LLM-Enhanced Portfolio Construction

Nathan Sun[*] Jonathan Y. Tan[†]

December 2025

Abstract

Large Language Models have shown strong potential to handle financial tasks. In this work, we proposed a two-stage framework that uses the power of LLM to help with portfolio construction. In the first stage, we built an AI agent to screen the universe of assets and select a subset of assets that are valuable to invest. In the second stage, we use quantitative methods to estimate the precision matrix on the selected assets and use it to construct the portfolio. Experiments on S&P 500 stocks show that our framework returns portfolios with high Sharpe ratio under all the precision matrix estimation methods and the weight optimization methods, which outperform the baseline that constructs portfolios without stock screening.

1 Introduction

Portfolio construction in the stock market is traditionally very challenging due to the uncertainty and nonstationary nature of market data. A large body of literature has focused on studying how to select optimal weights among a set of stocks. However, most traditional approaches assume that the candidate asset universe is fixed, and place little emphasis on the problem of how to select a high-quality subset of investable stocks in the first place. In practice, maintaining a large and unfiltered stock universe often leads to the inclusion of many low-quality assets with poor performance. Including them in the portfolio optimization process will introduce noise and lead to a bad portfolio.

Maintaining a large universe of investable assets is not only economically inefficient but also statistically difficult. One main challenge in portfolio construction is to estimate the precision matrix, which is the inverse of the covariance matrix of the candidate stocks. Caner and Fan (2025) mentioned that as the number of stocks increases, the precision matrix becomes sparse, which makes accurate estimation more difficult. Although various sparse estimation and sparse regression methods have been proposed (see, e.g., Meinshausen and Bühlmann (2006)), they still suffer when the number of assets is large compared to the length of the time window. Therefore, intelligent stock screening is a crucial step to find a high-performance portfolio.

[*]Columbia University. Email: ns3903@columbia.edu
[†]Columbia University. Email: jyt2123@columbia.edu

To solve these problems, we develop a two-stage framework for portfolio construction. In the first stage, we design an AI agent for stock screening. The agent is able to read and analyze market data and generate a buy, hold, and sell rule for a given date. Then we use the screening rule to select a subset of stocks for further optimization. These signals are used to filter the investment universe and select a subset of candidate stocks for further optimization. In the second stage, we apply quantitative portfolio optimization techniques to the selected assets. Specifically, we use the historical return of the selected stocks to estimate the precision matrix. Based on the estimated precision matrix, we compute the optimal weights for each stock and construct the final portfolio.

We use experiments on a real world dataset to validate our portfolio construction method. Using market data for S&P 500 stocks at the end of each year from 2019 to 2023, we run the AI agent to select a subset of candidate stocks for the next year. Then we used different quantitative methods to construct portfolios from this subset every month. We compare the Sharpe ratios of these portfolios with the baseline where we do not use the screening agent but directly construct the portfolio on all S&P 500 stocks. Our experiments show that under the same precision matrix estimation methods and weight optimization methods, doing stock screening always has better performance.

The following Figure 1 shows how our framework works. Given a universe of stocks S and the stock dataset, the AI agent interacts with the dataset and selects a subset of stocks S' as the candidates of the portfolio. Then we use the corresponding stock data to estimate the precision matrix and compute the optimal weights to obtain the final portfolio P to invest.

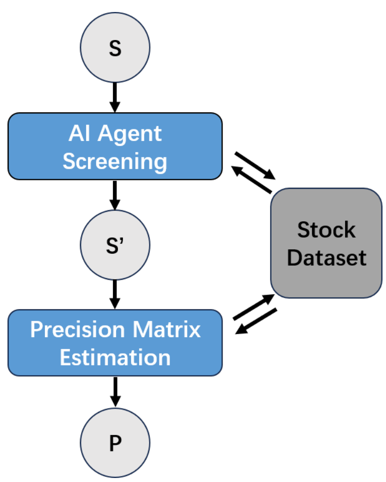

Figure 1: Two-Stage Portfolio Construction

2 Related Work

Much previous work has been done to show that ML-based techniques are able to consistently achieve high out-of-sample performances. Gu et al. (2020) and Kelly et al. (2025) have applied a variety of ML frameworks, including neural networks, decision trees, and transformers, to show large economic gains compared to traditional regression-based methods. Stock selection is important too: Brandt et al. (2009) devised a model that parametrized each stock's weight in a portfolio as a direct function of firm characteristics. In the process, their model contains a component similar to a ReLU activation function, which allows them to zero out firm completely from a portfolio. Their work shows substantial improvement upon the equal- and value-weighted benchmarks.

