κ is $1/N$, where N is the number of assets, and it is reached for the equally-weighted portfolio.

3.3 A direct estimation of weights

The second, less straightforward, way to proceed is the following. We index the dates of the data contained in $\mathbb{D}_t$ with s. Thus, with the knowledge of the realizations of returns, it is possible, for each date s, to compute the in-sample optimal weights:

$$\mathbf{w}_s^* = \underset{\mathbf{w}_s}{\text{argmax}} \ \{u(\mathbf{w}_s'\boldsymbol{r}_s), \text{ s.t. } \mathcal{C}(\mathbf{w}_s) \succeq \mathbf{0}\}, ^{6} \quad s < t. \tag{4}$$

Note that there is no time shift between the weights and the returns because the former are based on the synchronous realizations of the latter. For instance, if the utility function is $u(x) = x - \gamma x^2/2$ and the constraints are $\mathbf{w}_s'\mathbf{1}_N = 1$ (budget constraint) and $\mathbf{w}_s'\mathbf{w}_s \leq \kappa$ (diversification), then

$$\mathbf{w}_s^* = (\boldsymbol{r}_s\boldsymbol{r}_s' + \delta \boldsymbol{I}_N)^{-1}(\boldsymbol{r}_s + \eta \mathbf{1}_N), \tag{5}$$

which is the equivalent of (3) when there is no estimation step because only one date is available. The shrinkage term $+\delta\boldsymbol{I}_N$ serves as regularization and ensures that the inverse matrix is well-defined. From the in-sample optimal weights $\mathbf{w}_s^*$ that are obtained with realized *past* returns only, the second approach aims to learn portfolio compositions in a supervised fashion:

$$\mathbf{w}_{s+1} = f_d(\mathbb{D}_s) + \boldsymbol{e}_{s+1,d}, \quad s < t. \tag{6}$$

In the above equation, the time shift underlines that the weights are learned from prior data $\mathbb{D}_s$. Plainly, Equation (6) is the twin version of Equation (1). The latter (benchmark) predicts returns, while the former seeks to forecast optimal weights. The subscript d stands for *direct* estimation of portfolio weights. The final time-t weights are then simply

$$\boldsymbol{w}_d^* = \hat{f}_d(\mathbb{D}_t),$$

where $\hat{f}_d$ is the estimated model. By construction, they may not sum to one and satisfy the budget constraint, thus it may be required to normalize them a posteriori.

One notable advantage of the direct approach is that if the sample encompasses several dates, the target portfolio weights for each individual asset can be smoothed so as to reduce trading costs. Often, predictors are slowly moving (macro-economic indicators, accounting values and ratios, etc.), so that persistent dependent variables are preferable in panel models (see Coqueret (2021)). The rationale is that if predictors are stable in time, they can only forecast variables that also have smooth time-series.

3.4 A toy example

To illustrate this abstract framework, let us consider the case of regularized mean-variance portfolios. This example does not even require any supervision and corresponds to a utility function of

$$u(\boldsymbol{w}) = \boldsymbol{w}'\boldsymbol{\mu} - \boldsymbol{w}'\boldsymbol{\Sigma}\boldsymbol{w} - \delta\boldsymbol{w}'\boldsymbol{w}, \tag{7}$$

which the investor seeks to maximize under the budget constraint $\boldsymbol{w}'\mathbf{1} = 1$. The penalization $\delta\boldsymbol{w}'\boldsymbol{w}$ improves diversification when $\delta > 0$ and regularizes the optimal solution à

[6] More generally, the utility could be written as $u(\boldsymbol{w}_s, \mathcal{E}_s)$, but we stick to simpler notations to ease readability.

la Tikhonov. Given series of $(N \times 1)$ returns $(\boldsymbol{r}_t)_{1\le t\le T}$, the direct estimation of weights is such that they are proportional to

$$\boldsymbol{w}_b^* \propto \left(T^{-1}\sum_{t=1}^{T}(\boldsymbol{r}_t-\bar{\boldsymbol{r}})(\boldsymbol{r}_t-\bar{\boldsymbol{r}})'+\delta \boldsymbol{I}_N\right)^{-1}\times\sum_{t=1}^{T}\boldsymbol{r}_t, \tag{8}$$

where $\bar{\boldsymbol{r}}$ is the sample mean of $\boldsymbol{r}_t$, and the constant δ tunes the penalization: as it increases, the portfolio leverage decreases. In the limit $\delta \to \infty$, we recover the naive $1/N$ portfolio if all average returns are equal ($\bar{\boldsymbol{r}} \propto \mathbf{1}$). $\boldsymbol{w}_b^*$ will serve as the benchmark portfolio. On the other hand, the simplest learning technique is to take the average of local (in-sample pointwise) well-defined weights, so that we have

$$\begin{aligned}
\boldsymbol{w}_d^* &= T^{-1}\sum_{t=1}^{T}\underbrace{c_t\left(\boldsymbol{r}_t\boldsymbol{r}_t'+\delta\boldsymbol{I}_N\right)^{-1}\boldsymbol{r}_t}_{\text{pointwise weights}} && (9)\\
&= (\delta T)^{-1}\sum_{t=1}^{T}c_t\left(\boldsymbol{I}_N-\frac{\delta^{-1}\boldsymbol{r}_t\boldsymbol{r}_t'}{1+\delta^{-1}\boldsymbol{r}_t'\boldsymbol{r}_t}\right)\boldsymbol{r}_t\\
&= \left(\delta^{-1}\sum_{t=1}^{T}c_t\right)\mathbf{1}_N-(\delta^2T)^{-1}\sum_{t=1}^{T}c_t\frac{\boldsymbol{r}_t'\boldsymbol{r}_t}{1+\delta^{-1}\boldsymbol{r}_t'\boldsymbol{r}_t}\boldsymbol{r}_t && (10)
\end{aligned}$$

where we have used the Sherman-Morrison identity in the second equality. In each line, c_t is the constant that scales the weights at time t. Clearly, in this case again, we see the effect of δ: as it increases, the first term goes to zero at the speed of δ^{-1}, whereas the second term decreases with δ^{-2}. Thus, only the first term survives asymptotically after normalization of the weights - when δ is arbitrarily large.

There is no simple analytical expression for these weights. In the first case, the distribution of mean-variance optimal weights along the efficient frontier is discussed in Kan and Smith (2008). Adding the regularization further complicates the expressions. In the second case, the expression (9) shows products of random vectors. Even if we assume that they are Gaussian, the distributions underpinning these products are complex if we assume nonzero correlations (see Nadarajah and Pogány (2016)). Adding the diagonal term, taking the inverse, and then summing leaves no hope for tractable formulae.

We must thus resort to simulations and implement the following protocol, summarized in Table 1. Given a covariance matrix $\boldsymbol{\Sigma}$ (detailed below), we generate bivariate returns. The first method computes an estimator for $\boldsymbol{\Sigma}$ so as to derive the traditional penalized minimum variance weights. The second method averages the point-wise weights as written in Equation (10). The efficiency of both types of weights is evaluated via the realized utility.

We consider 2 assets with Gaussian returns. The assets have mean returns of 5% and 10%, and volatilities of 10% and 20%, respectively. We test two configurations for their correlations: -0.3 and +0.6, which could correspond to a bond-stock and a stock-stock situation. We also allow for another degree of freedom: the samples use to estimate the weights consist of 20, 60, or 300 simulated points. In terms of daily points, this corresponds roughly to one, three and fifteen months of data.

The resulting average utilities are plotted in Figure 2 as a function of δ. The orange curve, which corresponds to the traditional approach is decreasing: as the penalization

Step	Benchmark	Direct
Given $\mathbf{\Sigma}$, for each simulation $n = 1, \dots, N$: do		
0	Simulate n samples $N(\mathbf{0}, \mathbf{\Sigma})$	
1	Compute weights $\boldsymbol{w}_b$ from (8)	Compute weights $\boldsymbol{w}_d$ from (10)
2	Compute utility $u_{n,b} = u(\boldsymbol{w}_b)$ from (7)	Compute volatility $u_{n,d} = u(\boldsymbol{w}_d)$ from (7)
return average utlities $N^{-1}\sum_{n=1}^{N} u_{n,b}$ and $N^{-1}\sum_{n=1}^{N} u_{n,d}$		

Table 1: Pseudo-code of simulation exercise.

increases, the utility decreases. The weights are more regularized, which may, in an out-of-sample backtest be beneficial (see Jagannathan and Ma (2003)). The leftmost points on the blue curve are clearly suboptimal. This is because when δ is small, the inversions linked to the in-sample weights are ill-defined (the original matrices are not well conditioned). This can be seen in expression (10), wherein the terms, in all generality, increase in magnitude when δ decreases to zero.

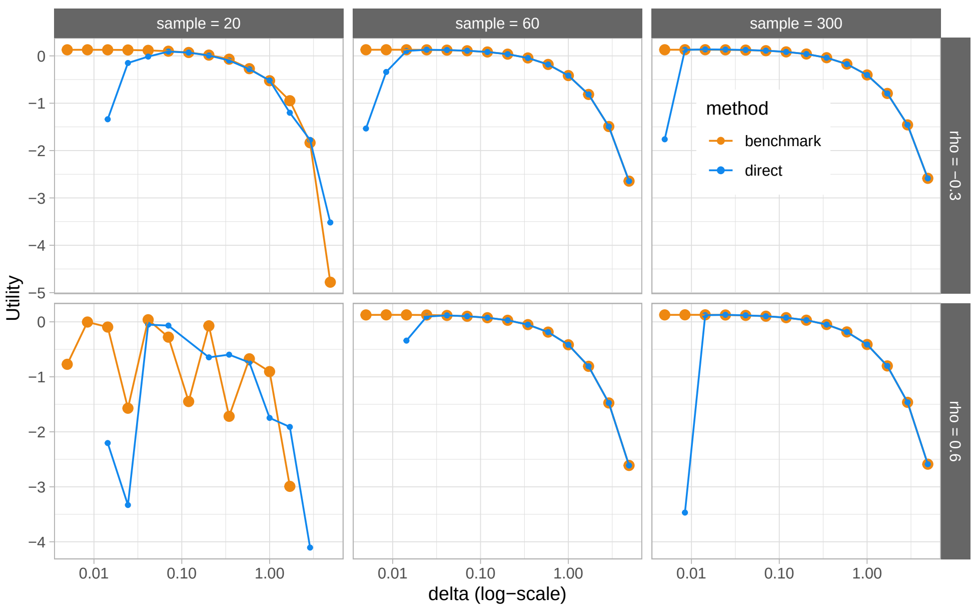

Figure 2: **Regularized mean-variance choice**. We plot the realized utilities defined in Table 1 as a function of δ (x-axis), sample size (left, center and right panels), and asset correlation (negative for the top graphs, positive for the bottom ones). We run 10^5 simulations in total. Portfolio types (benchmark versus direct estimation) are coded via colors. Values below -5 are omitted to ease readability.

