ARTICLE IN PRESS

International Journal of Forecasting xxx (xxxx) xxx

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Single-stage portfolio optimization with automated machine learning for M6☆

Xinyu Huang [a], David P. Newton [a], Emmanouil Platanakis [a], Charles Sutcliffe [b,*]

[a] University of Bath - School of Management, United Kingdom
[b] ICMA Centre, Henley Business School, University of Reading, United Kingdom

ARTICLE INFO

Article history:
Dataset link: https://github.com/xh646-alt/M6_APO_codes/

Keywords:
Automated machine learning
Portfolio selection
Investment analysis
Estimation risk
Parameter uncertainty

ABSTRACT

The goal of the M6 forecasting competition was to shed light on the efficient market hypothesis by evaluating the forecasting abilities of participants and performance of their investment strategies. In this paper, we challenge the 'estimate-then-optimize' approach with one that directly optimizes portfolio weights from data. We frame portfolio selection as a constrained penalized regression problem. We present a data-driven approach that automatically performs model selection and hyperparameter tuning to maximize the objective without noisy or potentially misspecified intermediate steps. Finally, we show how the portfolio weights can be optimized using the Method of Moving Asymptotes. Testing on the M6 competition data, our approach achieves a global rate of return of 9.5% and an information ratio of 5.045, which is in stark contrast to the mean IR of the M6 competition teams of −3.421 and the IR of 0.453 for the M6 benchmark.

1. Introduction

The M6 forecasting competition aimed to test the efficient market hypothesis by assessing participants' ability to predict the performance rankings of financial assets and use these predictions to guide investment decisions. The competition included 100 financial assets, divided equally between 50 stocks and 50 exchange-traded funds (ETFs). It featured two main tasks: probabilistic forecasting, where participants ranked future returns of the assets, and investment decision-making, where they proposed portfolio allocations. This paper focuses on the investment decision-making component and introduces an automated optimization framework designed to forecast portfolio weights directly.

Modern portfolio theory, as proposed by Markowitz (1952) in his mean–variance framework, involves a two-step process: first, estimating the statistical moments of the asset returns, and second, optimizing over possible combinations of assets based on these estimates. In practice, this approach leads to several problems when creating portfolios. These include extreme asset weightings, highly concentrated portfolios, sensitivity to input parameters, and poor out-of-sample performance (Best & Grauer, 1991; Board & Sutcliffe, 1994; Grauer & Shen, 2000; Green & Hollifield, 1992; Harris et al., 2017; Platanakis et al., 2021; Platanakis & Urquhart, 2019). A study by DeMiguel et al. (2009) examining 14 variants of the standard mean–variance model across various datasets found that none consistently outperform the 1/N strategy in terms of Sharpe ratio or certainty-equivalent return. These difficulties primarily arise from errors in estimating model inputs. The mean–variance framework relies on precise knowledge of the mean returns and the covariance matrix of assets, which are generally unknown and

☆ We are grateful to this journal's editor, an associate editor, and two anonymous referees for their helpful and constructive comments, which helped us improve the paper significantly.

The numerical results presented in this manuscript were reproduced by the Editor-in-Chief on the 4th of September 2024.

* Corresponding author.
E-mail addresses: xh357@bath.ac.uk (X. Huang), D.P.Newton@bath.ac.uk (D.P. Newton), E.Platanakis@bath.ac.uk (E. Platanakis), c.sutcliffe@icmacentre.ac.uk (C. Sutcliffe).

https://doi.org/10.1016/j.ijforecast.2024.08.004
0169-2070/

must be estimated from market data. However, accurately estimating returns is challenging due to the need for extensive time series data, and estimates of the variance–covariance matrix are often prone to misspecification errors (Cong et al., 2020).

In the past few years, there has been a growing interest in applying big data and machine learning (ML) to the cross-section prediction of asset returns. Fischer and Krauss (2018) use Long Short-Term Memory networks to predict directional movements for the S&P500 constituent stocks from 1992 until 2015, achieving daily returns of 0.46% and an annualized Sharpe ratio of 5.8 before transaction costs. Gu et al. (2020) develop a trading system for all CRSP stocks from 1987 to 2016. Using 94 firm-specific features, 74 industry dummies, and eight macroeconomic predictors to identify potentially mispriced stocks, they achieve an annualized out-of-sample Sharpe ratio of 2.45. Wu et al. (2021) show that hedge fund portfolios in the top decile selected by ML methods consistently outperform the corresponding style-specific HFR Index across equity hedge, event-driven, and relative value fund styles. Bianchi et al. (2021) find strong statistical evidence that ML models can successfully predict Treasury bond returns. DeMiguel et al. (2023) show that ML-based selection of mutual fund portfolios yielded significant out-of-sample annual alphas of 2.4% net of all costs. Bali et al. (2023) show that using a dataset of over 12 million observations and 273 characteristics from options and stocks, a long-short trading strategy informed by nonlinear ML predictions remains highly profitable even after accounting for high transaction costs.

Despite the proliferation of idiosyncratic characteristics and ML to explain the cross-section of asset returns, recent work has challenged the credibility of the evidence on stock return predictability. Cakici et al. (2023) examine stock return predictability for ML across 46 countries and find that OLS has better risk-adjusted performance than eight other ML methods, including random forest and two neural network types. The superior performance of OLS in their study highlights the severe impact of estimation error, suggesting that the benefit of capturing non-linearities and nonlinear interactions in ML methods may be smaller than expected. Harvey et al. (2016) find that 80 to 158 of the 296 published anomalies are likely false discoveries. Huck (2019) argues that more predictors mean potentially more noise – using 600 predictors and up to 300 US large-cap stocks, the ML models considering only the stock lagged returns as predictors show the highest returns. The author also finds that when rebalancing the portfolio frequently, 50% of ML models tested produce negative alphas after adjusting for traditional risk factors. Avramov et al. (2023) report similar findings - when reasonable transaction costs are considered, ML faces difficulties in achieving statistically and economically significant risk-adjusted returns due to high turnover and extreme positions in the tangency portfolio implied by the pricing kernel. In other words, employing ML algorithms and big data (e.g., hundreds of predictors) for return predictions is not guaranteed to boost portfolio performance.

Brandt (1999) highlights the critical relationship between statistical models, investment returns, and investor decision-making. When investors rely on estimates from the implied conditional distribution of returns, any inaccuracies in the model's assumptions about the relationship between returns and forecasting variables can lead to flawed portfolios and investment choices. The complexities of financial markets, including high dimensionality, noise, intricate interactions, and non-stationary dynamics, make it challenging for investors to accurately approximate the underlying data-generating process of equity risk premiums (Cong et al., 2020). This complexity makes it difficult to exploit potentially mispriced assets, thus limiting opportunities for consistently outperforming the market through active trading or stock selection. This may explain why participants using data-driven methods, such as time-series analysis and machine learning, had an average information ratio of −3.374 in the M6 competition.

We propose an ML-based approach to directly estimate portfolio weights from the data to avoid additional errors or misspecification in the intermediate step. We design an automated and dynamic trading rule that frames portfolio selection as a constrained regression problem, where the regression coefficients determine the weights of the assets. The goal is to minimize the generalization errors (i.e., the deviation of portfolio returns from a reference return) through an optimal choice of asset weights. The function that connects the portfolio's objective and asset weights is assumed to be time-varying. Rather than fitting model parameters based on a fixed structure and hyperparameters, we broaden our search to include algorithm and hyperparameter selection. During each training session, we evaluate various combinations of model structures and hyperparameters to find the optimal solution for the portfolio objective. We then assess validation errors and make trading decisions based on the configuration with the lowest validation error. We perform a fresh search for model structures and hyperparameters for each submission point and rebalance the portfolio accordingly.