Our model is broadly related to two strains of literature: using Agentic AI for portfolio optimization, and precision matrix estimation methods. Zhao et al. (2025) built a multi-agent system composed of LLMs for portfolio selection. The latter strain is much more developed: Callot et al. (2021), Caner et al. (2021), Caner and Daniele (2025), Ledoit and Wolf (2012), and Fan et al. (2013) are all developments in estimating the covariance/precision matrix in high dimensional environments.

In the following report, we are inspired by the recent rise of Agentic AI in portfolio choice. We construct a system with two agents: a fundamental-based LLM designed to screen for only high-quality stocks, and a mathematical/statistical agent that uses state-of-the-art precision matrix techniques to determine the optimal weights for each filtered stock. Intuitively, the argument for stock filtering is clear: rather than forcing the investor to potentially have a position in every stock (which current covariance matrix estimation techniques do), if we filtered out low-quality stocks, we remove underperforming stocks from the portfolio, leading to higher out-of-sample performances.

3 Model

3.1 Agent Design

AI agents have shown great potential to deal with financial tasks. Zhao et al. (2025) developed a structure that enables multiple AI agents to get information from different sources and communicate each other to make equity portfolio decisions. Saha et al. (2025) provided foundations for LLM agents in investment management. In this subsection, we introduce the way to build a stock screening agent that can use firm data to generate stock screening rules. We use Google Gemini-2.5-flash as the core reasoning engine and use CrewAI as the framework to build the AI agent.

First, we provide several tools (functions) that enable the AI agent to interact with the dataset. By calling the tools, the AI agent can directly query the dataset, get the schema of each firm's data, identify firms with extreme characteristic values, and compute the statistics of firm characteristics. In this way we do not have to directly put the whole dataset as an input of the prompt, which significantly reduces the computational and memory cost.

Second, we provide a detailed guideline for the AI agent to analyze data and make the final screening rule. In this step, we explain the meaning of the rows and columns of the dataset to the agent, and describe how to explore the dataset. Then we ask the agent to come up with economic intuitions from its pretrained knowledge, and the derive the screening rule from these intuitions. We also restrict the output rule to have an explicit form which helps for further explanation. To help the AI agent understand its responsibilities, we provided several examples for each step.

Third, we define a concrete task that the agent must solve. To leverage the zero-shot capabilities of LLM models, we only provide firm characteristics at the end of each year, and ask it to output buy, hold, and sell signals for the next year. Importantly, the agent does not use statistical learning on historical data. Instead, it relies on pretrained domain knowledge and in-context reasoning to analyze the current cross-section of firms and produce screening decisions. This design fundamentally differs from traditional learning-based stock screening methods, which depend on explicit model training and future return estimation over historical return data.

3.2 Precision Matrix Estimation

Given a subset of stocks of size N, let Y be the $N \times T$ matrix of returns of all stocks over a time window T. We apply three main techniques to estimate the $N \times N$ precision matrix $\hat{\Sigma}^{-1}$: naive nodewise regression (Meinshausen and Bühlmann (2006), Callot et al. (2021)), residual nodewise regression (Caner et al. (2021)), and deep learning (Caner and Daniele (2025)). An in-depth explanation of all the above methods can be found in Fan and Caner (2025). However, these methods have one thing in common: their estimate of the precision matrix achieves consistency to the true precision matrix, even in high-dimensional environments when $N >> T$.

Note that in our stock selection environment, we are frequently operating in a high-dimensional environment where $N >> T$. In this case, estimating the covariance matrix $\hat{\Sigma}$ using standard statistical techniques and inverting it may lead to huge numerical instability: even if $\hat{\Sigma}$ may be numerically nonsingular, a small error will be magnified in $\hat{\Sigma}^{-1}$. Hence it is important to use precision matrix techniques when constructing $\hat{\Sigma}^{-1}$, as it has been proven to give a consistency guarantee in high-dimensional environments.

4 Results

Our empirical setup is as follows. We obtain S&P500 monthly fundamental data from CRSP/Compustat. Specifically, for each firm, we obtain their log size, log book-to-market, and 12-month momentum characteristics. Our test period is from January 2020-April 2024.

We first use the AI agent to select candidate stocks for each year. Given all the S&P 500 stocks, the AI agent is able to see their normalized firm characteristics at the end of the previous year. The agent will follow our guidelines and provide a screening rule for all the stocks. The screening rules are thresholds for the firm characteristics or the combination of them. Then we can use the given rules to check the characteristics of every stock, and generate a buy, hold or sell signal for each of them. We will pick the stocks with buy or sell signals to the next step, since we will likely be able to make a profit by including the stocks with buy signals in the portfolio and short the stocks with sell signals.