However, when δ increases, both methods yield indistinguishable results, meaning that the simple averaging for pointwise weights *works* as well as the optimal one. However, this is only true when samples are deep enough, as we see in the left panels that averaging over an insufficient number of weights generates noise. In fact, in the lower left panel, it is plain that even the traditional approach fails because the estimators are not precise enough.

We underline that because we are in an i.i.d. setting, the orange curve is optimal, so that the direct approach can at best match it, which it often does. In practice, as is well-known, returns and risk-premia are risk varying (see, e.g., Chaieb et al. (2021)), so that the benchmark approach is often not the best out-of-sample. This has been documented at least since the work of Jorion (1985).

Moreover, in this toy model, there is no proper supervision stage, as defined in Equation (6): the learning phase consists of a simple averaging. When supervision and learning stages occur in the processing of weights, the analysis gains in complexity and analytical results are out of reach. We thus detail several applications in the remainder of the paper.

4 Empirical protocol

This section is dedicated to the detailed presentation of our backtests. The latter can be reproduced via our public code on Google Colab.

4.1 Datasets and variables

Portfolio Instruments

The performance of our method is evaluated on three different investment universes:

- the 10 US industry portfolios from Kenneth French's website.[7]. Consequently, the assets are sector portfolios: Consumer-Discretionary, Consumer-Staples, Manufacturing, Energy, Technology, Telecommunication, Wholesale and Retail, Health, Utilities, and Other.
- the 25 portfolios from the same source, sorted on size and book-to-market ratios.
- four asset class indices, namely: two equity indices (MSCI World index and MSCI emerging index), and two bond indices (Bank of America Merrill Lynch Global Corporate Index and Bank of America Merrill Lynch Global Government Bond Index).

All asset returns and variables are sampled at the daily frequency, from January 1997 to September 2020.

The rationale for choosing these investment universes is twofold. First, the data for the first two is public and makes our results handily reproducible. To this purpose, a Jupyter Notebook is openly available to allow researchers to replicate our results.[8] Second, the small number of assets alleviates issues in the estimation of covariance matrices. It is possible to evaluate (and invert) the sample estimator even with just one month of daily data.

Explanatory variables

We only resort to two types of predictors in all of our models, which are available for any asset class with public quotes. The first type is simply derived from the prices. The second type of predictors is of macro-economic nature.

Prices are translated into returns at various horizons so as to assess momentum and reversals, and into realized volatilities, which approximate asset-specific risk. This choice

[7]Available at http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html
[8]It is available here.

is dictated by two considerations. First, as is reported in Gu et al. (2020) and He et al. (2021), past returns (momentum) and volatilities are important drivers of the cross-section of returns.[9] Recently, Rapach et al. (2019) document the importance of past returns for industry portfolios, which are thus relevant in the context of the present empirical study. The second reason that incites us to stick with past returns and volatilities is that these indicators can be derived for all asset classes. Thus, it makes it easier to transpose our result for other markets.

As regards momentum, we compute the past annualized growth rate of each instrument sectors over four horizons: 21 days ($\sim$ one month), 63 days ($\sim$ 3 months), 126 days ($\sim$ 6 months) and 252 days ($\sim$ 12 months). For example, the time-t 126-day annualized momentum is computed as $(\frac{P_t}{P_{t-126}} - 1) * \frac{252}{126}$, where P_t is the value of the original time-series.

With repsect to realized volatility, we straightforwardly resort to the standard deviation of daily realized returns of each instrument over the same four horizons, which span 1, 3, 6, and 12 months.

The second type of predictors relates to macro-economic indicators. We consider four classical economic variables: the yield curve, the credit spread, the Ted spread and the VIX index[10]. All the macroeconomic series were download from the FRED (https://fred.stlouisfed.org).

The yield curve is calculated as the spread between 10-Year Treasury Constant Maturity and 3-Month Treasury Constant Maturity. The credit spread is calculated as the daily time-series of US High Yield Spread Index minus the values of the US AA Corporate Spread Index. The Ted spread is the difference between the three-month Treasury bill and the three-month LIBOR based in U.S. dollars. At last, the VIX, which is often referred to as the fear index or fear gauge, is a measure of the stock market's expectation of volatility based on S&P 500 index options.

4.2 The two-step approaches

4.2.1 Portfolio construction

All portfolio policies are based on constrained Markowitz mean-variance optimization. Let n denote the number of instruments in the portfolio and $\boldsymbol{w}_t \in \mathbb{R}^n$ the portfolio allocation vector. Given $\hat{\boldsymbol{\mu}}_t \in \mathbb{R}^n$ the vector of expected returns (annualized) over a given horizon and $\hat{\boldsymbol{\Sigma}}_t \in \mathbb{R}^{n,n}$ the corresponding covariance matrix (annualized) at time t, the solution $\boldsymbol{w}^*$ can be written:

$$\boldsymbol{w}^*(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\Sigma}}_t) = \arg\max_{\boldsymbol{w_t}} \left\{ \hat{\boldsymbol{\mu}}_t^T \boldsymbol{w}_t - \frac{\gamma}{2} w_t^T \hat{\boldsymbol{\Sigma}}_t \boldsymbol{w}_t \ , s.t. \ C(\boldsymbol{w}_t) \right\}, \tag{11}$$

with γ the risk aversion coefficient and $C(\boldsymbol{w}_t)$ constraints on weights. In our empirical study, we impose *box* constraints, meaning that weights must lie in specific intervals, e.g., $[0, 0.2]$, which corresponds to long-only portfolios in which an asset can account at most

[9] In a similar vein, Han et al. (2013), Han et al. (2016) and Han et al. (2021) show that technical indicators, based on past prices are also relevant predictors and are priced in the cross-section of stocks.

[10] See Welch and Goyal (2008) and Hull and Qiao (2017) for more information on the ability of macroeconomic predictors to forecast returns.

for 20% of the portfolios. Again, we recall that this optimization is carried out at different stages, depending on the method:

- in the **benchmark** traditional approach: it is applied on ML-based predictions.
- in the **direct** (supervised weights) approach: it is applied in-sample to obtain the past optimal weights, which are then used as labels by the ML engine.

In the subsection below, we detail how we obtain the estimates $\hat{\boldsymbol{\mu}}_t$ and $\hat{\boldsymbol{\Sigma}}_t$.

4.2.2 Prediction model

As described in the asset pricing literature (Gu et al. (2020)), we forecast the time-t ($N \times 1$) vector of dependent variables with a standard prediction error model:

$$\boldsymbol{y}_{s+1} = g_t(\boldsymbol{X}_s, \boldsymbol{M}_s) + \boldsymbol{\epsilon}_{s+1},$$

where $\boldsymbol{X}_s$ represents the matrix of instruments' specific explicative features (momentum and volatilities) and $\boldsymbol{M}_s$ the macro-economic variables plus a constant vector of ones. The s index stands for all the dates in the training sample used at time t.

The function g_t is approximated with a machine learning model and its parameters are updated using a pooled regression on an expanding window training scheme (see sections below). Hence, the number of observations is much higher than the number of explicative features and the risk of over-fitting is less of a concern, as long as g_t does not include too many parameters.

In line with Gu et al. (2020), we define the the baseline set of covariates $\boldsymbol{Z}_s$ as $\boldsymbol{Z}_s = \boldsymbol{X}_s \otimes \boldsymbol{M}_s$, where $\otimes$ is the Kronecker product. The total number of features (the number of columns of $\boldsymbol{Z}_{\boldsymbol{s}}$) is thus 8 * (4 + 1) = 40 because there are eight momentum and volatility predictors and four macro-economics variables (plus a fifth constant term). The model is thus:

$$\boldsymbol{y}_{t+1} = g_t(\boldsymbol{Z}_t) + \boldsymbol{\epsilon}_{t+1}, \tag{12}$$

and corresponds to Equation (3) for the traditional method and to Equation (6) for the direct approach. Naturally, each technique will lead to a dedicated (trained) model g_t. Importantly, there is no guarantee that the predicted weights from the direct approaches will sum to one, though they should be close to satisfying the budget constraint. Consequently, after each batch of predictions, we normalize the weights accordingly.

As regards the target (label), we consider three variables:

- $y_{t+1} = r_{t+1}$: Future returns; this is the **benchmark** (standard) approach (Equation (3)).
- $y_{t+1} = w_{t+1}$: Weights coming from the optimizer when knowing both expected returns and future covariance; this is a first **direct** approach to predict the portfolio policies (see Equation (6)).
- $y_{t+1} = w_{t+1} - \frac{1}{N}$: Difference between w_{t+1} and the equally-weighted portfolio. The idea here is to make the distribution of the target vector more time-homogeneous (its mean does not change in time, which can be useful if the number of assets evolves). This is an alternative **direct** approach to forecast the portfolio allocations (again, this corresponds to Equation (6)).

Finally, the expected return $\hat{\boldsymbol{\mu}}_t = g_t(\boldsymbol{Z}_t)$, that is, the prediction based on the current predictor values. With regard to the covariance matrix, we simply take the sample estimator over the realized returns posterior to s. For simplicity, the sample size is equal to the horizon we seek to predict. For example, if we predict three-month returns, we use three months of data.

4.2.3 Gradient boosting decision trees

We now turn to the ML engine that we use in our empirical work (g_t in Equation (12)), namely boosted trees. *Boosting* is based on the idea of creating an accurate learner by combining many so-called "weak learners" (Schapire (1990)), i.e., with high bias and small variance, in order to form a single "strong" predictive model. With boosting, each trees are fitted sequentially. In other words, each tree is fitted using information from the previously fitted tree. They have been shown to perform well for predictive tasks in finance (Krauss et al. (2017), Guida et al. (2018) and Gu et al. (2020)). Recently, Januschowski et al. (2021) and Makridakis et al. (2022) have shown that in the latest (M5) forecasting competition, tree methods remain highly popular and often outperform neural networks, especially on tabular datasets (Shwartz-Ziv and Armon (2022), Grinsztajn et al. (2022)).