In out-of-sample forecasting, multicollinearity often subjects the regression coefficients to large estimation errors, rendering the coefficients biased and inefficient. To get around multicollinearity, we augment the objective function with penalties based on (1) the sum of the absolute values of portfolio weights, (2) the sum of the squared values of portfolio weights, or (3) a combination of both. These penalties are modulated by a regularization coefficient, allowing for flexible impact adjustment. As a result, the regularized portfolios tend to be sparse, featuring fewer active positions and lower transaction costs.

To determine the optimal portfolio weights under a budget constraint, we use the Method of Moving Asymptotes (MMA) introduced by Svanberg (1987). MMA is a convex programming algorithm for nonlinear optimization tasks with multiple equality and inequality constraints. The idea is to construct and solve a series of subproblems, where convex approximations expand the objective function and constraints in explicit form. This approach effectively handles general nonlinear programming problems. It can accommodate various constraints, provided that the derivatives of the constraint functions concerning the design variables (in this case, the portfolio weights) are computable.

Our main contributions can be summarized as:

- We present an automated and dynamic trading strategy that optimizes portfolio weights. Recognizing the challenge of finding a statistical model that precisely links asset returns with predictive variables, we employ a data-driven method that monitors the pseudo-outcome of trading actions (validation error) and adapts model selection dynamically to optimize the objective. This dynamic approach allows for greater flexibility in approximating the true underlying regression function, potentially leading to improved performance.
- We add a penalty term to the objective function to combat multicollinearity and account for transaction costs. Additionally, our framework accommodates various portfolio constraints, e.g., the sum of portfolio weights must be less than or equal to one.
- We implement the Method of Moving Asymptotes to determine the optimal weights. We calculate the derivatives of the objective and constraint functions for each ML algorithm considered in this study. We then illustrate how MMA uses linearization, specifically the first-order Taylor expansion, to approximate both the objective and constraint functions. To our knowledge, this is the first study to implement MMA for portfolio design and optimization.

Based on the M6 data and the specified out-of-sample period, the proposed automated portfolio optimization (APO) strategy achieves a notable return of 9.5% and an information ratio (IR) of 5.045 across the entire sample. This is a significant improvement compared to the mean IR of −3.421 achieved by the M6 competition teams and the M6 forecasting benchmark of 0.453. We also analyze data from the broad S&P500 index constituents to compare APO with variants of the mean–variance model. The APO strategy outperforms all competing allocation rules examined in our study, with and without considering transaction costs. Additionally, we investigate the sources of performance advantage by comparing APO with (1) traditional mean–variance models with similar shrinkage techniques and (2) APO methods using more conventional optimization approaches, the Splitting Conic Solver (SCS) and the Embedded Conic Solver (ECOS). Our experiments validate the effectiveness of APO's modeling and optimization methods.

The remainder of the paper is organized as follows. Section 2 explores the relationship between machine learning objectives and portfolio optimization and introduces the APO framework. Section 3 details the data used in the study. Section 4 presents the results of our out-of-sample tests, while Section 5 discusses the additional analysis performed. Section 6 provides the concluding remarks.

2. Methodology

2.1. The role of ML in the optimization problem

In the broader context of regression problems, the future value of a target variable r can be expressed as a prediction error model:

$$r_{t+1} = E(r_{t+1}) + \epsilon_{t+1}, \tag{1}$$

where the random error (ϵ_{t+1}) is typically assumed to have an expected value of zero. Therefore, r_{t+1} can be forecast using its conditional expectation: $E(r_{t+1}) = g_t^*(\boldsymbol{x}_t, \boldsymbol{\beta})$, where g_t^* is a flexible function of the model's inputs $\boldsymbol{x}_t$ and a parameter set $\boldsymbol{\beta}$. $\boldsymbol{x}_t$ is an N-dimensional vector of predictive variables containing past information gathered at or before period t. The forecasting function g_t^* describes the relationship between input features $\boldsymbol{x}_t$ and the target variable r_{t+1}, which is assumed to be time-varying. Our objective is to minimize the generalization error $F(\boldsymbol{\beta})$ for realized r_{t+1} by using a standard least squares objective function described as follows:

$$F(\boldsymbol{\beta}) = E[(r_{t+1} - g_t^*(\boldsymbol{x}_t, \boldsymbol{\beta}))^2]. \tag{2}$$

In the context of portfolio optimization, suppose investors invest a portion of their wealth w_f in the risk-free asset with a given return r_f and invest another portion of their wealth $\boldsymbol{w} \in \mathbb{R}^N$ among N risky assets, with excess returns of over r_f denoted by the vector $\boldsymbol{x}$, so that $w_f + \mathbf{1}^\top \boldsymbol{w} = 1$. In theory, the optimal strategy is to design a portfolio rule $\hat{\boldsymbol{w}}$ that minimizes the utility loss $L(\boldsymbol{w}^*, \hat{\boldsymbol{w}})$ resulting from estimation risk:

$$\min L(\boldsymbol{w}^*, \hat{\boldsymbol{w}}) = U(\boldsymbol{w}^*) - E[U(\hat{\boldsymbol{w}})], \tag{3}$$

where $U(\boldsymbol{w}^*)$ is the expected utility of the true optimal portfolio rule $\boldsymbol{w}^*$, and $E[U(\hat{\boldsymbol{w}})]$ is the expected utility of $\hat{\boldsymbol{w}}$. Assuming investors have mean–variance preferences described by the utility function $U(r)$ over portfolio return r, characterized by a quadratic utility function: $U(r) = r - \frac{1}{2}\delta r^2$, where $\delta > 0$ is the risk aversion level, and r is the portfolio return. Kinn (2018) shows that maximizing the expected utility of a quadratic utility function when selecting portfolio weights $\boldsymbol{w}$ is equivalent to minimizing generalization error. Specifically, consider $\tilde{\boldsymbol{x}}$ as the returns of the risky assets, i.e., $\tilde{\boldsymbol{x}} = r_f + \boldsymbol{x}$, this equivalence is captured by the equation:

$$F(\boldsymbol{w}) = -E[U(r_f w_f + \tilde{\boldsymbol{x}}^\top \boldsymbol{w})] = E[(\bar{r} - \boldsymbol{x}^\top \boldsymbol{w})^2], \tag{4}$$

where $\bar{r}$ is a reference return calculated as $\bar{r} = (1 - \delta r_f)/\delta$. The expression in Eq. (4) explains why ML is well-suited to handle portfolio selection and forms the basis for our subsequent analysis. We can now interpret the target variable in Eq. (1) using the constant $\bar{r}$, and g_t^* using the linear function $\boldsymbol{x}^\top \boldsymbol{w}$. Therefore, the portfolio selection problem can be viewed as a regression task where the input features are the historical returns, and the coefficients are the estimated weights.

2.2. Optimizing portfolio weights using an automated system

Narrowing our focus to linear forecast functions, Eq. (1) can be expressed as follows:

$$r_{t+1} = g_t^*(\boldsymbol{x}_t; \boldsymbol{w}) + \epsilon_{t+1} = w_1 x_{1,t} + w_2 x_{2,t} + \cdots + w_N x_{N,t} + \epsilon_{t+1}, \tag{5}$$

where N represents the total number of assets in the investment universe. The coefficients w_j for $j = 1, 2, \ldots, N$ are estimated using ordinary least squares (OLS) for basic linear models. The OLS assumption of no multicollinearity implies that there should be no linear relationship among the independent variables, which suggests that asset returns should be uncorrelated. However, this assumption does not always hold true in practical scenarios with finite samples. Multicollinearity can lead to significant estimation errors, rendering the coefficients biased and inefficient.

We adopt a two-pronged strategy to get around with multicollinearity and obtain meaningful results for such ill-conditioned problems. First, we shrink the coefficients in each regression problem towards zero to mitigate extreme asset positions. Second, we employ subset selection to eliminate redundant stocks in each portfolio. This approach also naturally accounts for transaction costs by effectively reducing the number of trades and the volume of assets. Additionally, the simplicity and lower complexity of managing a sparse portfolio further decrease the need for frequent rebalancing, thus minimizing transaction costs associated with portfolio management. This cost efficiency is an inherent advantage of promoting sparsity in portfolio selection, as discussed in Brodie et al. (2009).