Figure 2 shows an example output of the AI agent. We can see that the agent gives a concrete buy rule and sell rule for all the stocks. It also give some justifications of why it chooses such thresholds of firm characteristics. In this example, the agent tends to give buy signals to stocks with high book-to-market, large size, relatively high momentum, and sell signals to stocks with have opposite features.

````
Data Exploration Summary:
- The data includes standardized book-to-market (bm), momentum (mom12m), and market value of equity (mve) for S&P 500 firms a
s of December 2023.
- All features are standardized to have a mean of 0 and a standard deviation of 1.
- Extreme values were explored to understand the distribution of characteristics.

BUY RULE:
if bm > 0.95 AND mve > 0.3 AND mom12m > -0.5:
    signal = BUY

SELL RULE:
if (bm < -0.75 OR mom12m < -0.55 OR mve < -0.75) AND NOT (mom12m > 1.5):
    signal = SELL

HOLD RULE:
else:
    signal = HOLD

Rationale:
- BUY Rule: This rule targets undervalued (high bm), reasonably sized (mve > 0.3) companies with positive momentum (mom12m >
-0.5). The economic intuition is to buy companies that are currently cheap but have shown some signs of recovery or positive
market sentiment.
- SELL Rule: This rule aims to sell companies that are overvalued (low bm), have negative momentum (mom12m < -0.55), or are s
mall in size (mve < -0.75). The `NOT (mom12m > 1.5)` condition prevents selling companies with extremely high momentum, even
if they meet the other criteria, as these might be temporary situations or represent significant growth opportunities. The ec
onomic intuition is to avoid holding onto companies that are losing value or are too small to provide substantial returns, un
less they are exhibiting exceptional positive momentum.
- Expected signal distribution: The BUY rule is expected to generate signals for a small percentage of firms (around 2%), foc
using on higher-conviction value opportunities. The SELL rule is expected to affect a larger percentage of firms (around 5
0%), filtering out less desirable investments and managing risk. The remaining firms will be held.
```
````

Figure 2: Example output of the agent

In each test period, we use a 15-year historical rolling window to construct a series of returns Y for stocks that the LLM has filtered. From Y, we then construct the various precision matrix estimates $\hat{\Sigma}^{-1}_{NW}$, $\hat{\Sigma}^{-1}_{RNW}$, and $\hat{\Sigma}^{-1}_{DL}$, for naive nodewise, residual nodewise, and deep learning, respectively.

Then given a precision matrix $\hat{\Sigma}^{-1}$, we can calculate the global minimum variance (GMV) portfolio, the mean-variance (MV) optimal portfolio, and the maximum-Sharpe ratio (MSR) portfolio. Specifically, the mean-variance optimal weights are

$$w^* = \arg\min_{w \in \mathbb{R}^N} w'\Sigma w, \quad w'1_N = 1.$$

The mean-variance optimal portfolio targeting a monthly return of ρ is

$$w^* = \arg\min_{w \in \mathbb{R}^N} w'\Sigma w, \quad w'1_N = 1, w'\mu = \rho.$$

Lastly, the maximum Sharpe ratio optimal portfolio is

$$w^* = \arg\max_{w \in \mathbb{R}^N} w'\mu, \quad w'\Sigma w \leq \sigma^2.$$

For all the methods, we can estimate μ as the mean excess return in our 15-year window: $\hat{\mu}$. Hence we can estimate the variance of our portfolio as $\hat{\mu}\hat{\Sigma}^{-1}\hat{\mu}$.

In the following, we report the out-of-sample Sharpe ratio from 01/2020-04/2024 for all combinations of precision matrix estimation techniques and portfolio construction techniques. This
````

serves as a robustness check when comparing our model to a model that utilizes these precision matrix techniques without filtering for stocks first (henceforth referred to as the baseline model). In other words, the baseline models proxies for an investor who is knowledgable about these precision matrix techniques but is still forced a priori to have positions in all stocks in the S&P500. Table 1 presents the out-of-sample Sharpe ratios for the baseline model with no stock filtering, while Table 2 presents the out-of-sample Sharpe ratios with our model, where we first use the model described in Section 3.1 to choose *which stocks are optimal* to include, and using precision matrix techniques to determine *how much we should weight* each included stock.

	GMV	MV	MSR
NW	0.451	0.474	0.477
Residual NW	0.046	0.065	-0.103
Deep learning	-0.118	-0.033	-0.497

Table 1: Sharpe ratios with different methods of estimating the precision matrix, with different objective functions, applied to all firms in the S&P500. GMV=Global minimum variance portfolio, MV=mean-variance portfolio with target returns as 1% monthly, MSR=maximum Sharpe ratio portfolio.

