Prompting Large Language Models for Portfolio Optimization

Letizia Dimonopoli

May 18th, 2026

Abstract

This paper investigates whether Large Language Models are capable of constructing portfolios under diverse levels of increasing prompt guidance. We design an experiment where three models (GPT-4.1, GPT-5.4, and Claude Sonnet 4.6) have to allocate portfolios over the Dow Jones Industrial Average constituents. We find that an increase in prompt sophistication shifts the model reasoning towards formal optimization, however, it isn't able to achieve a fully efficient portfolio. The results highlight a gap between the ability of LLMs to explain their reasoning for optimization and their capacity to implement it.

Contents

1 Introduction

1.1 Motivation and Context

The recent advances in Large Language Models (LLMs) have stimulated an increasing interest in their potential applications in many fields, including in financial settings. In particular, LLMs have been proposed as tools for portfolio allocation, hoping that, through accurate prompts, they would produce sophisticated and plausible investment strategies. This would also mean more democratization in investment opportunities, as LLM-generated portfolios could also be created by non-expert users. However, there are doubts about whether LLMs can be effective substitutes for traditional quantitative methods.

Even though LLMs have the ability to process and generate natural language, and this could be combined with a broad exposure to financial concepts during training, portfolio construction is a quantitative task, typically framed as an optimization problem. In fact, the classical approaches, such as mean-variance optimization, rely on explicit mathematical formulations and numerical procedures, which balance expected return and risk. LLMs are not designed as numerical solvers and very often struggle to directly solve these optimization frameworks and computations.
This gap between LLMs' linguistic capability and their quantitative precision motivates us to closely examine of how LLMs behave when prompted to construct a portfolio.

1.2 Research Question(s)

The central question of this paper is whether increasing the level of optimization-related structure in the prompts improves the quality of the portfolios generated by the LLMs. Is the limitation of LLMs in portfolio construction due to a lack of economic intuition or a lack of mathematical structure? We design an experiment in which multiple LLMs are asked to construct portfolios over a fixed universe of assets, always using the same input data but varying the level of guidance embedded in the prompt. We use a range of guidance from minimal instructions to fully specified mean-variance optimization formulations.

We evaluate the resulting portfolios along five dimensions. We assess (1) the feasibility properties of the results, i.e. whether the models respected the constraints given to it. Then, we compute (2) their similarity to benchmark portfolios, which in our case are: the naive equally-weighted allocations, mean-variance and minimum variance portfolios. We also examine (3) some performance metrics, such as expected returns and volatility, the Sharpe

ratio and Herfindahl index, comparing them with the values from benchmark portfolios. Finally, we evaluate (4) the robustness of the results for each of the model and we perform (5) a semantic analysis of the LLM-generated comments.

2 Related Literature

2.1 Large Language Models in Finance

Recent work is exploring the role of LLMs in finance, specifically their use as robo-advisors for portfolio allocation, as for example in Huang et al. [1]. Moreover, studies such as Cao et al. [2] document how the collaboration "Man + Machine" can guide how people leverage AI solutions.

In particular, our paper can be seen as an extension to Guidolin et al. [3], whose contributions highlight that LLMs can produce economically meaningful allocations beyond the simplest diversification, but often fail to match the superior performance of traditional quantitative optimization strategies.

2.2 The Optimization Problem

This paper builds on Guidolin et al. [3]'s line of research by focusing on a slightly different dimension. We study LLMs' ability to approximate a well-defined portfolio optimization problem when provided with the necessary mathematical inputs. In particular, we investigate whether the way in which the quantitative concepts of optimization are given to the model through different levels of prompt design affect the quality of the resulting allocation.

3 Experimental Design

We address the research question by asking different LLMs to construct the portfolios under the same investment universe but varying the level of prompt detail.

3.1 Data: Dow Jones Industrial Average (DJIA)

The investment universe consists of the 30 constituents of the Dow Jones Industrial Average (DJIA). This choice reflects a trade-off between relevance and manageability. In fact, on the one hand, the DJIA contains large and well-known U.S. firms, which makes our experiment a realistic test case for portfolio allocation. On the other hand, its limited dimension keeps

the optimization problem manageable and helps contain API costs.

Thus, we rely on the Eikon Datastream DJ30 stocks as the data input to our prompts. In particular, we use weekly returns to estimate expected returns and covariance matrices and we fix an historical window between January 1st 2018 and June 30th 2024. While this window is shorter than the sample used in Guidolin et al. [3], this choice still ensures stable estimates and maintains consistency with the overall research design.