This approach is fast and efficient, but with a higher risk of over-fitting. Therefore, it is crucial to correctly perform a thorough cross-validation of the hyper-parameters (HPs), see next section below. The set of important HPs for boosted trees are well documented and we will optimize the following:

- the **number of trees**. More trees imply a more complex model, and a higher risk of overfitting.
- the **learning rate**. A lower rate leaves more room for future trees (the model learns more slowly) and reduces the risk of overfitting.
- the **maximum depth of trees**. The deeper the trees, the more sophisticated the model.
- the **minimum sum of instance weight needed in a child**. If a leaf node has the sum of instance weight that is too small, the partitioning of the tree stops. This prevents the tree from growing to much and helps mitigate the risk of overfitting.

Before training the models, we apply the following normalization techniques:

- *Quantile normal transformation for the macroeconomic variables*: we standardize the time-series into quantile and then map the values to a normal distribution. This is of course done on the training sample at each step and does not imply any forward looking leakage of any kind.
- *Cross sectional normalization for predictors*: we scale the cross sectional values (i.e. among the 10 industrial sectors) between 0 and 1 using the empirical cumulative distribution function. This standard is commonplace in the asset pricing literature (Koijen and Yogo (2019), Kelly et al. (2019), Freyberger et al. (2020) and Gu et al. (2020)).
- *Tanh scaling for the dependent variable*: we mutate the time-series of dependent variables using the hyperbolic tangent (*tanh*) function. The rationale for this is to

center the label variables and make them more comparable by taming outliers. The transform is the following:

$$y_{norm} = 0.5 * (\tanh(0.01 * ((y - y_{mean})/y_{std}))), \quad (13)$$

where the y_{mean} and y_{std} are straightforwardly the sample mean and standard deviation of the labels. The reverse transformation is performed after the prediction to transform back the labels into its original values (for prediction purposes), using the below equation:

$$y = \text{arctanh}(y_{norm}/0.5) * (y_{std}/0.01) + y_{mean}.$$

4.2.4 Random search cross-validation

The simple K-fold cross-validation works by dividing the data into K contiguous folds. For the k^{th} part, the learning method is fit to the other $K-1$ parts of the data and calculate the prediction error of the fitted model when predicting the k^{th} part of the data. This is done for $k = 1, 2, ..., K$ and the K prediction error estimates are averaged. For this study, $K = 5$ was chosen, which is a commonly used value, and the five splits are performed chronologically, generating 5 non-overlapping ordered blocks for testing.

Working with time-series adds an important risk of data **leakage**, whereby information contained in the testing set is partly present in the training set, which of course never happens in practice. In order to avoid such risk, and as is advised in Section 7.4 in López de Prado (2018), each training dataset is purged. More specifically, at each rebalancing time t, for each K-fold (20% of the sample for testing and 80% for training), all observations from the training set whose labels overlap in time with those labels included in the testing set are removed. Thus, training sets are in fact slightly smaller than 80% of the time-t dataset.

However, instead of resorting to the simple average of the mean squared error (MSE) for the evaluation of the cross validation prediction error, the ratio of the average and the standard deviation of the MSE was used: $\frac{MSE_{avg}}{MSE_{std}}$. Indeed, this metric allows to more efficiently select the hyper-parameters combination that has both a low prediction bias and a low prediction variance (i.e. generalize better to new observations).

In practice, a random search with 500 iterations was used to generate the combination of hyper-parameters that were evaluated via the cross validation. Bergstra and Bengio (2012) show empirically and theoretically that randomly chosen trials are more efficient for HP optimization than trials on a user-specified grid.

For the sake of exhaustiveness, the tuning of hyper-parameters is performed for *each* configuration described in the next subsection. More precisely, for each risk aversion level, horizon of predicted returns, label transformation, etc., we evaluate which boosted tree architecture is the best.

In Figure 3, we plot the dispersion of tuned HPs across the three methods we evaluate. Overall, the inter-quartile ranges are not very large, which means that HPs do not vary too much across configurations. This is particularly true for the maximum depth, which is overwhelmingly equal to two. The number of trees oscillates around 60, which is sufficiently small to prevent overfitting, especially with tree depths of two.

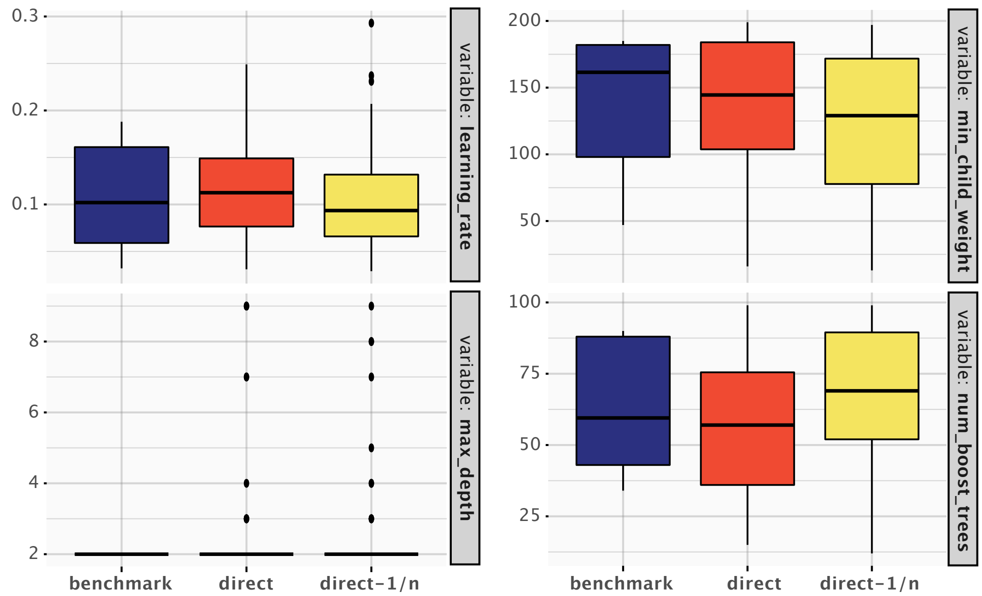

Figure 3: **Distribution of hyper-parameters**. We illustrate the distribution of HPs in boxplots. Each point corresponds to a combination of risk aversion and horizon of forecasted returns.

4.3 Investment strategies

4.3.1 Portfolio construction parameters

The first 5 years of the dataset (i.e. from January 1997 to December 2001) are used as a buffer period to fine-tune the hyper-parameters using K-fold cross-validation (cf. previous paragraph). These cross-validated hyper-parameters are then used to compare the various strategies on an out-of-sample period that ranges between January 2002 and September 2020.

The weights are computed every Friday, at close of business (CoB) time. Returns are also calculated on a weekly basis between each Monday CoB (using portfolio weights from the previous Friday). It replicates a strategy where the portfolio managers evaluate new weights on every Monday's morning and invest on Monday's close.

Prediction models are re-estimated every semester on samples of expanding windows. More precisely, each semester, the training dataset is incremented with the last 6 months of data and a new model is fitted on it (with fixed HPs values).

We allow for five additional degrees of freedom:

- the **horizon** of the returns used in the optimization and the learning stages. Three values are tested, namely 21, 63 and 126 days.
- whether or not to apply the **hyperbolic tangent transform** to the labels, as defined in Equation (13).
- the level of **risk aversion**, γ in (11), for which we test three values: $\gamma \in \{1, 3, 10\}$ (low, moderate and high).

- the **sample size** for the estimation of the covariance matrix (for the traditional approach only). Again, we test three depths: 3, 6 and 12 months.
- the **stringency of the box constraint** of optimal portfolio weights. All policies are long only, meaning that the minimal weight is larger or equal to zero. In order to impose diversification, we also impose weight constraints. For the 10 industry and 25 sorted portfolios, we enforce a maximum weight so that all weights are smaller than some upper threshold which we fix contingently on the investment universe, as described in Table 2 below. For the 4 asset class universe, we fix reasonable target weights in Table 3, and the constraints ensure that the weights remain in the vicinity of the benchmark values. For instance, if the target weight is 52%, the 30% constraint imposes the interval $52\% \times [1.3, 0.7] = [67.6\%, 36.4\%]$.

Dataset	**Upper bound on weights**		
	Strong	Intermediate	Loose
10 industries	20%	30%	40%
25 sorted portfolios	10%	15%	20%
	Relative leeway on benchmark		
4 asset classes	10%	20%	30%

Table 2: **Summary of constraints**

Asset class	**Target benchmark weight**
Developed Markets - Equities	52%
Emerging Markets - Equities	8%
Global Markets - Sovereign Bonds	20%
Global Markets - Corporate Bonds	20%

Table 3: **Benchmark weights for the 4 asset classes**

Overall, for the traditional method, this makes 162 combinations while for the direct ones, we carry out 54 versions of the portfolios. The results for each combination of these options will be systematically reported, or averaged to ease clarity of exposition. The first two (horizon and risk aversion) will always be reported, which will make nine types (or clusters) of portfolios. For some indicators, we will only document the average across the other degrees of freedom.

4.3.2 Performance criteria

The portfolio strategies are evaluated on a traditional set of performance measures:

- The **transaction cost-adjusted Sharpe Ratio** ($\mathcal{SR}$):

$$\mathcal{SR} = \frac{\tilde{\mu} - r_f}{\sigma} \tag{14}$$

where $\tilde{\mu}$ is the expected return of the strategy, net of trading costs, and σ the standard deviation and r_f the risk-free rate.[11] $\tilde{\mu}$ is computed by replacing the gross performance with the performance net of trading costs. More precisely, in this case, the weekly returns are discounted by a factor as follows.

$$r_{net} = (1 - TO * TC) * (1 + r_{gross}) - 1$$

Where TC is the average estimated trading cost and r_{gross} the original weekly return. TC is set to 5 basis points (bps) for the first and third datasets, and to 15 bps for the 25 sorted portfolios. This is because the latter are likely harder to trade, compared to the other two, which can be packaged in ETFs. $TO = \sum_{i=1}^{n} |w_{i,t} - w_{i,t-1}^{FwdAdj}|$ is the weekly turnover where $w_{i,t-1}^{Fwd\_Adj}$ is the portfolio weight from previous Friday CoB, which is forward adjusted to take into account weights deviation between 2 rebalancing dates.