We thus seek to find a vector of portfolio weights that solves:

$$\arg\min_{\boldsymbol{w}\in\mathbb{R}^N} F(\boldsymbol{w}) = \arg\min_{\boldsymbol{w}\in\mathbb{R}^N} E[(\bar{r} - \boldsymbol{x}_i^\top \boldsymbol{w})^2] + \phi(\boldsymbol{w}), \tag{6}$$

where $\phi(\boldsymbol{w})$ corresponds to ridge, lasso, and elastic net regression:

$$\phi(\boldsymbol{w}) = \begin{cases} \frac{\lambda}{2}\sum_{j=1}^{N}(w_j)^2 & \text{Ridge} \\ \lambda\sum_{j=1}^{N}|w_j| & \text{Lasso} \\ \lambda(1-\rho)\sum_{j=1}^{N}|w_j| + \frac{\lambda}{2}\rho\sum_{j=1}^{N}(w_j)^2 & \text{Elastic net.} \end{cases} \tag{7}$$

Regularization is commonly used to mitigate overfitting in datasets featuring numerous input variables. Lasso regression (Tibshirani, 1996) employs a penalty proportional to the l1-norm term to regulate the sparsity of the estimated parameter vector $\boldsymbol{w}$. In contrast, ridge employs an l2-norm term, which pulls all estimated parameters toward zero without strictly enforcing exact zeros. This results in a dense parameter vector $\boldsymbol{w}$ and enhanced stability. Elastic net, with $0 < \rho < 1$, achieves a compromise between ridge and lasso regression as it combines the penalties of both. We set the parameter ρ to the default value of 0.5 to balance the strengths of both l1 and l2 penalties. This intermediate value effectively manages the trade-off between sparsity and coefficient shrinkage, as discussed in Gu et al. (2020). In all three regression models, the impact of regularization is adjusted by the hyperparameter λ. Further, given the data-driven nature of our framework, which automatically performs model selection and hyperparameter tuning to maximize the objective without potentially misspecified intermediate steps, we expect that APO will perform well when applied to both short—and long-range dependent data.

To find the optimal choice of algorithm and λ for prediction, we divide the dataset into three sub-samples: nine months for training, three months for validation, and one month for trading. We produce forecast errors over the validation set conditional on the estimated parameters from the training set. We evaluate the model fit for various combinations of algorithms and hyperparameter settings to search for the configuration that minimizes the mean squared error over the validation set. The chosen algorithm and λ value remain constant throughout the trading period. We do not cross-validate (randomly select independent subsets of data) to preserve the temporal structure of the data. Finally, the trading set contains samples excluded from the training and validation process. While this study focuses on three algorithms and one key hyperparameter for each algorithm, we present the general framework below for broader application.

Let $\boldsymbol{\mathcal{A}} = \{\mathcal{A}^1, \mathcal{A}^2, \ldots, \mathcal{A}^Q\}$ be a set of algorithms, where $\mathcal{A}^q$ denotes an algorithm with z hyperparameters. We denote the zth hyperparameter domain by Λ_z^q, and the overall hyperparameters configuration space as $\boldsymbol{\Lambda}^q = \Lambda_1^q \times \Lambda_2^q \times \cdots \times \Lambda_z^q$. A vector of hyperparameters is denoted by $\boldsymbol{\eta} \in \boldsymbol{\Lambda}^q$, and $\mathcal{A}^q$ with its hyperparameters instantiated to $\boldsymbol{\eta}$ is denoted by $\mathcal{A}^q_{\boldsymbol{\eta}}$. Furthermore, let $\mathcal{D} = \{(\boldsymbol{x}_1, \bar{r}), (\boldsymbol{x}_2, \bar{r}), \ldots, (\boldsymbol{x}_T, \bar{r})\}$ be the in-sample dataset, which is divided into a training set $\mathcal{D}_{train}$ and a validation set $\mathcal{D}_{valid}$. Finally, let $L(\mathcal{A}^q_{\boldsymbol{\eta}}, \mathcal{D}_{train}, \mathcal{D}_{valid})$ represent the loss associated with algorithm $\mathcal{A}^q$ and hyperparameters $\boldsymbol{\eta}$, trained on $\mathcal{D}_{train}$ and validated on $\mathcal{D}_{valid}$. Our goal is to identify the joint algorithm and hyperparameter settings that minimize the generalization error:

$$\mathcal{A}^*, \boldsymbol{\eta}^* = \arg\min_{\mathcal{A}^q\in\boldsymbol{\mathcal{A}}, \boldsymbol{\eta}\in\boldsymbol{\Lambda}^q} E_{(\mathcal{D}_{train},\mathcal{D}_{valid})}[L(\mathcal{A}^q_{\boldsymbol{\eta}}, \mathcal{D}_{train}, \mathcal{D}_{valid})]. \tag{8}$$

Applying the above process to the portfolio selection problem:

$$\arg\min_{\boldsymbol{w}_\alpha} F(\boldsymbol{w}) = E[(\bar{r} - g_t^*(\boldsymbol{x}_t, \boldsymbol{w}_\alpha, \lambda_\alpha, \text{algor}_\alpha))^2], \tag{9}$$

$$s.t. \sum_{j=1}^{N} w_j^\alpha - 1 \le 0, \quad w_j^\alpha \in [0, 1], \tag{10}$$

where algor_α is the ML algorithm indexed by α, $\boldsymbol{x}_t$ is the vector of input features, $\boldsymbol{w}_\alpha$ is the model parameters of algor_α, and λ_α is the model hyperparameter of algor_α. Note than Eq. (9) can be extended as $\arg\min_{\boldsymbol{w}_\alpha} F(\boldsymbol{w}) = \frac{1}{T}\sum_{t=1}^{T}[\bar{r} - g_t^*(\boldsymbol{x}_t, \boldsymbol{w}_\alpha, \lambda_\alpha, \text{algor}_\alpha)]^2$, we follow Brodie et al. (2009) and absorb the factor $\frac{1}{T}$ in the hyperparameter λ_α (see Fig. 1).

Conventional optimization methods, such as linear and quadratic programming, often struggle with the inherent complexity and high dimensionality of asset allocation problems. To address these challenges, a recent study by Ai et al. (2024) shows how to use Differential Evolution to optimize asset allocation and maximize returns under a risk constraint. This study employs the Method of Moving Asymptotes (MMA) to address the nonlinear programming problem. Our objective is to minimize the loss function (Eq. (9)) while satisfying a linear constraint on the sum of design variables ($\boldsymbol{w}$). The primary optimization problem is transformed throughout the MMA optimization process into a sub-problem within a neighboring domain bounded by lower and upper asymptotes. Specifically, let $\tilde{U}_j$ and $\tilde{L}_j$ be the upper and lower bound value for $\boldsymbol{w}_j$, for $j = 1, 2, \ldots, N$, MMA suggests that at

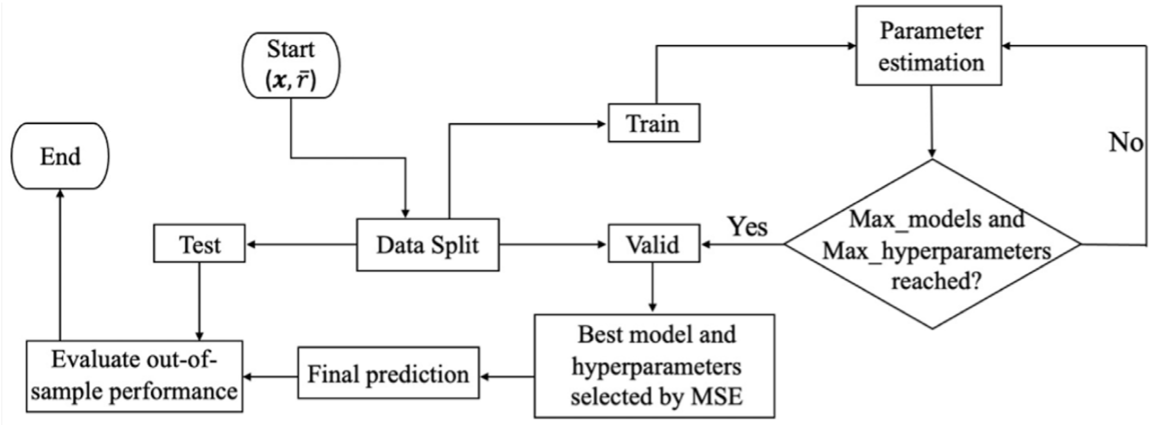

Fig. 1. Flow Chart of the APO framework.