3.2 Models

We conduct the experiment using three LLMs: GPT 4.1, GPT 5.4 and Sonnet 4.6. As part of this group of models, we choose GPT 4.1, even though it is relatively old compared to the other two, to be able to follow the design of Guidolin et al. [3], who also used the same model. In general, the purpose of including multiple models is to reduce the risk that the results are driven by the behavior of a single architecture and to assess the heterogeneity across the models. As in Guidolin et al. [3], we ensure statistical power and reproducibility by generating the portfolio allocations with carefully studied prompts and systematic API calls. In this paper, we have 200 calls per prompt, and being 4 prompts per model, we thus have a total of 800 calls per different LLM.

3.3 Prompt Structure and Levels of Guidance

The design of this experiment is made by a four-level prompting structure. Each level reflects a different amount of guidance provided to the model. The levels can be interpreted as an "educational ladder," ranging from a user with very limited financial knowledge (e.g. a high school student) to one with advanced technical training (e.g. a PhD student). Taken together, the four prompt levels provide a structured way to study how the amount of optimization-related information affects the behavior of the LLM. Level 0 captures basic financial intuition, Level 1 introduces some qualitative reasoning, Level 2 formalizes the optimization problem, and Level 3 adds numerical and procedural guidance.

The prompts are carefully crafted and tested on a pilot experiment before getting to the 800 calls batch. We have some common denominators across the prompts: in fact, every prompt level mentions that the LLM is advising a retail investor who is risk-averse and wants a diversified portfolio. Across all prompt levels, the output format requirement also remains fixed: the model must return only a table containing ticker symbols and portfolio weights, with non-negative weights summing exactly to one and no short-selling constraints. We also

ask the models to provide a short explanation (maximum 3 sentences) of how the portfolio was constructed. We set the temperature to 0.2 to minimize variability between the runs. As mentioned in Huang et al. [1], the temperature sets the level of randomness in the model's responses. Therefore, what changes between prompts is only the degree to which portfolio optimization concepts are made explicit.

This design makes it possible to test not only whether prompt complexity matters, but also whether there are diminishing returns to an increasingly technical prompting.

3.3.1 Level 0: No Guidance

The prompt at Level 0 corresponds to a very limited-information user, that we can interpret as as a high-school level investor. In this level, no portfolio optimization technique is mentioned. With Level 0, we want to establish a baseline allocation.

3.3.2 Level 1: Low Guidance

The first prompt level introduces the economic intuition that underlies portfolio construction, but without explicitly formalizing it. This level can be interpreted as corresponding to a bachelor's-level user. The prompt states that the investor wants to balance expected return and risk, and it reminds the model that higher expected returns are desirable and lower volatility and lower correlations are preferable. The objective of this level is to test whether the model reacts to qualitative financial logic even in the absence of a formal optimization problem.

3.3.3 Level 2: High Guidance

Level 2 moves from qualitative reasoning to a more formal and explicit explanation. This prompt can be interpreted as corresponding to a master's-level user. Here, the model is directly asked to construct a portfolio using a mean-variance optimization framework, even though we don't explicitly specify the objective function and constraints in mathematical form yet. This level tests whether formalizing the optimization problem, introducing the mean-variance concept and clearly setting lambda equal to 0.5 leads the model to produce allocations closer to a benchmark mean-variance solution.

3.3.4 Level 3: Over-Guidance

The prompt Level 3 provides extensive background information and detail, and it can be interpreted as corresponding to a PhD-level user or almost an algorithmic instruction set.

In addition to the explicit mean-variance objective and constraints with the formula (with lambda set to equal 0.5), the prompt explains that the problem is a constrained quadratic optimization problem and suggests a possible numerical solution approach (quadratic programming). The model is also given the following list of steps to follow:

```
Steps:
1. Use the expected returns and covariance matrix
2. Apply mean-variance logic
3. Ensure that all weights are non-negative and sum exactly to 1
4. Return the final portfolio weights
```

This level is designed to test whether providing a very detailed background and a step-by-step guidance improves the allocation, or instead only generates confusion and additional noise. In further extensions, this level could also give the LLM short technical materials on mean-variance optimization.

4 Results

The implementation of this experiment is available online [4]. We evaluate the results using different metrics across five dimensions.