- The **maximum drawdown** ($\mathcal{MDD}$) is an indicator of permanent loss of capital. It measures the largest single drop from peak to bottom in the value of a portfolio. In brief, the $\mathcal{MDD}$ offers investors a worst case scenario:

$$\mathcal{MDD}(T) = \max_{\tau \in (0,T)} \left[\max_{t \in (0,\tau)} X(t) - X(\tau) \right] \tag{15}$$

  Because it is a second measure of risk, we will report it as a fraction of volatility ($|\mathcal{MDD}|/\sigma$), which corresponds to the worst drawdown of the portfolio, for a given unit of volatility. This ratio shows the extreme risk after controlling for σ and is thus a truly complementary risk metric.

- The average weekly **turnover** ($\mathcal{ATO}$), simply defined as:

$$\mathcal{ATO} = \frac{1}{T} \sum_{t=1}^{T} \left\{ \sum_{i=1}^{n} |w_{i,t} - w_{i,t-1}^{FwdAdj}| \right\}. \tag{16}$$

  It will evaluate the magnitude of asset rotation in the portfolios. High rotation is naturally associated with higher trading costs.

4.4 Data Snooping and statistical testing

Harvey (2017) and Harvey and Liu (2021) contend that most claimed research findings in financial economics are likely false due to data snooping.[12] Data snooping occurs when the same data set is employed more than once for inference and model selection. It leads to the possibility that any successful results may be spurious because they could be due to chance (White (2000)). In other words, looking long and hard enough at a given data set will often reveal one or more models that seems promising but are in fact spurious.

Diebold (2015) notes that, unfortunately, many studies employ the Diebold and Mariano (1995) test for comparing models in (pseudo-) out-of-sample environments, and concludes: "*unavoidable and crucially-important issues arise, related both to finite-sample*

[11] Given the values of r over the timeframe of our study, a risk-free interest rate of zero is assumed when calculating the $\mathcal{SR}$.

[12] In full transparency, this claim is disputed in Chen (2021) and Jensen et al. (2021).

analysis versus asymptotic analysis, and more importantly, to comparisons of two models versus many models". To alleviate this issue, the model confidence set (MCS) of Hansen et al. (2011) proposes a model selection algorithm, which sequentially filters a set of models from a given entirety of models. It aims to first find the best model from a group and, second, to test if other models are statistically indistinguishable from this best candidate.

The primary output is a set of p-values, where models with a p-value above the size are selected in the best models set. Small p-values indicate that the model is easily rejected from the set that includes the best. In this paper, the MCS is applied with a risk-adjusted profit maximization loss function (three-year rolling Sharpe ratio).

5 Empirical results

In this section, we show that predicting the optimal weights directly instead of the traditional two-step approach leads to more stable portfolios, with better risk-adjusted performance measures.

5.1 Sharpe ratios

Our first important set of results pertains to the MCS p-values which we present in Figure 4. The results are clustered according to two degrees of freedom: the horizon of the returns to be predicted (columns of graphs) and the risk aversion parameter (rows of graphs). In addition, each data set (i.e., investment universe) has its own horizontal panel.

The interpretation is as follows. The y-axis shows the transaction-cost adjusted Sharpe ratio defined in Equation (14). The x-axis corresponds to the p-value of the test for which the null hypothesis is that a portfolio is comparable to the *best* portfolio. The test is performed at the cluster level, so that the best portfolio is always located at the top right of the nine sub-panels for each universe.

When the risk aversion is low or moderate ($\gamma \in \{1, 3\}$) - in the upper and middle panels), the traditional approach is outperformed by the direct ones because almost all dark blue points lie below (and to the left of) the yellow squares and orange triangles. However, for high values of γ, when the volatility term is predominant, this is no longer true and the results become in favor of the traditional approach. Note that for high γ values, the contribution of the expected return forecast term in the optimizer is highly reduced, in comparison to the risk term. In such case, the optimized weights are closer to a minimum variance allocation.

In Tables 4 and 5, we provide the values of the indicators for all three investment universes. This allows to access the two terms of the Sharpe ratio. We see that supervised portfolios are linked to higher average returns and to higher volatilities, which is why their Sharpe ratio is comparable (though slightly superior) to those of the benchmark method.

5.2 Maximum drawdown

In the present paper, we assess the risk of portfolio strategies via the most extreme and conservative measure, the maximum drawdown, which we depict in Figure 5. The MDD is the second risk metric we report after the volatility. Because the two are linked, we choose to display the MDD, relative to the volatility. Therefore, the numbers we display

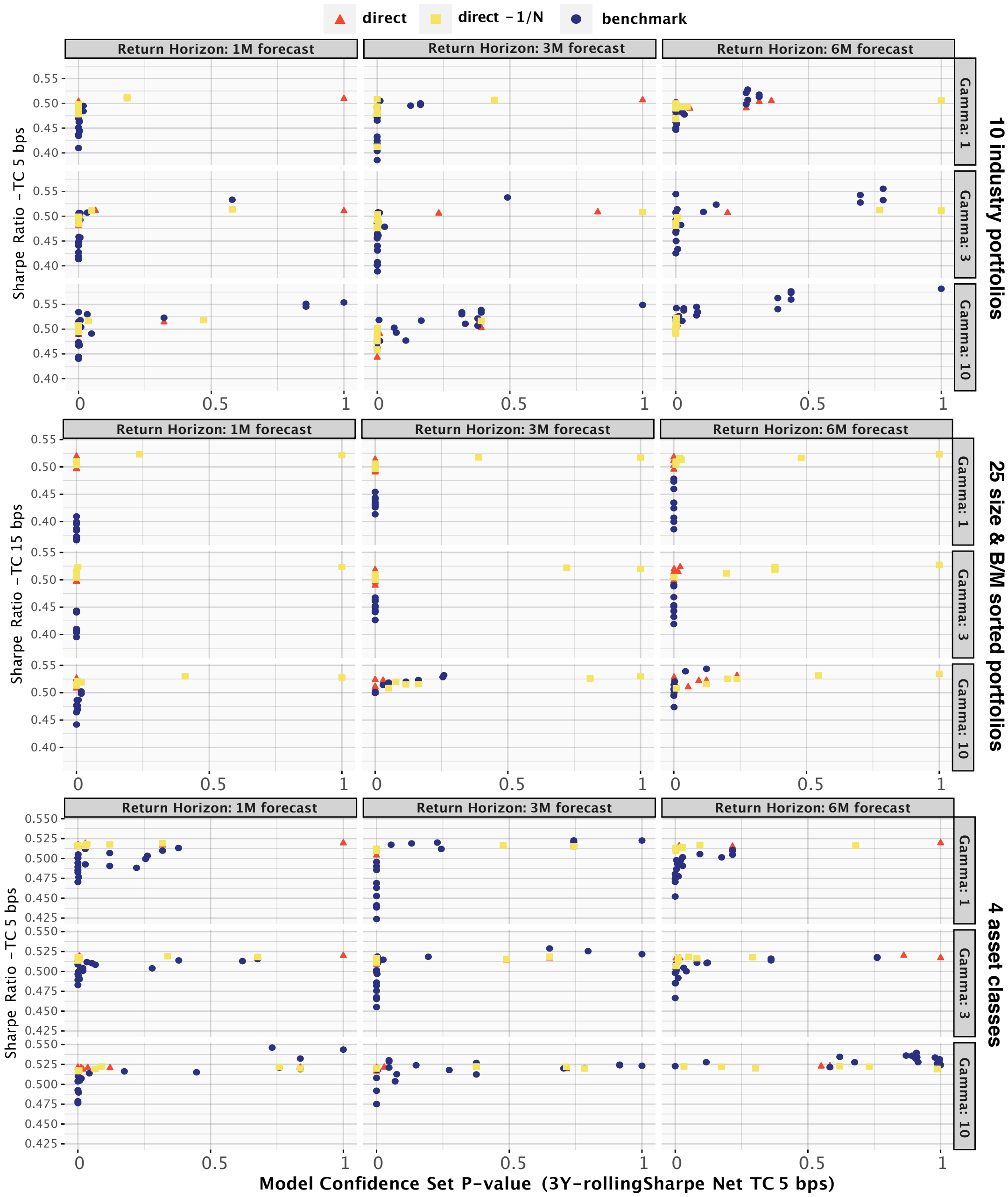

Figure 4: **Transaction cost-adjusted Sharpe ratio and MCS p-values.** We use a 3-year rolling Sharpe Ratio net of trading cost (5 or 15 bps, depending on the universe) as loss function for the MCS by group of *Return horizon* (columns of the figure) and *Gamma* (rows in the panels). Then we plot the realized Sharpe Ratio net of trading cost over the full period. The best performing strategies by cluster are in the top right corner of each cell. Colors and shapes code the methodology for the portfolio construction.

pertain to extreme risk, after controlling for baseline risk. One way to think about this is to imagine a portfolio manager who has a risk budget (say, a vol of 10%) and wants to

Dataset	Type	Horiz.	γ	$\mathcal{SR}$	Ann. ret.	Ann. vol.	$\frac{\text{Max DD.}}{\text{vol.}}$	Avg. Weekly TO
10 industries	Trad.	1M	1	0.468	0.092	0.197	2.618	0.579
			3	0.473	0.087	0.184	2.614	0.555
			10	0.503	0.084	0.167	2.650	0.463
		3M	1	0.463	0.092	0.198	2.487	0.504
			3	0.459	0.086	0.186	2.526	0.482
			10	0.505	0.085	0.168	2.636	0.385
		6M	1	0.485	0.092	0.189	2.570	0.434
			3	0.497	0.089	0.178	2.534	0.395
			10	0.541	0.089	0.165	2.582	0.310
	Direct	1M	1	0.498	0.096	0.193	2.807	0.067
			3	0.499	0.096	0.192	2.817	0.060
			10	0.505	0.095	0.188	2.825	0.076
		3M	1	0.493	0.095	0.193	2.785	0.066
			3	0.485	0.093	0.192	2.789	0.087
			10	0.491	0.092	0.187	2.797	0.091
		6M	1	0.494	0.095	0.192	2.799	0.085
			3	0.494	0.094	0.189	2.825	0.093
			10	0.506	0.093	0.184	2.835	0.105
	Direct - 1/N	1M	1	0.496	0.096	0.193	2.818	0.087
			3	0.500	0.096	0.192	2.815	0.063
			10	0.507	0.095	0.188	2.828	0.060
		3M	1	0.481	0.093	0.194	2.780	0.096
			3	0.491	0.094	0.192	2.787	0.076
			10	0.493	0.092	0.187	2.792	0.090
		6M	1	0.491	0.094	0.192	2.804	0.095
			3	0.495	0.094	0.190	2.807	0.102
			10	0.507	0.093	0.184	2.853	0.082
4 asset classes	Trad.	1M	1	0.496	0.057	0.116	3.513	0.059
			3	0.503	0.057	0.114	3.459	0.060
			10	0.512	0.056	0.109	3.549	0.061
		3M	1	0.491	0.056	0.115	3.420	0.062
			3	0.500	0.056	0.112	3.424	0.063
			10	0.516	0.055	0.107	3.472	0.057
		6M	1	0.490	0.057	0.116	3.446	0.055
			3	0.502	0.057	0.113	3.432	0.056
			10	0.530	0.056	0.106	3.474	0.055
	Direct	1M	1	0.518	0.058	0.112	3.619	0.021
			3	0.518	0.058	0.112	3.618	0.021
			10	0.522	0.058	0.111	3.625	0.021
		3M	1	0.512	0.058	0.113	3.588	0.021
			3	0.513	0.058	0.112	3.594	0.021
			10	0.520	0.057	0.110	3.609	0.021
		6M	1	0.516	0.058	0.112	3.601	0.025
			3	0.518	0.058	0.111	3.603	0.025
			10	0.522	0.057	0.110	3.609	0.024
	Direct - 1/N	1M	1	0.517	0.058	0.112	3.616	0.022
			3	0.517	0.058	0.112	3.614	0.022
			10	0.520	0.058	0.111	3.621	0.022
		3M	1	0.512	0.058	0.113	3.577	0.022
			3	0.515	0.058	0.112	3.583	0.023
			10	0.521	0.057	0.110	3.592	0.022
		6M	1	0.514	0.058	0.112	3.588	0.026
			3	0.515	0.057	0.111	3.592	0.026
			10	0.522	0.057	0.110	3.590	0.026