each iteration k, the value of the objective function should be obtained by a linearization of the original objective function in variables of the form $1/(w_j - \tilde{L}_j)$ or $1/(\tilde{U}_j - w_j)$, depending on the signs of the derivatives evaluated at iteration k. This sub-problem is carefully formulated to ensure convexity and is rapidly solved using a dual method. Compared to alternative nonlinear programming methods like the Sequential Linear Approximation Algorithm and Quadratic Approximation Method, MMA stands out for its automatic asymptote adjustment. This feature enhances sub-problem convexity, improves local search capabilities, and accelerates convergence.

Let $\boldsymbol{X}$ be the training dataset, where the ith training sample is denoted by $\boldsymbol{x}_i$. Given prior hyperparameters λ_α, the MMA algorithm proceeds as follows:

Initialize: Assign initial values for candidate weights $\boldsymbol{w}^0 = (w_1^0, w_2^0, \ldots, w_N^0)$, let the iteration index $k = 0$.

Step 1: For an iteration k, determine the upper bound $\tilde{U}_j^k$ and lower bound $\tilde{L}_j^k$ for $w_j^k (j = 1, 2, \ldots, N)$. Calculate the values of the objective and constraint functions, along with their gradient information. In this study, we focus on lasso, ridge, and elastic net, where the objective function and its gradient are expressed as follows[1]:

$$F_0^{lasso}(\boldsymbol{w}^k) = \sum_{i=1}^{N}(\bar{r} - \boldsymbol{x}_i^\top \boldsymbol{w}^k)^2 + \lambda_\alpha \sum_{j=1}^{N}|w_j| \tag{11}$$

$$\frac{\partial F_0^{lasso}(\boldsymbol{w}^k)}{\partial w_j} = -2\boldsymbol{X}^\top(\bar{r}\mathbf{1}^\top - \boldsymbol{X}\boldsymbol{w}^k) + \lambda_\alpha \mathrm{sign}(\boldsymbol{w}^k) \tag{12}$$

$$F_0^{ridge}(\boldsymbol{w}^k) = \sum_{i=1}^{N}(\bar{r} - \boldsymbol{x}_i^\top \boldsymbol{w}^k)^2 + \frac{\lambda_\alpha}{2} \sum_{j=1}^{N}(w_j)^2 \tag{13}$$

$$\frac{\partial F_0^{ridge}(\boldsymbol{w}^k)}{\partial w_j} = -2\boldsymbol{X}^\top(\bar{r}\mathbf{1}^\top - \boldsymbol{X}\boldsymbol{w}^k) + \lambda_\alpha * \boldsymbol{w}^k \tag{14}$$

$$F_0^{enet}(\boldsymbol{w}^k) = \sum_{i=1}^{N}(\bar{r} - \boldsymbol{x}_i^\top \boldsymbol{w}^k)^2 + \lambda_\alpha(1-\rho)\sum_{j=1}^{N}|w_j| + \frac{\lambda_\alpha}{2}\rho \sum_{j=1}^{N}(w_j)^2 \tag{15}$$

$$\frac{\partial F_0^{enet}(\boldsymbol{w}^k)}{\partial w_j} = -2\boldsymbol{X}^\top(\bar{r}\mathbf{1}^\top - \boldsymbol{X}\boldsymbol{w}^k) + \lambda_\alpha(1-\rho)\mathrm{sign}(\boldsymbol{w}^k) + \lambda_\alpha \rho \boldsymbol{w}^k \tag{16}$$

For the budget constraint (Eq. (10)), the objective function $F_1(\boldsymbol{w}^k)$ and its gradient $\frac{\partial F_1(\boldsymbol{w}^k)}{\partial w_i}$ can be computed accordingly.

Step 2: Generate a sub-problem $\hat{P}(k)$ that by approximating explicit functions $F_i^k, i = 0, 1$ based on the calculations from step 1:

$$F_i^k(\boldsymbol{w}) = \tau_i^k + \sum_{j=1}^{N}\left(\frac{\hat{p}_{ij}^k}{\tilde{U}_j^k - w_j} + \frac{\hat{q}_{ij}^k}{w_j - \tilde{L}_j^k}\right) \tag{17}$$

$$\hat{p}_{ij}^k = \begin{cases} (\tilde{U}_j^k - w_j^k)^2 \cdot \frac{\partial F_i(\boldsymbol{w}^k)}{\partial w_j}, & \text{if } \frac{\partial F_i(\boldsymbol{w}^k)}{\partial w_j} > 0 \\ 0, & \text{if } \frac{\partial F_i(\boldsymbol{w}^k)}{\partial w_j} \le 0 \end{cases} \tag{18}$$

$$\hat{q}_{ij}^k = \begin{cases} 0, & \text{if } \frac{\partial F_i(\boldsymbol{w}^k)}{\partial w_j} \ge 0 \\ -(w_j^k - \tilde{L}_j^k)^2 \cdot \frac{\partial F_i(\boldsymbol{w}^k)}{\partial w_j}, & \text{if } \frac{\partial F_i(\boldsymbol{w}^k)}{\partial w_j} < 0 \end{cases} \tag{19}$$

$$\tau_i^k = F_i(\boldsymbol{w}^k) - \sum_{j=1}^{N}\left(\frac{\hat{p}_{ij}^k}{\tilde{U}_j^k - w_j^k} + \frac{\hat{q}_{ij}^k}{w_j^k - \tilde{L}_j^k}\right) \tag{20}$$

Step 3: Solve sub-problem $\hat{P}(k)$ and update the design variables $\boldsymbol{w}$.

Step 4: Evaluate the solution and check if convergence criteria are met or the maximum number of iterations is reached. If not, set the optimal solution of this sub-problem as the next iteration point $\boldsymbol{w}^k$. Let $k = k + 1$ and return to step 1.[2]

[1] For Eq. (11), Eq. (13), Eq. (15), the factor $1/T$ is absorbed in the parameter λ_α.

[2] We are grateful to Professor Krister Svanberg for kindly sharing the MMA code with us.

3. Data and estimation method

3.1. Data

First, we use the M6 forecasting competition dataset, which includes 50 S&P500 stocks and 50 international ETFs, offering a comprehensive representation of various asset categories and countries. We then extend our analysis to the entire S&P500 index constituents by randomly sampling N individual stocks for $N \in \{25, 50, 100\}$. We use lists of S&P500 constituents from July 3, 2021, to February 17, 2023, to eliminate survivor bias. We retrieved daily adjusted closing prices from the Wharton Research Data Services (WRDS) database for all stocks that were part of the index during this period. The out-of-sample testing for the S&P500 data starts on March 7, 2022, and runs through February 17, 2023, consistent with the M6 competition timeframe.

For each training session, we generate the input features and target as follows:

- Input features $\boldsymbol{x}_t$: We denote the adjusted closing price of asset i at the end of a trading day t as $P_{i,t}$. The return for asset i at period t is computed as: $x_{i,t} = \frac{P_{i,t}}{P_{i,t-1}} - 1$.
- Target variable $\bar{r}$: The reference return is defined as $\bar{r} = (1 - \delta r_f)/\delta$ where δ is the risk aversion coefficient. As noted in Makridakis et al. (2023), if the total absolute weights fall short of 100%.