	GMV	MV	MSR
NW	0.591	0.608	0.543
Residual NW	0.603	0.598	0.250
Deep learning	0.661	0.674	0.658

Table 2: Sharpe ratios with different methods of estimating the precision matrix, with different objective functions, applied to firms that the LLM has screened. GMV=Global minimum variance portfolio, MV=mean-variance portfolio with target returns as 1% monthly, MSR=maximum Sharpe ratio portfolio.

Notice that we are able to consistently able to achieve higher out-of-sample Sharpe ratios with LLM-guided filtering (in fact, our model outperforms the baseline for *every* precision matrix estimation/portfolio construction technique combination). Moreover, compared to the best baseline model, we are able to increase the out-of-sample Sharpe ratio from 0.477 to 0.674, over a 40% increase. This suggests that there is some financial value in LLM-based screening, as an investor is no longer forced to have positions in every stock in the market.

5 Conclusion

In this paper, we devised a multi-agent system used for constructing optimal, screened portfolios. The first agent, a fundamental-based LLM, chooses which stocks to buy/sell. The second agent, a statistical quantitative agent, uses the state-of-the-art precision matrix estimation techniques to obtain the optimal weights for each stock. We demonstrate that our model is consistently able to achieve higher performance than the baseline model (using only the precision estimation) for all combinations of precision matrix estimation techniques and portfolio construction techniques, improving the best model by up to 40%. Our work showcases the strength of optimally filtering for stocks before investing, and the power of LLMs (as generalizers) to accomplish that task.

However, there is much future work to do. First, FinBERT is another version of an LLM, trained specifically on financial text. Ensembling FinBERT with our current LLM provides a more sophisticated, and could potentially increase performance. On a related note, ensembling multiple LLMs together may work to reduce idiosyncratic error and hallucinations that have been well-documented in the LLM literature. Further, to emphasize how much our LLM is improving performance, we would also like to compare our performance to human analysts. Lastly, for more robustness checks, there are precision matrix estimation techniques like nonlinear shrinkage or POET (Thresholding Principal Orthogonal Complements) that we can implement to demonstrate conclusively that our method provides a significant out-of-sample improvement upon state-of-the-art portfolio construction techniques.

References

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

Callot, Laurent, Mehmet Caner, A Özlem Önder, Esra Ulaşan. 2021. A nodewise regression approach to estimating large portfolios. *Journal of Business & Economic Statistics* **39**(2) 520–531.

Caner, Mehmet, Maurizio Daniele. 2025. Deep learning based residuals in non-linear factor models: Precision matrix estimation of returns with low signal-to-noise ratio. *Journal of Econometrics* **251** 106083.

Caner, Mehmet, Qingliang Fan. 2025. A practitioner's guide to ai+ ml in portfolio investing. *Available at SSRN 5546200* .

Caner, Mehmet, Marcelo C Medeiros, Gabriel FR Vasconcelos. 2021. Residual based nodewise regression in factor models with ultra-high dimensions: Analysis of mean-variance portfolio efficiency and estimation of out-of-sample and constrained maximum sharpe ratios. Tech. rep., Texto para discussão.

Fan, Jianqing, Yuan Liao, Martina Mincheva. 2013. Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **75**(4) 603–680.

Fan, Qingliang, Mehmet Caner. 2025. A practitioner's guide to ai+ ml in portfolio investing. *arXiv preprint arXiv:2509.25456* .

Gu, Shihao, Bryan Kelly, Dacheng Xiu. 2020. Empirical asset pricing via machine learning. *The Review of Financial Studies* **33**(5) 2223–2273.

Kelly, Bryan T, Boris Kuznetsov, Semyon Malamud, Teng Andrea Xu. 2025. Artificial intelligence asset pricing models. Tech. rep., National Bureau of Economic Research.

Ledoit, Olivier, Michael Wolf. 2012. Nonlinear shrinkage estimation of large-dimensional covariance matrices .

Meinshausen, Nicolai, Peter Bühlmann. 2006. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics* .

Saha, Preetha, Jingrao Lyu, Arnav Saxena, Tianjiao Zhao, Dhagash Mehta. 2025. Large language model agents for investment management: Foundations, benchmarks, and research frontiers. *Benchmarks, and Research Frontiers (August 01, 2025)* .

Zhao, Tianjiao, Jingrao Lyu, Stokes Jones, Harrison Garber, Stefano Pasquali, Dhagash Mehta. 2025. Alphaagents: Large language model based multi-agents for equity portfolio constructions. *arXiv preprint arXiv:2508.11152* .