Journal of
Risk and Financial Management

MDPI

Article

Few-Shot Portfolio Optimization: Can Large Language Models Outperform Quantitative Portfolio Optimization? A Comparative Study of LLMs and Optimized Portfolio Allocators

Lamukanyani Alson Mantshimuli * and John Weirstrass Muteba Mwamba

Department of Finance and Investment Management, College of Business and Economics, University of Johannesburg, P.O. Box 524, Auckland Park, Johannesburg 2006, South Africa; jwmm@yorku.ca
* Correspondence: mantshimuli@gmail.com

Abstract

Recent advances in large language models (LLMs) have raised questions about their potential role in portfolio allocation beyond traditional sentiment analyses. This study investigated whether LLMs, when prompted directly, can autonomously generate portfolio weights that compete with classical optimization and AI-enhanced strategies. We evaluated seven medium-sized open-source LLMs—Gemma-7B, Mistral-7B, Jansen Adapt-Finance-Llama2-7B, DeepSeek-R1-8B, QuantFactory Llama-3-8B-Instruct-Finance, Qwen-7B, and Llama2-7B—using systematic prompt engineering and temperature tuning. Portfolios were constructed from financial news headlines for S&P 500 equities and benchmarked against mean–variance optimization (MVO), the Black–Litterman model, AI-driven optimizers, and naive diversification strategies. The results show that, while LLM-generated portfolios outperformed naive diversification (Sharpe ratio up to 0.741), they lagged behind AI-optimized benchmarks (Sharpe ratio up to 1.361). A transaction cost analysis revealed that low-turnover LLM strategies retain their competitiveness post-costs, surpassing cap-weighted benchmarks. Statistical tests confirmed significant performance differences ($p \leq 0.01$). These findings highlight the ability of LLMs to extract signals from unstructured text, but also their limitations without explicit optimization. Future research should explore hybrid frameworks that combine LLM reasoning with quantitative optimization for cost-sensitive environments.

Keywords: large language models (LLMs); portfolio optimization; few-shot prompting; prompt engineering; temperature tuning; risk-adjusted performance; transaction costs

check for updates

Academic Editor: Thanasis Stengos

Received: 7 November 2025
Revised: 1 December 2025
Accepted: 2 December 2025
Published: 28 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

In recent years, large language models (LLMs) have surged to the forefront of artificial intelligence research, showcasing an unprecedented performance in understanding natural language, engaging in complex reasoning, and adapting to tasks with minimal examples, known as few-shot learning. These advancements have sparked interest across a broad array of disciplines, including finance, where decision-making traditionally relies on rigorous mathematical frameworks and data-driven insights. One particularly intriguing question has emerged: Can LLMs, when explicitly prompted, design investment portfolio allocations that rival those generated by conventional optimization strategies or more modern approaches infused with machine learning techniques? If so, this would signal a paradigm shift, where language-based AI systems could begin to play an active role not

only in interpreting and providing insights on financial text data, but also in formulating tactical asset allocation strategies that are competitive, robust, and context-aware.

Building on this context, while prior research has focused on integrating LLM-derived signals into quantitative optimizers, this study explored whether LLMs can autonomously generate portfolio weights that balance profitability and risk under realistic constraints. This question is particularly relevant in regulated financial environments, where frameworks such as the Dodd–Frank Act (United States), Markets in Financial Instruments Directive II (MiFID II), and Undertakings for Collective Investment in Transferable Securities (UCITS) (European Union) demand transparency, reproducibility, and robust risk management. By benchmarking LLM-driven allocations against classical and AI-enhanced strategies, we assessed not only their risk-adjusted performance, measured through Sharpe, Sortino, and CVaR ratios, but also their compliance implications. This positions our work at the intersection of natural language reasoning, profitability optimization, and regulatory-grade portfolio design, introducing a new paradigm for cost-sensitive and interpretable investment strategies.

To explore this possibility, researchers have begun experimenting with prompt engineering as a way to steer LLMs toward portfolio-related tasks, ranging from stock selection driven by scenario prompts (Huang et al., 2024) to hybrid frameworks that use LLMs to discover allocation heuristics and then apply explicit mathematical weighting rules (Yuksel, 2025). Other studies have shown gains in ChatGPT-assisted (ChatGPT based on GPT-3.5) asset selection and diversification when coupled with downstream optimizers (Ko & Lee, 2024; Romanko et al., 2023) or metaheuristics (Huang et al., 2024), and LLM-guided macro/sector allocation using policy and economic text streams (Heng et al., 2025). These developments collectively indicate that prompt engineering can produce useful signals and structures for portfolio design, though most approaches ultimately rely on explicit optimization engines to generate the final weights.

1.1. Research Gap

Previous studies have largely used LLMs for sentiment extraction or as auxiliary components within hybrid frameworks, relying on downstream optimizers for the final weight generation. Few works have evaluated LLMs as autonomous allocators producing complete portfolio weights without mathematical optimization. This paper addresses this gap by benchmarking LLM-driven allocations against classical and AI-enhanced methods. To achieve this, we define the specific research objectives before outlining the key contributions of our work.