4.1 Feasibility Metrics

First, for each generated portfolio, we compute the feasibility rate, which we define as the fraction of valid portfolios across the replicas. A "valid" portfolio would be one that satisfies all the following constraints: all weights sum to one, all weights are non-negative, the full asset universe is used (i.e. all 30 assets are included in the output, with no duplicates) and the output is correctly formatted and parsable. The main problem in the results is that, in all the levels, at least 9% of the generated portfolio weights do not sum to one, even though the models are explicitly instructed to do so. In particular, in Table 1 we notice that Sonnet-4.6 seems to have specific trouble in respecting this constraint.

Apart from this, the LLMs perfectly understand the non-negativity constraints (with 0% error rate for all models and prompts, hence its omission from Table 1), and generally also the other requirements: not having duplicates and including all the tickers without "inventing" any. Only GPT-4.1, in the results from prompt Level 0, has a minimal percentage (1.5%) of non-feasible portfolios: two replicas have an additional row, resulting in having a duplicate ticker. One replica only has 29 rows, resulting in a ticker mismatch (because of the missing

ticker). We also observe one replica with an unexpected ticker ("XOM", instead of "WMT", which is missing) and finally, one replica has one duplicated ticker, thus also causing a ticker mismatch.

The table below displays the fraction of violations across the models and prompt levels. A lower value indicates better compliance to the requirements.

Table 1: Error rates of LLM-generated portfolios across prompt levels

Model	Level	Sum $\neq 1$	Wrong Rows	Duplicates	Ticker Mismatch
GPT-4.1	L0	0.995	0.015	0.015	0.015
	L1	0.403	0.000	0.000	0.000
	L2	0.403	0.000	0.000	0.000
	L3	0.289	0.000	0.000	0.000
GPT-5.4	L0	0.910	0.000	0.000	0.000
	L1	0.886	0.000	0.000	0.000
	L2	0.090	0.000	0.000	0.000
	L3	0.547	0.000	0.000	0.000
Sonnet	L0	0.955	0.000	0.000	0.000
	L1	1.000	0.000	0.000	0.000
	L2	1.000	0.000	0.000	0.000
	L3	1.000	0.000	0.000	0.000

Before proceeding to the remaining sections, we clean the dataset. We normalize the weights in each portfolio, so that their sums equal exactly 1. Moreover, we remove the unexpected and invalid ticker, aggregate the duplicate tickers (by summing their weight) and the missing tickers are added with a weight of 0.

4.2 Similarity to Benchmark Portfolios

To evaluate how closely LLM-generated portfolios approximate standard allocation strategies, we compare them to benchmark portfolios using the following distance and similarity measures: the L1 distance, the Euclidean (L2) distance, the correlation between the weight vectors and the overlap in the top holdings. The benchmark portfolios are the following: (1) the naive equally-weighted portfolio, (2) the mean-variance optimization portfolio and (3) the minimum-variance portfolio.

The results show that Level 0 prompts have a very low L1-distance (≤ 0.34) to the naive equally-weighted portfolio. This shows that when prompted with Level 0, the model behaves

like naive diversification. GPT-4.1 with prompt Level 2 shows the lowest L1-distance (0.72) to the mean-variance portfolio and also the highest correlation to it (0.85). Thus, this result shows that an LLM can get close to optimize the portfolios if the prompts are well structured. However, as a consequence of this, we also notice that having an excessively detailed prompt does not always improve the results: in terms of L1-distance, Level 3 prompt is slightly worse than Level 2 for GPT-4.1 and GPT-5.4. For Sonnet, Level 3 improves the correlation to mean-variance but doesn't change much compared to prompt Level 2 when measuring the distance.

4.3 Performance Metrics

To detect the performance of the portfolios, we compute the expected return, expected variance, ex-ante Sharpe ratio, assuming a constant risk-free rate, and Herfindahl index (HHI), as a measure of portfolio concentration.

The results in Table 2 show that the portfolios generated by the LLMs have a progression across prompt levels. In fact, we notice that lower prompt levels produce more diversified portfolios with lower expected returns, resembling the values of equally-weighted allocations. As the LLM receives more instructions, the allocations exhibit a higher concentration and also higher expected returns, being closer to the mean-variance optimization strategy.