Table 4: **Average summary statistics**. We gather the descriptive statistics of the combinations of portfolios tested in the paper. The trading cost-adjusted Sharpe ratio, $\mathcal{SR}$ is defined in Equation (14). The annual return corresponds to the numerator of the Sharpe ratio. Similarly, the annualized volatility is simply the standard deviation of weekly returns, scaled for annualization and corresponds to the denominator of the Sharpe ratio. The maximum drawdown and the average turnover $\mathcal{ATO}$ are specified in Equations (15) and (16), respectively. Note that we standardized the maximum drawdown by the volatility so that it can be comparable in terms of risk budget. All indicators are averaged at the cluster level.

Dataset	Type	Horiz.	γ	$\mathcal{SR}$	Ann. ret.	Ann. vol.	$\frac{\text{Max DD.}}{\text{vol.}}$	Avg. Weekly TO
25 portfolios	Trad.	1M	1	0.384	0.081	0.210	2.667	0.640
			3	0.418	0.086	0.205	2.738	0.585
			10	0.478	0.095	0.198	2.890	0.432
		3M	1	0.433	0.091	0.210	2.673	0.526
			3	0.450	0.092	0.205	2.731	0.482
			10	0.515	0.102	0.198	2.870	0.348
		6M	1	0.433	0.090	0.208	2.744	0.450
			3	0.460	0.094	0.203	2.760	0.406
			10	0.512	0.102	0.198	2.863	0.306
	direct	1M	1	0.508	0.114	0.225	2.769	0.079
			3	0.510	0.114	0.224	2.782	0.079
			10	0.518	0.114	0.220	2.813	0.082
		3M	1	0.503	0.113	0.224	2.796	0.091
			3	0.506	0.112	0.222	2.817	0.090
			10	0.515	0.112	0.217	2.879	0.090
		6M	1	0.510	0.114	0.223	2.806	0.088
			3	0.514	0.113	0.221	2.828	0.085
			10	0.520	0.112	0.215	2.893	0.082
	direct - 1/N	1M	1	0.512	0.115	0.225	2.765	0.068
			3	0.514	0.115	0.224	2.774	0.068
			10	0.521	0.115	0.221	2.806	0.071
		3M	1	0.507	0.114	0.224	2.789	0.076
			3	0.511	0.114	0.222	2.809	0.076
			10	0.519	0.113	0.217	2.866	0.077
		6M	1	0.514	0.115	0.223	2.799	0.074
			3	0.517	0.114	0.221	2.819	0.072
			10	0.523	0.113	0.216	2.885	0.069

Table 5: **Average summary statistics - continued**. We gather the descriptive statistics of the combinations of portfolios tested in the paper. The trading cost-adjusted Sharpe ratio, $\mathcal{SR}$ is defined in Equation (14). The annual return corresponds to the numerator of the Sharpe ratio. Similarly, the annualized volatility is simply the standard deviation of weekly returns, scaled for annualization and corresponds to the denominator of the Sharpe ratio. The maximum drawdown and the average turnover $\mathcal{ATO}$ are specified in Equations (15) and (16), respectively. Note that we standardized the maximum drawdown by the volatility so that it can be comparable in terms of risk budget. All indicators are averaged at the cluster level.

compare the drawdowns linked to this fixed level of dispersion in returns.

The individual values of relative drawdowns are averaged across degrees of freedom so as to display clustered values. The exhaustive results for each combination and each universe are available upon request.

One interesting property is that the relative drawdown is not very sensitive to γ with the direct methods, as if the supervision focused on expected returns, regardless of risk aversion. Thus, when increasing γ, both simple risks and extreme risks can be curtailed. Overall, in spite of a few exceptions, the traditional approach appears less extremely risky than the direct approaches. Also, both direct methods yield very comparable results. This was expected, because the number of assets is invariant, hence the demeaning of weights adds little value in this case.

As a point of comparison, we provide in Table 8 of Appendix B the average metrics pertaining to the portfolios computed with knowledge of realized returns. Because these in-sample portfolios remain subject to the same constraints as our baseline allocations, the relative drawdowns remain substantial (between 1.4 and 3.3 times the volatility, across all

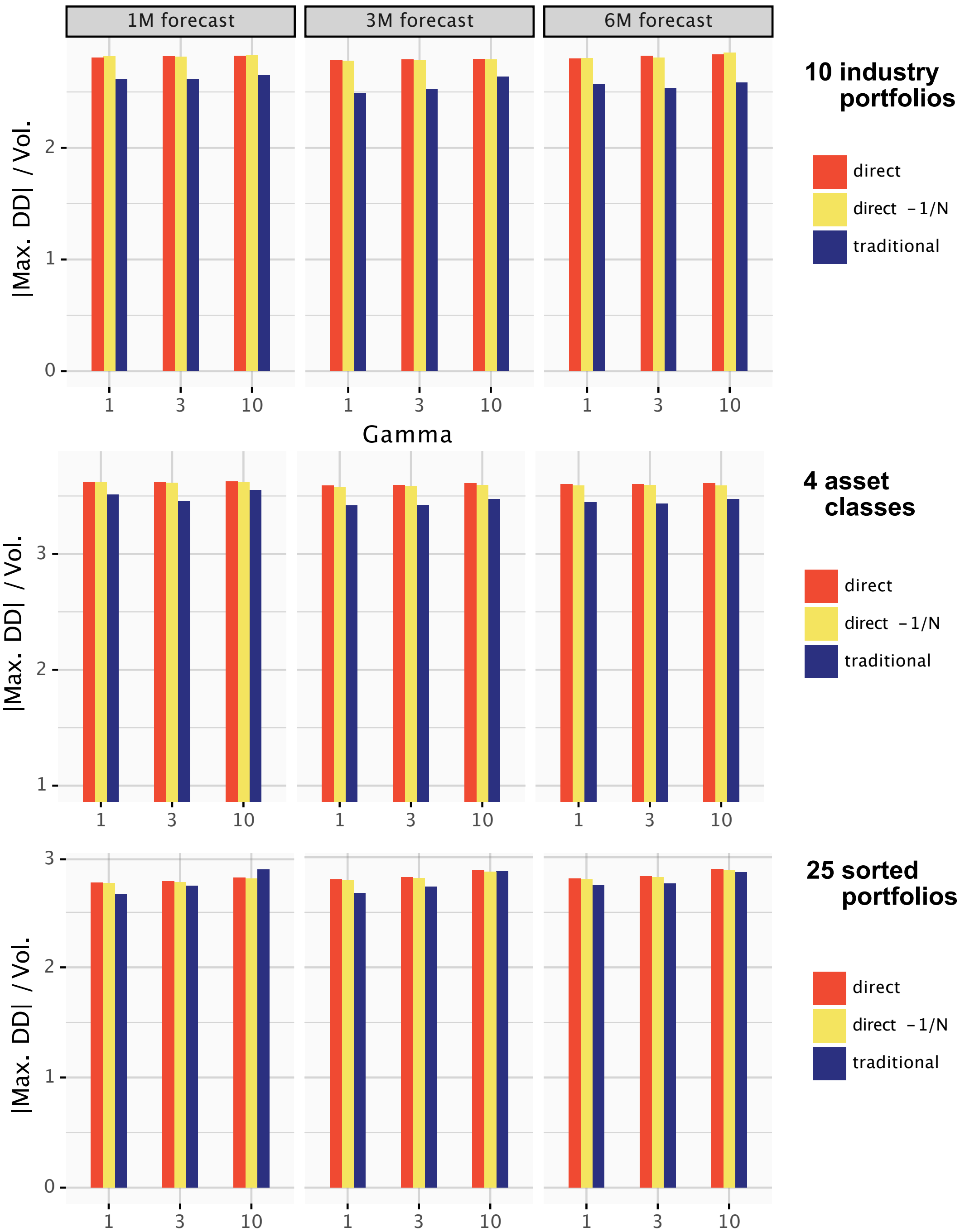

Figure 5: **Relative maximum drawdown**. We report the average of maximum drawdowns net of trading cost (5 or 15 bps, depending on the universe) divided by volatility, within each *Return horizon* and *Gamma* cluster (i.e., average across weight constraints, hyperbolic tangent transform, and covariance matrix sample depth for the traditional approach).

universes). This shows that any reasonable allocation (especially in the equity space) was subject to strong losses at one point during the backtest (see also our sub-period analysis in Section 6.1).

Moreover, to shed additional light on the differences between the direct and indirect methods, we focus on the timing of the maximum drawdown for the four asset classes in Figure 6. We note that for three out of the four classes, it occurs around the end of 2007 and the beginning of 2008, during the subprime financial crisis. We observe that the traditional approach is more reactive in turbulent times, often with binary switches that saturate the investment constraints. For developed market equities, the direct method lowers the exposure slowly and never to the minimum authorized level. For emerging equities however, the position is stable, while the benchmark model hesitates and oscillates. For sovereign bonds, the direct approach fares better, but the magnitude of the drawdown (-8%) is less severe compared to the other classes. Overall it seems that reactivity favors the traditional approach for that matter.