3.2. Estimation strategy

For the M6 dataset, we use the past 252 days (approximately one trading year) of data for training and validation at each submission point. Forecasts of portfolio weights are generated recursively using a rolling window approach. We shift the training and validation samples forward in time to include more recent data while keeping the total number of periods fixed. The estimated portfolio weights remain unchanged throughout the trading period, with a 4-week holding period for each submission. For the S&P500 dataset, we implement APO in the same manner as for the M6 dataset and use the past 252 observations to estimate model parameters for the competing allocation strategies.

4. Empirical results

We now use the datasets discussed in Section 3 to evaluate the out-of-sample performance of APO versus the benchmark strategies. For the M6 dataset, we compare APO's performance against the benchmark set by the competition organizers and the average performance metrics of all the participating teams.

Turning to the S&P500 dataset, we evaluate the APO against well-established strategies from the existing literature. These include the volatility-timing (VT) strategy proposed by Kirby and Ostdiek (2012), the minimum-variance (MinVar) strategy, the classic mean–variance (MeanVar) strategy, the Bayes-Stein shrinkage strategy proposed by Jorion (1986), the Black-Litterman (BL) strategy proposed by Black and Litterman (1992) which incorporates market information with private investor insights, and the equal-weight strategy (1/N) highlighted by DeMiguel et al. (2009) as an incredibly rigorous benchmark. In Table 1, we provide details of implementing the benchmark strategies.

4.1. Evaluation metrics

To ensure comparability with the M6 competition results, we follow the methodology outlined by the M6 organization team for evaluating investment performance:

- Daily portfolio holding period return, $RET_t = \sum_{i=1}^{N} w_i(\frac{P_{i,t}}{P_{i,t-1}} - 1)$, where w_i is asset i's weight, $P_{i,t}$ is the adjusted closing price of asset i on trading day t.
- Continuously compounded portfolio returns: $ret_t = ln(1 + RET_t)$. For holding periods exceeding one day, returns are aggregated as: $ret_{t_1:t_2} = \sum_{t=t_1}^{t_2} ret_t$.
- Portfolio variance for holding from t_1 to t_2: $VaR_{t_1:t_2} = \frac{1}{K-1}\sum_{t=t_1}^{t_2}(ret_t - T^{-1}ret_{t_1:t_2})^2$, and $std_{t_1:t_2} = \sqrt{VaR_{t_1:t_2}}$.
- Portfolio IR, defined as the ratio of continuously compounded portfolio returns to the standard deviation of these returns: $IR = \frac{ret}{std}$.
- Since transaction costs are a non-trivial aspect that investors must consider, we calculate the portfolio returns net of proportional transaction costs (TC) as:

$$\tilde{\boldsymbol{r}} = (1 + \boldsymbol{R}_{t+1}^{\top}\boldsymbol{w}_t)(1 - \kappa \sum_{j=1}^{N} |w_{j,t+1} - w_{j,t}|) - 1, \quad (21)$$

where $\boldsymbol{R}_{t+1}$ is the vector of asset returns for period $t+1$, $w_{j,t}$ is the estimated portfolio weight of asset j at $t+1$, and κ is the proportional transaction costs. For daily observations, we assess post-transaction costs portfolio performance for $\kappa \in \{2, 5, 10\}$ basis points (bps).

4.2. M6 competition dataset: Empirical performance

Table 2 presents the out-of-sample performance, specifically the continuously compounded portfolio returns (Returns), the standard deviation (Risk), and the information ratio (IR) across the 12 submission points for our proposed APO framework, the M6 benchmark, and the average performance of the M6 competition teams before considering transaction costs. The continuously compounded portfolio return and standard deviation are 9.5% and 1.9%, respectively. This translates into an IR of 5.045, significantly surpassing both the M6 competition teams (−3.421) and the benchmark (0.453) in terms of IR. Only 16 teams in the M6 competition (11%) reported an IR higher than 6.411. The benchmark and mean of the M6 competition teams consistently exhibit lower risk than the APO in each period. Therefore, the overall outperformance of the APO is primarily attributed to its higher returns.

Table 1
Overview of benchmark allocation strategies.

Strategy	Optimization function	Required parameters
Volatility Timing	$w_i = \dfrac{1/\sigma_i^2}{\sum_{i=1}^{N}(1/\sigma_i^2)}$	Variances of asset returns
Minimum-variance	$\min VaR = \boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w}$	Sample covariance matrix
Mean-variance	$\max U = \boldsymbol{\mu}^\top \boldsymbol{w} - \dfrac{\delta}{2} \boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w}$	Risk-aversion level, covariance matrix and return estimates for all assets
Black-Litterman	$\max SR = \frac{\boldsymbol{\mu}_{BL}{}^\top \boldsymbol{w} - r_f}{\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_{BL} \boldsymbol{w}}}$ with: $\boldsymbol{\pi}_{BL} = \delta\, \boldsymbol{\Sigma} \boldsymbol{w}^{ref}$ $\boldsymbol{\mu}_{BL} = [(\tau\boldsymbol{\Sigma})^{-1} + \boldsymbol{P}^\top \boldsymbol{\Omega}^{-1} \boldsymbol{P}]^{-1} [(\tau\boldsymbol{\Sigma})^{-1} \boldsymbol{\pi}_{BL} + \boldsymbol{P}^\top \boldsymbol{\Omega}^{-1} \boldsymbol{V}]$, $\boldsymbol{\Sigma}_{BL} = \boldsymbol{\Sigma} + [(\tau\boldsymbol{\Sigma})^{-1} + \boldsymbol{P}^\top \boldsymbol{\Omega}^{-1} \boldsymbol{P}]^{-1}$	Risk-aversion, covariance matrix, views, reference portfolio, risk-free rate
Bayes-stein Shrinkage	$\max SR = \frac{\boldsymbol{\mu}_{BS}{}^\top \boldsymbol{w} - r_f}{\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_{BS} \boldsymbol{w}}}$ with: $g_{mv} = \dfrac{\mathrm{N}+2}{(\mathrm{N}+2) + \mathrm{M}(\boldsymbol{\mu}_{ml} - \mu_g \mathbf{1})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_{ml} - \mu_g \mathbf{1})}$ $\boldsymbol{\mu}_{BS} = (1 - g_{mv})\, \boldsymbol{\mu}_{ml} + g_{mv} \mu_g \mathbf{1}$ $\tilde{\upsilon} = \dfrac{M+2}{(\boldsymbol{\mu}_{ml} - \mu_g \mathbf{1})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_{ml} - \mu_g \mathbf{1})}$ $\boldsymbol{\Sigma}_{BS} = \boldsymbol{\Sigma} \left(\dfrac{M + \tilde{\upsilon} + 1}{M + \tilde{\upsilon}} \right) + \dfrac{\tilde{\upsilon}}{M(M + \tilde{\upsilon} + 1)} \dfrac{\mathbf{1}\mathbf{1}^\top}{\mathbf{1}^\top \boldsymbol{\Sigma}^{-1} \mathbf{1}}$	Risk-free rate, covariance matrix, length of estimation window, maximum-likelihood estimation for returns, global mean return
with: w_i=portfolio weight of asset i ($\boldsymbol{w}$=vector of portfolio weights)		r_f=risk-free rate
$\hat{\sigma}_i^2$=volatility estimate of asset i		$\boldsymbol{V}$= investors' views
U=utility function of investor		δ=risk version level
$\boldsymbol{\mu}$=vector of return estimates		$\boldsymbol{\Omega}$=reliability of views
$\boldsymbol{\pi}$=vector of implied return estimates		g_{mv}=shrinkage factor
$\boldsymbol{w}^{ref}$=reference portfolio weight		M= estimation window
τ= reliability measure of equilibrium expected returns		μ_g= global mean return
$\boldsymbol{\mu}_{ml}$=vector of maximum-likelihood estimation for returns		$\tilde{\upsilon}$=prior precision
$\boldsymbol{P}$ =unity matrix (in case of an absolute return estimate for each asset		$\boldsymbol{\Sigma}$=sample covariance matrix

This table summarizes details of the benchmark strategies employed in this paper. We report from left to right the model, its optimization functions for obtaining the optimal portfolio weights, and parameters required for implementing the optimization framework.