Table 2: Performance of LLM and benchmark portfolios

Portfolio	Return	Volatility	Sharpe	HHI
Panel A: LLM Portfolios				
gpt4.1-L0	0.0024	0.0251	0.0956	0.0358
gpt4.1-L1	0.0041	0.0260	0.1562	0.1555
gpt4.1-L2	0.0046	0.0314	0.1468	0.2717
gpt4.1-L3	0.0044	0.0313	0.1411	0.2230
gpt5.4-L0	0.0023	0.0238	0.0987	0.0387
gpt5.4-L1	0.0029	0.0227	0.1276	0.0598
gpt5.4-L2	0.0036	0.0226	0.1596	0.1322
gpt5.4-L3	0.0036	0.0246	0.1446	0.0739
sonnet-L0	0.0024	0.0251	0.0961	0.0354
sonnet-L1	0.0038	0.0249	0.1523	0.1026
sonnet-L2	0.0039	0.0252	0.1561	0.1174
sonnet-L3	0.0041	0.0260	0.1565	0.1215
Panel B: Benchmarks				
EW	0.0022	0.0260	0.0833	0.0333
MV	0.0055	0.0339	0.1624	0.5000
MinVar	0.0019	0.0200	0.0943	0.0980

In a similar way to what is done by Romanko et al. [5], we build a graph showing the LLM portfolios' risk-return profiles, which change across the models and their prompt levels. In Figure 1 below, we observe that, for instance, GPT-4.1 for Level 2 and 3 tends to produce high-risk and high-return portfolios, approaching the results of a mean-variance portfolio. On the other hand, GPT-5.4 exhibits more balanced allocations and Sonnet-generated portfolios are mostly clustered, showing limited dispersion.

Moreover, we notice that all Level 0 models are clustered around the naive equally-weighted portfolio. Again, we confirm that the high school (Level 0) prompt leads the LLM to behave in a naive diversification manner when allocating the weights. As we go towards higher level prompts, we get to higher returns, and sometimes also higher risk. Overall, GPT-5.4 looks more conservative and brings lower returns than GPT-4.1.

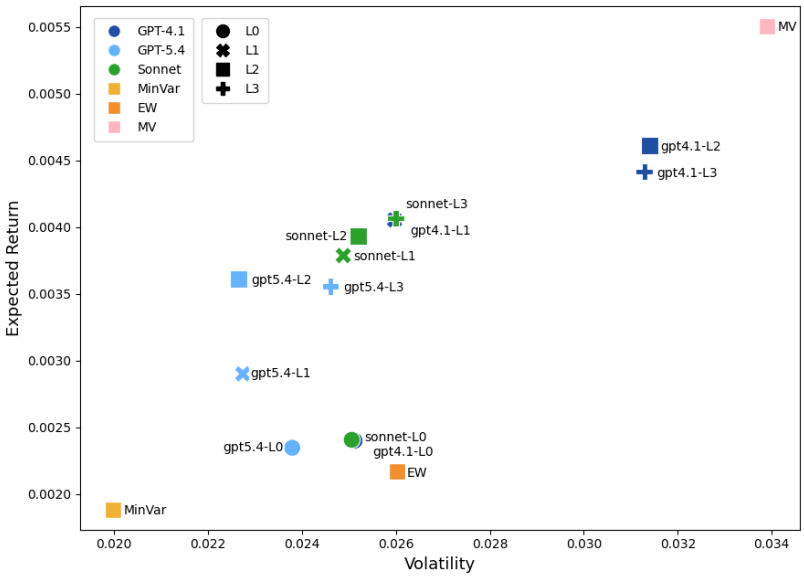

Figure 1: Risk-Return of LLM and Benchmark Portfolios

Throughout this section, we discovered that LLM portfolios don't fully match the mean-variance benchmark. Even though more structured prompts push towards more return-seeking outcomes and also closer to the optimal Sharpe ratio, LLMs are not fully able to optimize correctly.

4.4 Robustness Analysis

To account for the stochastic nature of the LLM outputs, we repeated each prompt multiple times to evaluate the variability of portfolio weights across replications and the stability of

top holdings. We also use the cosine similarity to evaluate the semantic robustness of the generated comments.

We discover that all models across all four prompts produce a rather consistent behavior, with very stable (almost deterministic) portfolio allocations. The variability of the weights is lower than 2% everywhere. Nonetheless, for the stability across the top holdings, we get more interesting results. We observe that the stability doesn't exhibit a monotonic behavior. For instance, GPT-4.1 shows a dip in L1 (with 63.6% overlap of top holdings), but has it peaks at L0 and L3 (83.5% and 86.1% respectively). Sonnet, instead, presents a peak at L1 (94.6%) and the lowest overlap is at L0 (77.9%). Thus, the consistency of the LLM's decision-making is influenced by the prompt structure, but also depends on the model used.