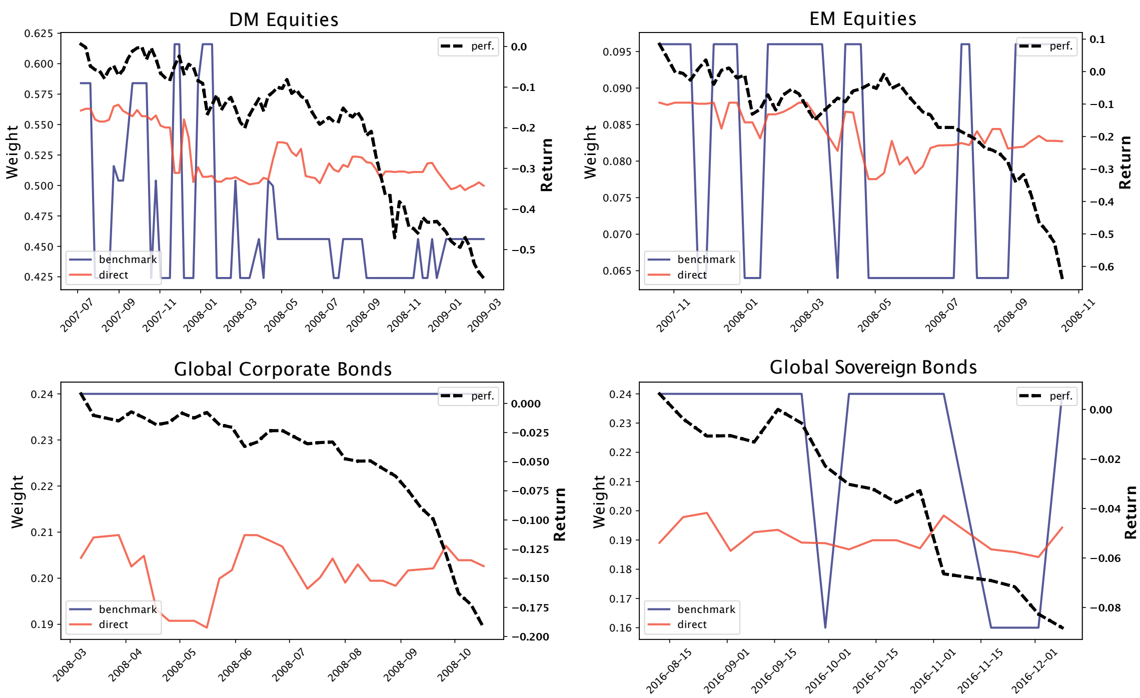

Figure 6: **Maximum drawdown of the four asset classes**. We display the period of the maximum drawdown of the 4 assets alongside the corresponding portfolio weights (left axis) for the two methods: benchmark in blue and direct in red. The dark dotted line shows the cumulative return of the asset class (right axis). The plot corresponds to the case of relative constraint of 20%, so that the maximum weight for developed market equities can be $52\% \times 1.2 = 62.4\%$ and the minimum can be $52\% \times 0.8 = 41.6\%$; 52% being the benchmark target.

5.3 Turnover

In Figure 7, we plot the average turnover of the strategies. The turnover metrics are shown by approach type and each color codes a constraint intensity. Naturally, the looser the constraint, the higher the turnover, on average. The most striking pattern in the figure is the substantially lower turnover generated by the direct estimation of the strategies.

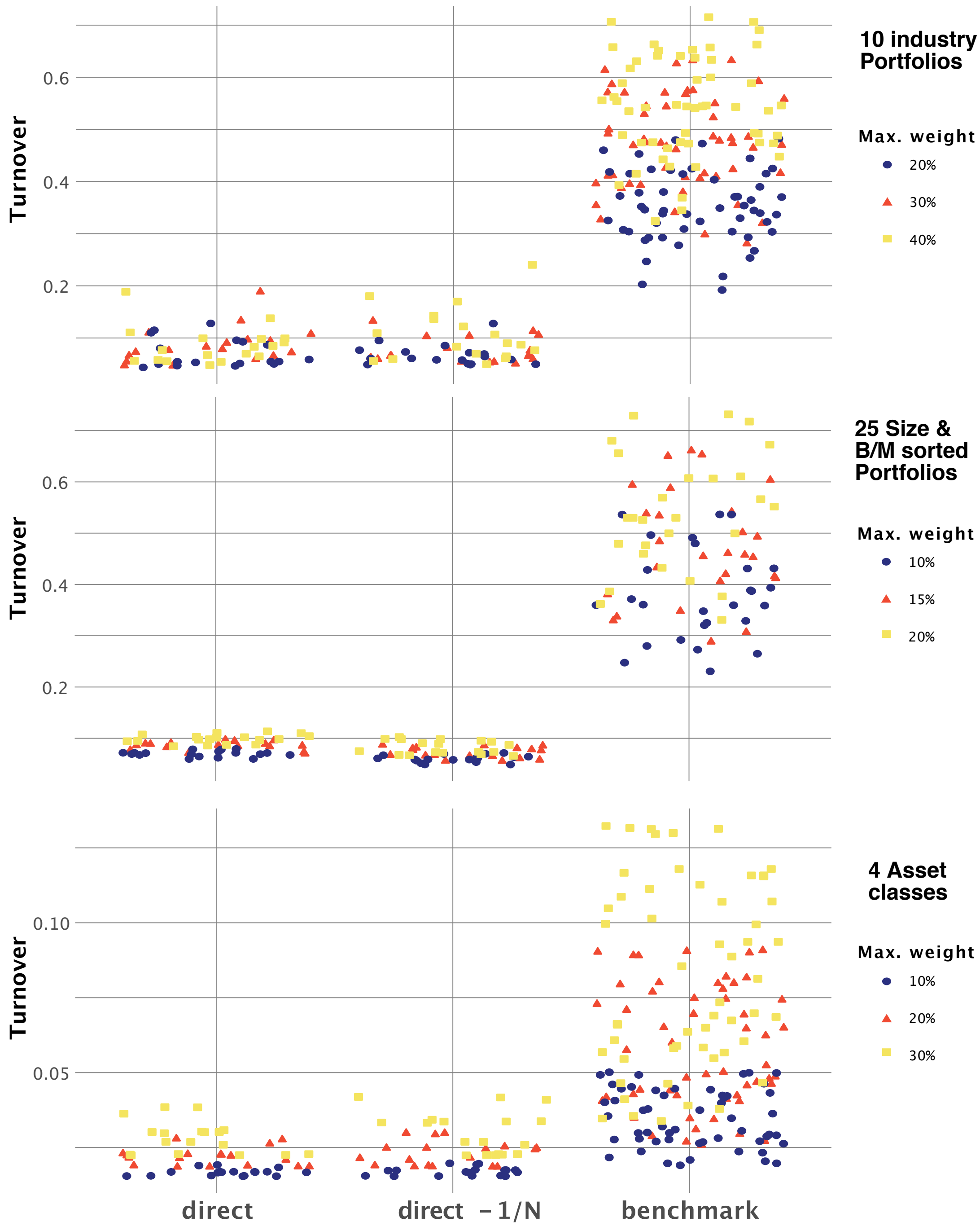

Figure 7: **Average Weekly Turnover**. We report the average weekly asset rotation for each strategy tested. The legend on the right corresponds to the maximum weight constraints used in the optimizers.

We observe a clear difference between the investment approaches. Note that since the traditional approach requires to specify the backward looking horizon window for the covariance matrix, there are three times more points in the *benchmark* column since we tested 3 different horizon windows.

In Figure 8 in the Appendix, we depict the evolution of weights for the first dataset. Each one of the ten assets has its own panel. Traditional weights are shown in red and directly supervised weights are plotted in blue. The former often hit the edges of the box constraints: this comes from the optimizer finding an corner solution. The result is an allocation that is often binary in the assets (either fully invested or with zero weight in the portfolio), which generates increased rotation and much higher turnover. In comparison, the weights that are learned directly exhibit much smoother patterns and thus, reduced trading intensity. This has major consequences because taking on-off positions is impossible in practice: asset managers fine-tune their positions with small shifts in their portfolios rather than extreme changes.

6 Robustness checks

6.1 Sub-period analysis

An interesting question for investors pertains to the behavior of their portfolios during times of financial turmoil. In Table 6 below, we compute the average returns realized in recent periods of market turbulence. The values are averaged over the degrees of freedom of the protocol: prediction horizon and risk aversion parameter.

The various methods yield relatively comparable results, and the latter are rather universe and crisis dependent. For instance, results are quite close during the Lehman crisis, favorable to the direct approach during the Brexit and the Covid-19 rebound. In the midst of the Covid-19 crash, results are only favorable to the direct method for the 4 asset class universe. Overall, our conclusions are not much impacted when looking at particular important moments in our samples.

6.2 Linear models

The results we have presented so far rely on boosted trees as the driving engine for supervision. Therefore, it is possible that our findings are contingent on this choice. The most natural alternative to tree methods would be neural networks, but they would require many architectural choices. Rather, we propose to evaluate the sensitivity of our results to the learning engine by resorting to the simplest model there is: the linear regression.

In Table 7, we gather the performance indicators obtained when replacing boosted trees by simple linear models. We report only the simple direct approach, as it is equivalent to the corrected one in the case of linear models. A first observation is that Sharpe ratios are overwhelmingly lower, compared to those of Table 4 and 5, meaning, that more sophisticated algorithms do imply improved forecasting ability. Nevertheless, we note that maximum drawdowns are lower with linear models.

When comparing the two approaches, the outcomes are very universe-dependent. For example, with the 4 asset classes, both yield similar Sharpe ratios. However, for the other two datasets, there is a very significant superiority of our direct approach. These additional findings corroborate the usefulness of the core idea outlined in the paper.

Type	Subprime crisis	Brexit	Covid-19	Covid-19 Rebound
PANEL A: 10 industries				
Trad.	-0.112	0.047	-0.211	0.053
direct	-0.129	0.052	-0.225	0.083
direct - 1/N	-0.129	0.052	-0.225	0.083
PANEL B: 4 asset classes				
Trad.	-0.109	0.035	-0.142	0.033
direct	-0.110	0.036	-0.128	0.035
direct - 1/N	-0.110	0.036	-0.127	0.034
PANEL 5: 25 sorted portfolios				
Trad.	-0.140	0.053	-0.229	0.084
direct	-0.141	0.059	-0.273	0.100
direct - 1/N	-0.142	0.059	-0.274	0.100

Table 6: **Crisis Performance**. We report the average of realized returns of the portfolios, across all combinations of horizons and risk aversions. The subsamples are defined as follows: Subprime: 2008-09-26 → 2008-10-10; Brexit: 2016-06-23 → 2016-06-27; Covid-19: 2020-02-19 → 2020-03-23; Covid-19 rebound: 2020-03-23 → 2020-04-27.