4.3. S&P500 datasets: Empirical performance

Table 3 presents the characteristics of portfolios of N stocks from those in the S&P500. We consider N values of 25, 50, and 100 and compare the performance of the APO against other portfolio selection methodologies. Across all portfolio sizes, APO shows favorable characteristics compared to different approaches. For instance, when $N = 50$, the APO achieves continuously compounded returns before transaction costs of 11.2%, outperforming other strategies such as 1/N (1.9%), VT (2.1%), MinVar (1.4%), MeanVar (2.8%), BS (4.1%), and BL (3%). Across all N levels, the APO consistently yields the highest returns. However, it exhibits the highest standard deviation, particularly noticeable for $N = 25$. Regarding the Information Ratio (IR), the APO maintains a clear advantage across all N sizes. Without transaction costs, the APO demonstrates impressive IRs of 3.104, 5.125, and 3.634 for N values of 25, 50, and 100, respectively. Notably, for $N = 50$, the APO outperforms the leading benchmark strategy by a significant margin. The 1/N strategy also outperforms the sample-based mean–variance strategy and its variations. Apart from APO, only VT and BS consistently achieve positive IRs across all datasets, yet none can consistently surpass the performance of 1/N. These findings confirm those of DeMiguel et al. (2009). Overall, we show that ML-driven automated and dynamic model selection and hyperparameter tuning provide strong statistical evidence in favor of portfolio optimization.

Table 2
Empirical performance of the M6 Dataset.

Period	APO			Benchmark			M6 competition teams		
	Returns	Risk	IR	Returns	Risk	IR	Returns	Risk	IR
1st Sub	0.005	0.013	0.429	0.044	0.011	3.990	0.015	0.010	1.285
2nd Sub	−0.049	0.019	−2.582	−0.063	0.010	−5.972	−0.028	0.008	−2.957
3rd Sub	0.115	0.022	5.165	0.018	0.015	1.215	0.006	0.011	0.649
4th Sub	−0.169	0.029	−5.926	−0.063	0.015	−4.139	−0.029	0.011	−2.186
5th Sub	−0.019	0.018	−1.025	0.005	0.009	0.577	−0.003	0.007	−0.361
6th Sub	0.116	0.018	6.543	0.051	0.008	6.060	0.019	0.007	2.658
7th Sub	0.008	0.016	0.471	−0.064	0.012	−5.273	−0.022	0.008	−1.891
8th Sub	0.026	0.026	0.971	−0.073	0.015	−4.834	−0.019	0.010	−1.020
9th Sub	0.124	0.016	7.540	0.110	0.014	7.839	0.028	0.010	2.223
10th Sub	−0.098	0.012	−8.161	0.000	0.008	−0.017	−0.004	0.006	−0.529
11th Sub	0.054	0.015	3.637	0.006	0.011	0.570	0.001	0.008	−0.015
12th Sub	−0.017	0.015	−1.166	0.034	0.007	5.122	0.005	0.006	0.021
Global	0.095	0.019	5.045	0.005	0.012	0.453	−0.031	0.009	−3.421

This table summarizes the performance of the APO strategy, the benchmark, and the average performance of the M6 competition teams with zero transaction costs across the 12 submission points. It reports the return, risk, and information ratio (IR) generated using the M6 competition data.

Table 3
Empirical performance of the S&P500 Dataset - TCs=0 bps.

Strategy	$N = 25$			$N = 50$			$N = 100$		
	Returns	Risk	IR	Returns	Risk	IR	Returns	Risk	IR
1/N	0.038	0.016	2.402	0.019	0.015	1.221	0.022	0.015	1.482
VT	0.037	0.013	2.773	0.021	0.013	1.687	0.011	0.013	0.893
MinVar	0.041	0.014	2.915	0.014	0.014	1.009	−0.033	0.011	−3.045
MeanVar	−0.042	0.026	−1.610	0.028	0.021	1.304	0.031	0.023	1.352
BS	0.046	0.021	2.230	0.041	0.015	2.656	0.023	0.016	1.475
BL	0.043	0.014	2.994	0.030	0.014	2.115	−0.090	0.015	−6.096
APO	0.076	0.024	3.104	0.112	0.022	5.125	0.080	0.022	3.634

This table summarizes the performance of the 1/N, the volatility-timing (VT), the minimum-variance (MinVar), the mean–variance (MeanVar), the Bayes-Stein (BS), the Black-Litterman (BL) and the APO strategy with zero transaction costs. It reports the return, risk, and information ratio (IR) generated using the S&P500 dataset for $N = 25, 50, 100$ assets.

5. Additional analysis

In the previous sections, we highlighted the benefit of our proposed method over traditional 'estimate-then-optimize' approaches. In this section, we conduct additional empirical analysis to explore the sources of profitability of the APO method. Specifically, we assess the performance advantage of APO by comparing it with two benchmarks: (1) APO methods using more conventional optimization approaches and (2) traditional mean–variance models incorporating similar shrinkage techniques. Additionally, we examine the performance of APO when proportional transaction costs = 2, 5, 10 basis points.

5.1. Values of the use of optimizers: MMA vs. ECOS/SCS

A key advantage of the APO method lies in the superior optimization capabilities of the Method of Moving Asymptotes (MMA) compared to standard solvers. To explore this advantage, we compare MMA's performance with that of two well-established solvers in the optimization field: the Splitting Conic Solver (SCS) and the Embedded Conic Solver (ECOS) for the M6 dataset. SCS uses an alternating direction method of multipliers (ADMM) approach, which is particularly effective for handling large-scale, sparse problems with complex constraints. On the other hand, ECOS is based on the interior-point method, is known for its efficiency in handling smaller to medium-sized problems, and offers a balance between performance and ease of use. Both optimizers are implemented in popular programming environments and are integrated with widely used software libraries, making them accessible and convenient for benchmarking. As shown in Table 4, MMA consistently outperforms both ECOS and SCS. Specifically, MMA improves the information ratio (IR) across eight submission periods and reduces risk in 9 submission periods. MMA achieves a global IR of 5.045, compared to 2.049 for ECOS and 2.132 for SCS. This demonstrates that MMA is more effective in solving the optimization problem and less likely to be trapped at local optima than more 'off-the-shelf' optimizers like ECOS and SCS.

5.2. Values of the use of shrinkage

The APO method's second source of performance advantage is using $l1$ and $l2$ shrinkage. One would need to carefully apply shrinkage to the covariance matrix and return forecasts to achieve benefits similar to the traditional mean–variance framework. Thus, we compare APO with portfolios constructed with the shrinkage covariance estimator of Ledoit and Wolf (2003), which propose shrinking

Table 4
M6 performance of MMA vs. ECOS/SCS.