Finally, we measure the semantic consistency of the LLM-generated comments using the cosine similarity between the embeddings. We don't observe a constant increase to a more standardized reasoning as we go to higher prompt levels for all the models. In fact, GPT-5.4 shows similarity across all levels (meaning the LLM is quite indifferent to the differences in the prompts) and Sonnet demonstrates an increase until L2, with then a decline for L3. Especially with Sonnet and GPT-4.1, we understand that LLMs can reach a break point in the improvement of their reasoning structure, after which additional detail doesn't necessarily lead to more consistent outputs.

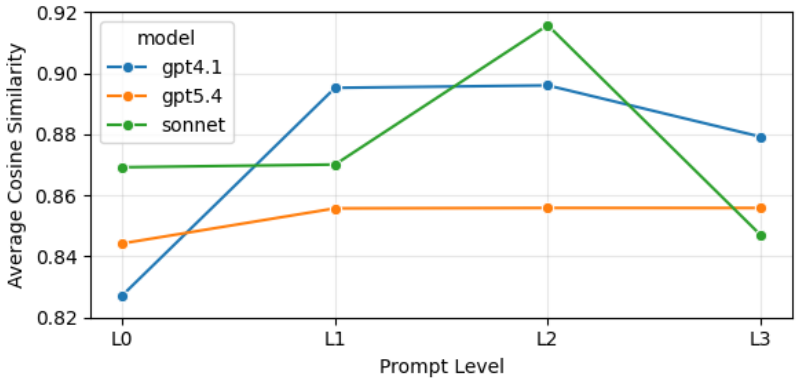

Figure 2: Cosine Similarity Across Prompt Levels

4.5 Semantic Analysis of the LLM Explanations

In this section, we analyze the textual comments provided by each LLM to assess more in details how reasoning evolves across the prompts.

In particular, we examine the presence of financial concepts (e.g., diversification, risk, return, mean-variance, covariance, etc) and whether there is consistency between the reasoning and how it allocated the portfolio, also considering the level of guidance the model had been given.

First, we perform a basic cleaning of the text by putting it all in lower case. We then define some semantic categories for our analysis. These categories are based on essential financial concepts (some of which are mentioned above), and each term is associated to a list of words correlated to it. Using this semantic dictionary, we detect the keywords in the comments generated by each model, per level. Figure 3 is a multi-panel of bar charts that shows the frequency of the key semantic features across the prompt levels and models. We observe that lower prompt levels are dominated by language like "diversification" and "heuristic". Instead, in higher prompt levels, there are more formal concepts such as "optimization", "mean-variance" and "covariance". This marks a shift in the explanations as we go to more sophisticated prompts.