7 Conclusion

In this article, we propose a general, integrated, framework for portfolio construction based on supervised learning. Traditionally, the supervision is only focused on mining expected returns which are subsequently sent to an optimization scheme in order to derive optimal portfolio weights. We advocate to process the weights *before* the supervision stage.

This allows the algorithm to learn from past time series of in-sample optimal weights and to infer the best weights from variables such as past performance, risk, and proxies of the macro-economic outlook.

Our open source empirical analysis shows that this idea is beneficial mostly when the focus of the optimization is on the first moment of returns, that is, when the risk aversion is low in the mean-variance framework. In this case, the trading cost-adjusted out-of-sample Sharpe ratios of our directly supervised portfolios outperform those stemming from the traditional approach. The former have larger average returns, along with higher volatilities, but the net effect is in their favor overall.

Given the sensitivity of results to trading costs, a direction for future research would be to include them directly into the optimizers and loss functions. This would allow to further control asset rotation. This is left for future research.

Another open field of investigation pertains to the theoretical reasons underpinning the superiority of direct estimation versus the traditional approach. One key element is the shift in the returns' distributions between the training phase and the testing phase (actual trading). However, the channels through which this may favor the direct estimation remain unclear. This is also fertile ground for future work.

Dataset	Type	Horiz.	γ	$\mathcal{SR}$	Ann. ret.	Ann. vol.	$\frac{\text{Max DD.}}{\text{vol.}}$	Avg. Weekly TO
10 industries	Trad.	1M	1	0.406	0.082	0.203	2.720	0.573
			3	0.407	0.079	0.193	2.749	0.555
			10	0.450	0.079	0.175	2.734	0.485
		3M	1	0.402	0.081	0.202	2.586	0.505
			3	0.392	0.075	0.193	2.629	0.489
			10	0.439	0.076	0.173	2.654	0.426
		6M	1	0.353	0.069	0.195	2.523	0.460
			3	0.384	0.071	0.184	2.554	0.434
			10	0.468	0.080	0.170	2.627	0.373
	Direct	1M	1	0.478	0.092	0.193	2.811	0.107
			3	0.479	0.092	0.192	2.814	0.107
			10	0.483	0.090	0.187	2.825	0.107
		3M	1	0.457	0.089	0.195	2.775	0.139
			3	0.460	0.089	0.193	2.780	0.136
			10	0.472	0.088	0.187	2.802	0.130
		6M	1	0.470	0.090	0.191	2.751	0.141
			3	0.473	0.089	0.188	2.766	0.136
			10	0.483	0.087	0.181	2.799	0.128
4 asset classes	Trad.	1M	1	0.503	0.054	0.108	3.654	0.090
			3	0.493	0.053	0.107	3.708	0.087
			10	0.510	0.053	0.104	3.677	0.078
		3M	1	0.481	0.052	0.109	3.471	0.078
			3	0.491	0.053	0.107	3.481	0.076
			10	0.499	0.052	0.104	3.560	0.066
		6M	1	0.484	0.053	0.109	3.438	0.070
			3	0.496	0.054	0.108	3.468	0.068
			10	0.511	0.053	0.104	3.539	0.060
	Direct	1M	1	0.503	0.057	0.113	3.614	0.030
			3	0.504	0.057	0.113	3.613	0.030
			10	0.507	0.057	0.112	3.616	0.030
		3M	1	0.501	0.057	0.113	3.581	0.031
			3	0.505	0.057	0.112	3.585	0.031
			10	0.509	0.056	0.110	3.600	0.032
		6M	1	0.502	0.056	0.112	3.562	0.036
			3	0.504	0.056	0.112	3.565	0.036
			10	0.507	0.056	0.110	3.599	0.035
25 portfolios	Trad.	1M	1	0.389	0.084	0.216	2.679	0.576
			3	0.405	0.085	0.211	2.694	0.542
			10	0.464	0.094	0.202	2.759	0.449
		3M	1	0.429	0.094	0.220	2.681	0.459
			3	0.434	0.093	0.215	2.712	0.436
			10	0.459	0.095	0.206	2.824	0.368
		6M	1	0.461	0.101	0.220	2.667	0.374
			3	0.467	0.100	0.215	2.666	0.353
			10	0.483	0.100	0.207	2.775	0.302
	Direct	1M	1	0.515	0.116	0.225	2.739	0.116
			3	0.519	0.116	0.223	2.747	0.115
			10	0.527	0.115	0.219	2.778	0.113
		3M	1	0.504	0.113	0.225	2.748	0.126
			3	0.508	0.113	0.223	2.767	0.124
			10	0.519	0.113	0.217	2.829	0.124
		6M	1	0.518	0.116	0.224	2.778	0.121
			3	0.519	0.115	0.221	2.810	0.121
			10	0.527	0.113	0.215	2.878	0.119

Table 7: **Average summary statistics using a linear regression model**. We gather the descriptive statistics of the combinations of portfolios tested in the paper. The trading cost-adjusted Sharpe ratio, $\mathcal{SR}$ is defined in Equation (14). The annual return corresponds to the numerator of the Sharpe ratio. Similarly, the annualized volatility is simply the standard deviation of weekly returns, scaled for annualization and corresponds to the denominator of the Sharpe ratio. The maximum drawdown and the average turnover $\mathcal{ATO}$ are specified in Equations (15) and (16), respectively. Note that we standardized the maximum drawdown by the volatility so that it can be comparable in terms of risk budget. All indicators are averaged at the cluster level.

Acknowledgments

The authors are grateful for the comments of two referees which have helped clarify and improve the empirical part of the paper.

A Sector weights

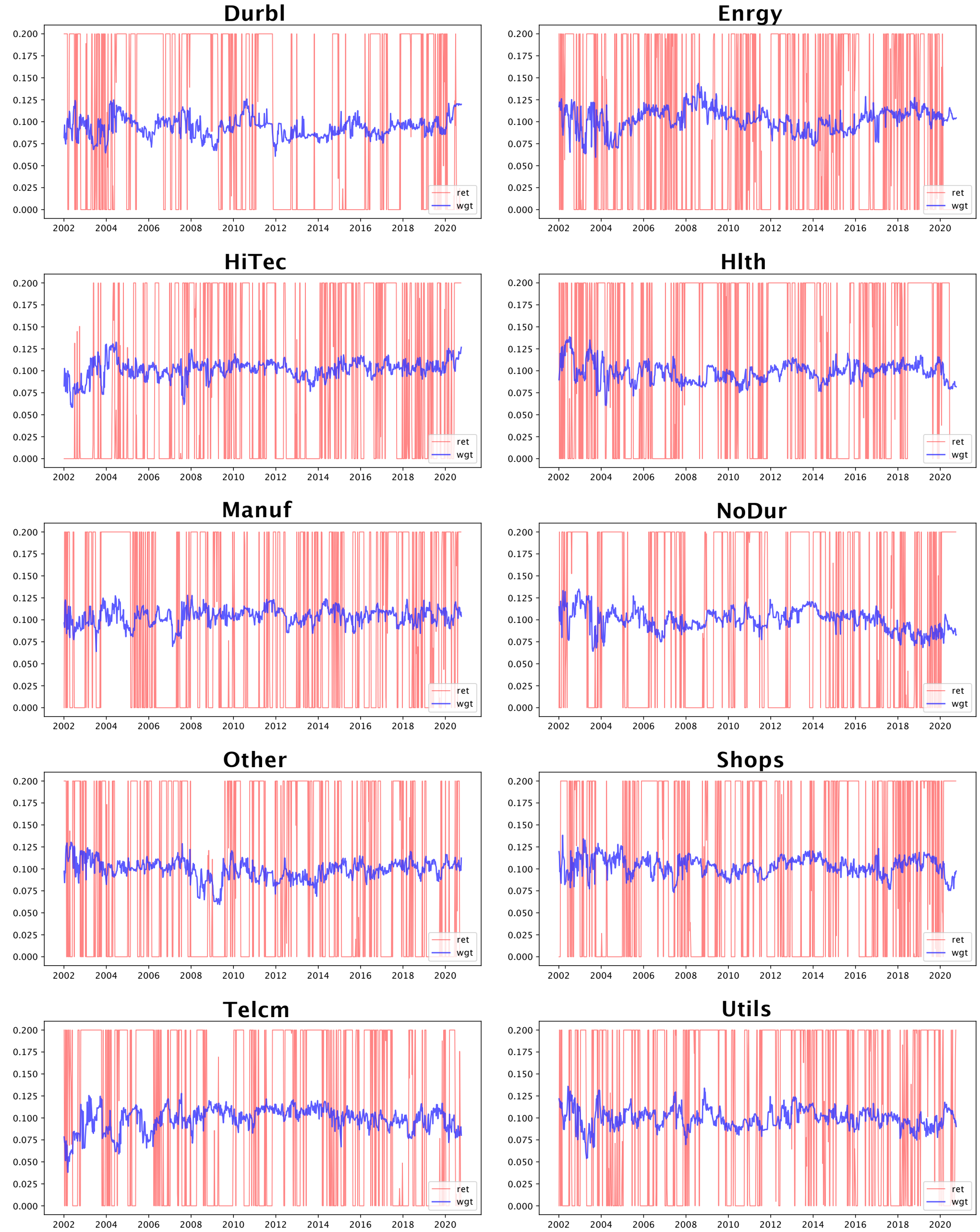

Figure 8: **Strategy exposures by sectors**. We compare the sector exposure between the benchmark (red) and direct estimation (blue). We use 63 days for the horizon window as well as for the covariance matrix lookback window. The maximum weight constraint is 20% and the risk aversion factor gamma = 1

B Perfect foresight (in-sample results)