Period	MMA			ECOS			SCS		
	Returns	Risk	IR	Returns	Risk	IR	Returns	Risk	IR
1st Sub	0.005	0.013	0.429	0.043	0.011	4.026	0.043	0.011	4.026
2nd Sub	−0.049	0.019	−2.582	−0.052	0.027	−1.882	−0.052	0.027	−1.881
3rd Sub	0.115	0.022	5.165	0.187	0.032	5.845	0.187	0.032	5.845
4th Sub	−0.169	0.029	−5.926	−0.232	0.039	−5.953	−0.232	0.039	−5.953
5th Sub	−0.019	0.018	−1.025	−0.013	0.011	−1.257	−0.013	0.011	−1.257
6th Sub	0.116	0.018	6.543	0.105	0.019	5.650	0.105	0.019	5.655
7th Sub	0.008	0.016	0.471	−0.061	0.012	−5.198	−0.061	0.012	−5.198
8th Sub	0.026	0.026	0.971	0.058	0.032	1.820	0.058	0.032	1.818
9th Sub	0.124	0.016	7.540	0.129	0.020	6.399	0.129	0.020	6.399
10th Sub	−0.098	0.012	−8.161	−0.192	0.018	−10.947	−0.190	0.017	−10.915
11th Sub	0.054	0.015	3.637	0.057	0.017	3.420	0.057	0.017	3.418
12th Sub	−0.017	0.015	−1.166	0.016	0.006	2.641	0.016	0.006	2.641
Global	0.095	0.019	5.045	0.046	0.022	2.049	0.047	0.022	2.132

This table compares the performance of the proposed APO strategy with MMA against alternative optimizers (ECOS and SCS). It reports the return, risk, and information ratio (IR) generated using the M6 competition data across the 12 submission points. We consider zero transaction costs.

the sample covariance matrix $\boldsymbol{\Sigma}$ towards a diagonal matrix $\boldsymbol{H}$, i.e., $\boldsymbol{\Sigma}_{lw} = \zeta\boldsymbol{H} + (1-\zeta)\boldsymbol{\Sigma}$, where ζ is the shrinkage intensity, $0 \leq \zeta \leq 1$. $\boldsymbol{H}$ is derived by scaling the identity matrix $\boldsymbol{I}$ by the average variance of the assets to minimize the bias of the shrinkage target, i.e., $\boldsymbol{H} = \overline{\sigma^2}\boldsymbol{I}$. The authors choose the shrinkage intensity ζ to minimize the expected quadratic loss of the shrinkage estimator. This is done using the Frobenius norm of the difference between the shrinkage estimator and the true covariance matrix $\boldsymbol{S}$:

$$L(\zeta) = E[\|\zeta\boldsymbol{H} + (1-\zeta)\boldsymbol{\Sigma} - \boldsymbol{S}\|^2], \tag{22}$$

where for a symmetric matrix $\hat{\boldsymbol{Z}}$, $\|\hat{\boldsymbol{Z}}\|^2 = Trace(\hat{\boldsymbol{Z}}^2)$.

We also consider portfolios with both the shrinkage estimator of the mean and shrinkage covariance matrix, i.e., we replace the sample covariance matrix $\boldsymbol{\Sigma}$ in the BL and BS optimization framework with the Ledoit-Wolf shrinkage covariance matrix $\boldsymbol{\Sigma}_{lw}$. Based on the results from Table 5, an interesting finding is that incorporating some form of shrinkage on the covariance reduces the effect of estimation error. This can be seen, for instance, by examining the MeanVaR strategy's IR, which increases from 1.352 to 2.580 when applying shrinkage towards a scaled identify matrix - almost doubling the performance ratio. However, while traditional mean–variance models benefit from using shrinkage estimators for asset return moment, they can only consistently outperform the 1/N. In contrast, APO consistently exceeds the 1/N benchmark and outperforms all competing strategies except the BL-lw approach when $N = 25$. This underscores the superior modeling capabilities of APO.

5.3. Incorporating transaction costs

In the context of stock return predictability, research by Huck (2019), Avramov et al. (2023), among others, has shown that ML-based performance often deteriorates in the presence of reasonable trading costs due to high turnover and extreme positions. To determine whether our results remain robust with transaction costs, we examine proportional transaction costs (TC) of 2, 5, and 10 basis points and assess the portfolio returns net of these costs across all benchmark strategies and datasets. The APO strategy consistently achieves significantly higher IRs than the benchmark strategies at all three levels of TC. As anticipated, the effect of transaction costs on portfolio returns is minimal. For example, with a portfolio size of 100 and a TC of 10 basis points, the IR of the APO drops only slightly from 3.634 to 3.267. Therefore, even with a ten basis point transaction cost, the performance of the APO strategy remains largely unaffected (see Tables 6–8).

6. Conclusions

The traditional method for portfolio selection involves estimating portfolio parameters and then optimizing asset weights based on these estimates. Our study proposes a new approach that optimizes portfolio objectives through automated machine learning, rendering portfolio selection a constrained, penalized regression problem. The penalization term addresses overfitting and helps to reduce transaction costs. We expanded the search space to include algorithms and hyperparameter selection and used moving asymptotes to determine the optimal weights. Our intuition is that the complexity of financial data, characterized by noise and non-stationary dynamics, presents significant challenges for investors trying to capture asset dependencies. This complexity makes it difficult to accurately approximate the underlying process of asset returns and exploit mispriced assets, thus limiting opportunities for consistent outperformance over the market through active trading or stock selection. Compared to methods that predict asset rank probabilities and then use these predictions for investment decisions, the proposed APO approach with MMA demonstrates superior performance. It achieves a higher information ratio than both the M6 competition benchmark and the average performance of the competition teams. We further explore the sources of APO's performance advantage. Extensive robustness experiments confirm the effectiveness of APO's modeling and optimization methods.

Table 5
Benchmark Performance with Ledoit-Wolf shrinkage covariance vs. APO.

Strategy	N=25			N=50			N=100		
	Returns	Risk	IR	Returns	Risk	IR	Returns	Risk	IR
1/N	0.038	0.016	2.402	0.019	0.015	1.221	0.022	0.015	1.482
VT-lw	0.037	0.014	2.694	0.021	0.013	1.662	0.011	0.013	0.872
MinVar-lw	0.041	0.014	2.876	0.014	0.014	1.003	−0.032	0.011	−2.898
MeanVar-lw	−0.024	0.026	−0.903	0.020	0.022	0.941	0.058	0.023	2.580
BS-lw	0.046	0.021	2.249	0.040	0.015	2.650	0.022	0.015	1.465
BL-lw	0.058	0.015	3.777	0.030	0.014	2.119	−0.055	0.016	−3.464
APO	0.076	0.024	3.104	0.112	0.022	5.125	0.080	0.022	3.634

This table summarizes the performance of the 1/N, the volatility-timing (VT-lw), the minimum-variance (MinVar-lw), the mean–variance (MeanVar-lw), the Bayes-Stein (BS-lw), the Black-Litterman (BL-lw) and the APO strategy with zero transaction costs. It reports the return, risk, and information ratio (IR) generated using the S&P500 dataset for $N = 25, 50, 100$ assets.

Table 6
Empirical performance of the S&P500 Dataset - TCs=2 bps.

Strategy	$N = 25$			$N = 50$			$N = 100$		
	Returns	Risk	IR	Returns	Risk	IR	Returns	Risk	IR
1/N	0.037	0.016	2.349	0.018	0.015	1.166	0.021	0.015	1.426
VT	0.037	0.013	2.714	0.020	0.013	1.623	0.011	0.013	0.830
MinVar	0.041	0.014	2.855	0.013	0.014	0.949	−0.034	0.011	−3.152
MeanVar	−0.043	0.026	−1.664	0.025	0.021	1.189	0.030	0.023	1.287
BS	0.045	0.021	2.157	0.039	0.015	2.550	0.022	0.016	1.389
BL	0.042	0.014	2.920	0.029	0.014	2.046	−0.093	0.015	−6.247
APO	0.074	0.024	3.041	0.110	0.022	5.015	0.078	0.022	3.561

This table summarizes the performance of the 1/N, the volatility-timing (VT), the minimum-variance (MinVar), the mean–variance (MeanVar), the Bayes-Stein (BS), the Black-Litterman (BL) and the APO strategy with proportional transaction costs of 2 basis points. It reports the return, risk, and information ratio (IR) generated using the S&P500 dataset for $N = 25, 50, 100$ assets.

Table 7
Empirical performance of the S&P500 Dataset - TCs=5 bps.