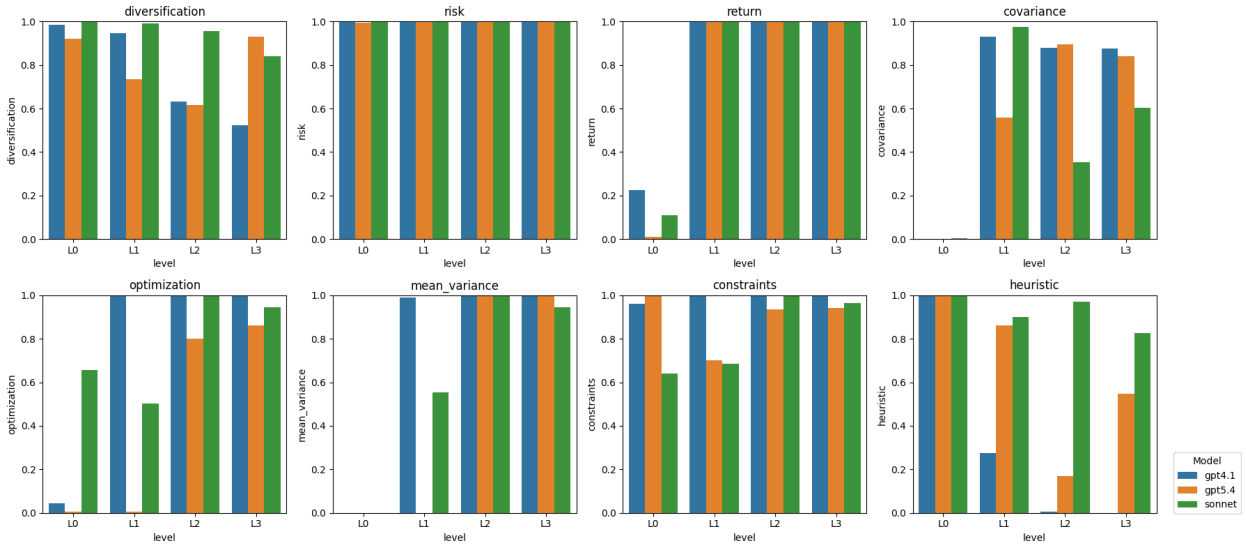

Figure 3: Semantic Features Across Prompt Levels and Models

A delta analysis, which shows the change in semantic feature frequency between the most basic (L0) and most structured (L3) prompt levels for each model, also reflects the same results. In Table 3 below, positive values indicate an increased usage of a given concept under the L3 prompt compared to L0, while the negative values reflect a reduction in its use. The fact that concepts related to diversification appear frequently even in L0 suggest that LLMs possess some prior intuitive logic.

Table 3: Change in semantic features from L0 to L3

Model	Δ Diversification	Δ Heuristic	Δ Optimization	Δ Mean-Variance	Δ Covariance
GPT-4.1	-0.46	-1	+0.95	+1.00	+0.88
GPT-5.4	+0.01	-0.45	+0.86	+1.00	+0.84
Sonnet	-0.16	-0.17	+0.29	+0.95	+0.6

We further analyze the top words that appear in each model, both in Level 0 and Level 3. With GPT-4.1, L0 shows a very human-like portfolio reasoning, with an intuitive logic related to diversification and sector allocation. In L3 instead, GPT-4.1 exhibits a more formal logic, with a vocabulary related to constraints and optimization. GPT-5.4 appears slightly more finance-aware than GPT-4.1 already from L0, as its top terms are for example "healthcare", "staples", "defensive", "technology". Its thinking is more technical and precise to the sector. With prompt L3, GPT-5.4 is also more formal and technical than GPT-4.1: it explicitly mentions lambda and understands constraints in a deeper way. Finally, Sonnet behaves a bit differently, as it frequently mentions specific stocks and sounds more data-driven across both prompt levels.

5 Conclusion

In conclusion, this paper investigated whether an increasing prompt sophistication improves the ability of Large Language Models to construct portfolios within a mean–variance optimization framework. The results show that higher level prompts do improve the reasoning, but not necessarily the allocation of the portfolio weights.

Across the models used in the experiment, we observe some growth from naive diversification in the lowest level to more return-seeking allocation in Levels 2 and 3, which are closer to the mean-variance benchmark as measured by distance and correlation metrics. Moreover, we notice that higher prompt levels do not always lead to increasing improvements. In some cases, they also brought diminishing returns. This is also supported by our semantic analysis of the LLM-generated comments, where even though the explanations are more technical and display a correct optimization reasoning, the allocations corresponding to them are inconsistent with the logic.

We can thus highlight a limitation of the LLMs: even though they have both the financial intuition and access to a mathematical structure, they keep struggling to execute the optimization procedure correctly and efficiently. Future research could explore hybrid approaches that combine LLMs with explicit optimization solvers, as well as using a greater or

smaller lambda, or also setting a different temperature.

More generally, our results are in line with the ones from Guidolin et al. [3], suggesting that autonomous LLMs, even with careful prompt engineering, aren't a good enough substitute for portfolio optimization techniques. Even though the results for some models come close to a mean-variance optimization framework, prompt engineering alone is insufficient to overcome the limitations of LLMs in quantitative decision-making tasks.

References

[1] Fangzhou Lu, Lei Huang, and Sixuan Li. ChatGPT, Generative AI, and Investment Advisory. SSRN, July 2023. SSRN Working Paper.

[2] Sean Cao, Wei Jiang, Junbo Wang, and Baozhong Yang. From Man vs. Machine to Man + Machine: The Art and AI of Stock Analyses. *Journal of Financial Economics*, 2024.

[3] Massimo Guidolin, Richard Harris, Giulia Paggini, and Manuela Pedio. Can LLM-Generated Portfolios Compete with Portfolio Theory? A Large-Scale Out-of-Sample Study. Working Paper, February 2026.

[4] Prompting Large Language Models for Portfolio Optimization: Source Code. `https://github.com/letiziadimonopoli/0526-LLM-for-portfolio-optimization`, 2026. GitHub Repository.

[5] Oleksandr Romanko, Akhilesh Narayan, and Roy H. Kwon. ChatGPT-based Investment Portfolio Selection, August 2023. First version.