Horiz.	γ	$\mathcal{SR}$	Ann. ret.	Ann. vol.	$\frac{\text{Max DD.}}{\text{vol.}}$	Avg. Weekly TO
PANEL A: 10 industries						
1M	1	2.210	0.434	0.196	1.521	0.633
	3	2.259	0.434	0.192	1.489	0.627
	10	2.302	0.422	0.183	1.551	0.604
3M	1	1.435	0.289	0.201	1.849	0.383
	3	1.482	0.290	0.195	1.852	0.384
	10	1.561	0.282	0.181	1.889	0.371
6M	1	1.222	0.250	0.204	2.082	0.293
	3	1.274	0.251	0.196	2.134	0.288
	10	1.311	0.239	0.182	2.041	0.269
PANEL B: 4 asset classes						
1M	1	0.944	0.103	0.109	3.124	0.097
	3	0.947	0.103	0.109	3.132	0.097
	10	0.949	0.102	0.108	3.132	0.093
3M	1	0.787	0.088	0.111	3.200	0.057
	3	0.808	0.089	0.110	3.241	0.058
	10	0.821	0.089	0.108	3.200	0.054
6M	1	0.704	0.079	0.112	3.272	0.042
	3	0.727	0.080	0.111	3.252	0.041
	10	0.751	0.081	0.108	3.283	0.039
PANEL C: 25 sorted portfolios						
1M	1	1.692	0.383	0.226	1.704	0.728
	3	1.702	0.380	0.223	1.697	0.724
	10	1.709	0.366	0.214	1.850	0.699
3M	1	1.223	0.278	0.227	1.999	0.433
	3	1.249	0.276	0.221	1.947	0.429
	10	1.242	0.260	0.209	2.027	0.399
6M	1	1.000	0.229	0.229	2.222	0.304
	3	1.018	0.227	0.223	2.241	0.299
	10	1.016	0.214	0.210	2.276	0.273

Table 8: **In-sample average summary statistics (perfect foresight target)**. We report the average metrics pertaining to the portfolios computed with knowledge of realized first moments of returns. The portfolios are subject to the same constraints as the baseline allocations.

References

Ammann, M., G. Coqueret, and J.-P. Schade (2016). Characteristics-based portfolio choice with leverage constraints. *Journal of Banking & Finance 70*, 23–37.

André, E. and G. Coqueret (2020). Dirichlet policies for reinforced factor portfolios. *SSRN Working Paper 3726714*.

Asness, C. S., T. J. Moskowitz, and L. H. Pedersen (2013). Value and momentum everywhere. *Journal of Finance 68*(3), 929–985.

Baker, M., B. Bradley, and J. Wurgler (2011). Benchmarks as limits to arbitrage: Understanding the low-volatility anomaly. *Financial Analysts Journal 67*(1), 40–54.

Banz, R. W. (1981). The relationship between return and market value of common stocks. *Journal of Financial Economics 9*(1), 3–18.

Bergstra, J. and Y. Bengio (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research 13*(10), 281–305.

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

Butler, A. and R. Kwon (2021). Integrating prediction in mean-variance portfolio optimization. *SSRN Working Paper 3788875*.

Campbell, J. Y. and L. M. Viceira (2002). *Strategic asset allocation: Portfolio choice for long-term investors*. Oxford University Press.

Cenesizoglu, T. and A. Timmermann (2012). Do return prediction models add economic value? *Journal of Banking & Finance 36*(11), 2974–2987.

Chaieb, I., H. Langlois, and O. Scaillet (2021). Factors and risk premia in individual international stock returns. *Journal of Financial Economics 141*(2), 669–692.

Chen, A. Y. (2021). Most claimed statistical findings in cross-sectional return predictability are likely true. *SSRN Working Paper 3912915*.

Cong, L. W., K. Tang, J. Wang, and Y. Zhang (2022). Alphaportfolio: Direct construction through deep reinforcement learning and interpretable ai. *SSRN Working Paper 3554486*.

Coqueret, G. (2015). Diversified minimum-variance portfolios. *Annals of Finance 11*(2), 221–241.

Coqueret, G. (2021). Persistence in factor-based supervised learning models. *Journal of Finance and Data Science Forthcoming*.

Coqueret, G. and T. Guida (2020). *Machine Learning for Factor Investing: R Version*. CRC Press.

Costa, G. and G. N. Iyengar (2022). Distributionally robust end-to-end portfolio construction. *arXiv Preprint* (2206.05134).

Diebold, F. and R. Mariano (1995). Comparing predictive accuracy. *Journal of Business And Economic Statistics 13*(3), 253–63.

Diebold, F. X. (2015). Comparing predictive accuracy, twenty years later: A personal perspective on the use and abuse of diebold–mariano tests. *Journal of Business & Economic Statistics 33*(1), 1–1.

Elmachtoub, A. N. and P. Grigas (2022). Smart "predict, then optimize". *Management Science 68*(1), 9–26.

Fama, E. and K. French (1992). The cross-section of expected stock returns. *Journal of Finance 47*(2), 427–465.

Fama, E. F. and K. R. French (2015). A five-factor asset pricing model. *Journal of Financial Economics 116*(1), 1–22.

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

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

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

Goetzmann, W. N. and A. Kumar (2008). Equity portfolio diversification. *Review of Finance 12*(3), 433–463.

Golosnoy, V. and B. Gribisch (2022). Modeling and forecasting realized portfolio weights. *Journal of Banking & Finance Forthcoming*, 106404.

Grinsztajn, L., E. Oyallon, and G. Varoquaux (2022). Why do tree-based models still outperform deep learning on tabular data? *arXiv Preprint* (2207.08815).

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

Guida, T., G. Coqueret, et al. (2018). Ensemble learning applied to quant equity: gradient boosting in a multifactor framework. *Big data and machine learning in quantitative investment*, 129–148.

Han, Y., Y. Liu, G. Zhou, and Y. Zhu (2021). Technical analysis in the stock market: A review. *SSRN Working Paper 3850494*.

Han, Y., K. Yang, and G. Zhou (2013). A new anomaly: The cross-sectional profitability of technical analysis. *Journal of Financial and Quantitative Analysis 48*(5), 1433–1461.

Han, Y., G. Zhou, and Y. Zhu (2016). A trend factor: Any economic gains from using information over investment horizons? *Journal of Financial Economics 122*(2), 352–375.

Hansen, P., A. Lunde, and J. Nason (2011). The model confidence set. *Econometrica 79*(2), 453–497.

Harvey, C. R. (2017). Presidential address: The scientific outlook in financial economics. *The Journal of Finance 72*(4), 1399–1440.

Harvey, C. R., J. C. Liechty, M. W. Liechty, and P. Müller (2010). Portfolio selection with higher moments. *Quantitative Finance 10*(5), 469–485.

Harvey, C. R. and Y. Liu (2021). Uncovering the iceberg from its tip: A model of publication bias and p-hacking. *SSRN Working Paper 3865813*.

He, X., L. W. Cong, G. Feng, and J. He (2021). Asset pricing with panel trees under global split criteria. *SSRN Working Paper 3949463*.

Hjalmarsson, E. and P. Manchev (2012). Characteristic-based mean-variance portfolio choice. *Journal of Banking & Finance 36*(5), 1392–1401.

Hull, B. and X. Qiao (2017). A practitioner's defense of return predictability. *The Journal of Portfolio Management 43*(3), 60–76.

Jagannathan, R. and T. Ma (2003). Risk reduction in large portfolios: Why imposing the wrong constraints helps. *Journal of Finance 58*(4), 1651–1683.

Januschowski, T., Y. Wang, K. Torkkola, T. Erkkilä, H. Hasson, and J. Gasthaus (2021). Forecasting with trees. *International Journal of Forecasting Forthcoming*.

Jegadeesh, N. and S. Titman (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *Journal of Finance 48*(1), 65–91.

Jensen, T. I., B. T. Kelly, and L. H. Pedersen (2021). Is there a replication crisis in finance? *Journal of Finance Forthcoming*.

Jorion, P. (1985). International portfolio diversification with estimation risk. *Journal of Business 58*(3), 259–278.

Kan, R. and D. R. Smith (2008). The distribution of the sample minimum-variance frontier. *Management Science 54*(7), 1364–1380.

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.

Krauss, C., X. A. Do, and N. Huck (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500. *European Journal of Operational Research 259*(2), 689–702.

Langlois, H. (2020). A new benchmark for dynamic mean-variance portfolio allocations. *SSRN Working Paper 3548138*.

Li, X., R. N. Sullivan, and L. Garcia-Feijóo (2016). The low-volatility anomaly: Market evidence on systematic risk vs. mispricing. *Financial Analysts Journal 72*(1), 36–47.

López de Prado, M. (2016). Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management 42*(4), 59–69.

López de Prado, M. (2018). *Advances in Financial Machine Learning*. John Wiley & Sons.

Makridakis, S., E. Spiliotis, and V. Assimakopoulos (2022). The M5 accuracy competition: Results, findings and conclusions. *Int J Forecast Forthcoming*.

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

Moody, J., L. Wu, Y. Liao, and M. Saffell (1998). Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting 17*(5-6), 441–470.

Nadarajah, S. and T. K. Pogány (2016). On the distribution of the product of correlated normal random variables. *Comptes Rendus Mathematique 354*(2), 201–204.

Raffinot, T. (2017). Hierarchical clustering-based asset allocation. *Journal of Portfolio Management 44*(2), 89–99.

Rapach, D. E., J. K. Strauss, J. Tu, and G. Zhou (2019). Industry return predictability: A machine learning approach. *Journal of Financial Data Science 1*(3), 9–28.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning 5*(2), 197–227.

Shwartz-Ziv, R. and A. Armon (2022). Tabular data: Deep learning is not all you need. *Information Fusion 81*, 84–90.

Simon, F., S. Weibels, and T. Zimmermann (2022). Deep parametric portfolio policies. *SSRN Working Paper 4150292*.

Uysal, S., X. Li, and J. M. Mulvey (2021). End-to-end risk budgeting portfolio optimization with neural networks. *SSRN Working Paper 3883614*.

Van Dijk, M. A. (2011). Is size dead? A review of the size effect in equity returns. *Journal of Banking & Finance 35*(12), 3263–3274.

Welch, I. and A. Goyal (2008). A comprehensive look at the empirical performance of equity premium prediction. *Review of Financial Studies 21*(4), 1455–1508.

White, H. (2000, September). A Reality Check for Data Snooping. *Econometrica 68*(5), 1097–1126.

Zhang, Z., S. Zohren, and S. Roberts (2020a). Deep learning for portfolio optimization. *Journal of Financial Data Science 2*(4), 8–20.

Zhang, Z., S. Zohren, and S. Roberts (2020b). Deep reinforcement learning for trading. *Journal of Financial Data Science 2*(2), 25–40.