Strategy	$N = 25$			$N = 50$			$N = 100$		
	Returns	Risk	IR	Returns	Risk	IR	Returns	Risk	IR
1/N	0.036	0.016	2.270	0.017	0.015	1.083	0.020	0.015	1.342
VT	0.035	0.013	2.625	0.019	0.013	1.526	0.009	0.013	0.736
MinVar	0.039	0.014	2.766	0.012	0.014	0.860	−0.036	0.011	−3.314
MeanVar	−0.046	0.026	−1.745	0.022	0.021	1.017	0.028	0.023	1.190
BS	0.043	0.021	2.047	0.037	0.015	2.391	0.020	0.016	1.260
BL	0.041	0.014	2.811	0.028	0.014	1.943	−0.096	0.015	−6.472
APO	0.072	0.024	2.946	0.106	0.022	4.850	0.076	0.022	3.450

This table summarizes the performance of the 1/N, the volatility-timing (VT), the minimum-variance (MinVar), the mean–variance (MeanVar), the Bayes-Stein (BS), the Black-Litterman (BL) and the APO strategy with proportional transaction costs of 5 basis points. It reports the return, risk, and information ratio (IR) generated using the S&P500 dataset for $N = 25, 50, 100$ assets.

Table 8
Empirical performance of the S&P500 Dataset - TCs=10 bps.

Strategy	$N = 25$			$N = 50$			$N = 100$		
	Returns	Risk	IR	Returns	Risk	IR	Returns	Risk	IR
1/N	0.034	0.016	2.137	0.015	0.015	0.945	0.018	0.015	1.202
VT	0.033	0.013	2.478	0.017	0.013	1.365	0.007	0.013	0.580
MinVar	0.037	0.014	2.617	0.010	0.014	0.711	−0.039	0.011	−3.583
MeanVar	−0.049	0.026	−1.880	0.016	0.021	0.729	0.024	0.023	1.027
BS	0.039	0.021	1.863	0.033	0.015	2.125	0.017	0.016	1.045
BL	0.038	0.014	2.628	0.025	0.014	1.772	−0.102	0.015	−6.847
APO	0.068	0.024	2.788	0.100	0.022	4.576	0.072	0.022	3.267

This table summarizes the performance of the 1/N, the volatility-timing (VT), the minimum-variance (MinVar), the mean–variance (MeanVar), the Bayes-Stein (BS), the Black-Litterman (BL) and the APO strategy with proportional transaction costs of 10 basis points. It reports the return, risk, and information ratio (IR) generated using the S&P500 dataset for $N = 25, 50, 100$ assets.

CRediT authorship contribution statement

Xinyu Huang: Writing – original draft, Software, Investigation, Formal analysis, Data curation. **David P. Newton:** Writing – review & editing. **Emmanouil Platanakis:** Supervision, Project administration, Methodology, Conceptualization. **Charles Sutcliffe:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data and code availability

The codes for reproducing the numerical results in this manuscript are openly shared on GitHub at the following link: https://github.com/xh646-alt/M6_APO_codes/. The primary dataset used in our study is not publicly available due to the API limitations.

References

Ai, H., Liu, C., & Lin, P. (2024). Robust returns ranking prediction and portfolio optimization for M6. *International Journal of Forecasting*.

Avramov, D., Cheng, S., & Metzker, L. (2023). Machine learning vs. economic restrictions: Evidence from stock return predictability. *Management Science*, *69*(5), 2587–2619.

Bali, T. G., Beckmeyer, H., Moerke, M., & Weigert, F. (2023). Option return predictability with machine learning and big data. *The Review of Financial Studies*, *36*(9), 3548–3602.

Best, M. J., & Grauer, R. R. (1991). On the sensitivity of mean-variance-efficient portfolios to changes in asset means: Some analytical and computational results. *The Review of Financial Studies*, *4*(2), 315–342.

Bianchi, D., Büchner, M., & Tamoni, A. (2021). Bond risk premiums with machine learning. *The Review of Financial Studies*, *34*(2), 1046–1089.

Black, F., & Litterman, R. (1992). Global portfolio optimization. *Financial Analysts Journal*, *48*(5), 28–43.

Board, J. L., & Sutcliffe, C. M. (1994). Estimation methods in portfolio selection and the effectiveness of short sales restrictions: UK evidence. *Management Science*, *40*(4), 516–534.

Brandt, M. W. (1999). Estimating portfolio and consumption choice: A conditional Euler equations approach. *The Journal of Finance*, *54*(5), 1609–1645.

Brodie, J., Daubechies, I., De Mol, C., Giannone, D., & Loris, I. (2009). Sparse and stable Markowitz portfolios. *Proceedings of the National Academy of Sciences*, *106*(30), 12267–12272.

Cakici, N., Fieberg, C., Metko, D., & Zaremba, A. (2023). Machine learning goes global: Cross-sectional return predictability in international stock markets. *Journal of Economic Dynamics & Control*, *155*, Article 104725.

Cong, L. W., Tang, K., Wang, J., & Zhang, Y. (2020). AlphaPortfolio: Direct construction through reinforcement learning and interpretable AI. *Social Science Research Network*, *3554486*.

DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *The Review of Financial Studies*, *22*(5), 1915–1953.

DeMiguel, V., Gil-Bazo, J., Nogales, F. J., & Santos, A. A. (2023). Machine learning and fund characteristics help to select mutual funds with positive alpha. *Journal of Financial Economics*, *150*(3), Article 103737.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669.

Grauer, R. R., & Shen, F. C. (2000). Do constraints improve portfolio performance? *Journal of Banking & Finance*, *24*(8), 1253–1274.

Green, R. C., & Hollifield, B. (1992). When will mean-variance efficient portfolios be well diversified? *The Journal of Finance*, *47*(5), 1785–1809.

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

Harris, R. D., Stoja, E., & Tan, L. (2017). The dynamic Black–Litterman approach to asset allocation. *European Journal of Operational Research*, *259*(3), 1085–1096.

Harvey, C. R., Liu, Y., & Zhu, H. (2016). . . . And the cross-section of expected returns. *The Review of Financial Studies*, *29*(1), 5–68.

Huck, N. (2019). Large data sets and machine learning: Applications to statistical arbitrage. *European Journal of Operational Research*, *278*(1), 330–342.

Jorion, P. (1986). Bayes-Stein estimation for portfolio analysis. *Journal of Financial and Quantitative analysis*, *21*(3), 279–292.

Kinn, D. (2018). Reducing estimation risk in mean-variance portfolios with machine learning. arXiv preprint arXiv:1804.01764.

Kirby, C., & Ostdiek, B. (2012). It's all in the timing: Simple active portfolio strategies that outperform naive diversification. *Journal of Financial and Quantitative Analysis*, *47*(2), 437–467.

Ledoit, O., & Wolf, M. (2003). Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, *10*(5), 603–621.

Makridakis, S., Spiliotis, E., Hollyman, R., Petropoulos, F., Swanson, N., & Gaba, A. (2023). The M6 forecasting competition: Bridging the gap between forecasting and investment decisions. arXiv preprint arXiv:2310.13357.

Markowitz, H. (1952). The utility of wealth. *Journal of Political Economy*, *60*(2), 151–158.

Platanakis, E., Sutcliffe, C., & Ye, X. (2021). Horses for courses: Mean-variance for asset allocation and 1/N for stock selection. *European Journal of Operational Research*, *288*(1), 302–317.

Platanakis, E., & Urquhart, A. (2019). Portfolio management with cryptocurrencies: The role of estimation risk. *Economics Letters*, *177*, 76–80.

Svanberg, K. (1987). The method of moving asymptotes—A new method for structural optimization. *International Journal for Numerical Methods in Engineering*, *24*(2), 359–373.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *58*(1), 267–288.

Wu, W., Chen, J., Yang, Z., & Tindall, M. L. (2021). A cross-sectional machine learning approach for hedge fund return prediction and selection. *Management Science*, *67*(7), 4577–4601.