1.2. Objectives of this Study

The primary objectives were as follows:

- Assess whether LLMs can autonomously generate portfolio allocations that are competitive with traditional and AI-based optimizers.
- Quantify the impact of prompt engineering and temperature tuning on allocation quality.
- Evaluate implementability through a transaction cost analysis and turnover metrics.

1.3. Contributions of this Study

This study contributes to the literature in four key ways:

- Methodological Innovation: We introduce a framework for direct LLM-prompted portfolio generation, thus eliminating intermediate optimization layers, and evaluated prompts/temperatures as inference-time design levers within a cost-aware, statistically tested backtest.

- Empirical Benchmarking: We provide a comprehensive comparison of LLM allocations against classical and AI-enhanced optimization benchmarks as well as naive diversification strategies, quantifying performance gaps.
- Transaction Cost Analysis: We demonstrate how LLM-generated portfolios' low turnover mitigates the trading cost drag, a previously underexplored advantage in the use of LLMs for portfolio allocation.
- Practical Insights: We identified prompt engineering and temperature tuning as critical levers for improving LLM-generated allocations, offering actionable guidance for future applications.

Our contribution is primarily empirical and methodological, situated within the broader literature on LLM-driven portfolio design. Previous studies have explored prompt engineering for asset selection and hybrid allocation strategies (Huang et al., 2024; Ko & Lee, 2024; Romanko et al., 2023; Yuksel, 2025), but these approaches generally rely on downstream optimizers. In contrast, we evaluated LLMs as autonomous allocators, prompting them to generate complete weight vectors under portfolio constraints and assessing their implementability relative to established benchmarks.

Methodologically, our approach differs from prior prompt-engineering efforts in three ways. (i) Standalone allocation: we required LLMs to emit normalized, long-only weights that sum to one, with no downstream optimizer (contrasting with hybrid prompt and optimizer pipelines (Huang et al., 2024; Ko & Lee, 2024; Romanko et al., 2023)). (ii) Inference-time tuning: we searched over prompt variants (zero-/few-shot with macro context and chain of thought) and decoding temperatures to maximize the out-of-sample risk-adjusted performance, treating prompts as design variables. (iii) Implementability focus: we quantified turnover and applied transaction-cost haircuts, and we tested the statistical significance of the Sharpe differences (Leung–Wong) to assess the real-world viability.

To evaluate the efficacy of LLMs in portfolio allocation, we tested seven medium-sized open-source models: Gemma-7B, Mistral-7B, Jansen Adapt-Finance-Llama2-7B, DeepSeek-R1-8B, QuantFactory's Llama-3-8B-instruct-Finance, Qwen-7B, and Llama2-7B. These models represent a mix of general-purpose and finance-fine-tuned architectures, allowing us to assess whether domain adaptation improves the performance. We examined their behavior across multiple prompt designs, including direct instructions and chain-of-thought reasoning. We also varied the temperature settings to explore the trade-off between creative exploration and deterministic output generation. For comparison, we benchmarked LLM-prompt-generated portfolios against six established strategies: (1) classical optimization methods (mean variance optimization (MVO) and Black–Litterman), which rely on quantitative inputs; (2) AI-enhanced approaches (deep learning MVO and FinBERT Black–Litterman), combining traditional optimization techniques with machine learning; and (3) naive diversification strategies (equally weighted and market-cap-weighted portfolios). This structured comparison helped determine whether LLMs can replicate traditional optimization, leverage unstructured data more effectively than hybrid AI methods, or merely approximate simple diversification heuristics.

Our findings revealed a clear performance hierarchy, with AI-optimized portfolios outperforming classical methods and LLM-generated allocations, though LLMs consistently surpassed naive diversification strategies. While LLMs lack explicit optimization mechanics and occasionally produce suboptimal weightings, their text-driven approaches demonstrate an ability to extract useful signals from unstructured financial data. The transaction cost analysis showed that low-turnover LLM variants can remain competitive in real-world settings, as their simpler allocations incur fewer trading costs compared to high-turnover benchmarks. These findings suggest that LLMs may serve as lightweight complements

to traditional methods, particularly in cost-sensitive environments, though they remain unable to replace mathematically rigorous optimization frameworks.

1.4. Organization of the Study

The remainder of this article is organized as follows: Section 2 reviews the existing literature on LLM applications in portfolio allocation while highlighting the research gaps. Section 3 details our methodological framework, including the transformer architecture fundamentals and key variations among LLMs (attention mechanisms, normalization techniques, and activation functions). Section 4 presents our empirical analysis, comprising a preliminary data analysis, the main results, and a detailed discussion. Finally, Section 5 concludes by summarizing the study's key contributions and outlining directions for future research.

2. Literature Review

Artificial intelligence techniques, particularly large language models (LLMs), are increasingly being applied to portfolio construction by extracting actionable insights from unstructured textual data, including financial news headlines, analyst reports, and social media content referencing financial assets such as single stocks and indices. Traditionally, portfolio allocation has relied on mathematical optimization techniques such as mean–variance optimization (MVO) Markowitz (1952) and the Black–Litterman model Black and Litterman (1992). However, the emergence of LLMs has introduced new paradigms for integrating insights from financial text data into portfolio allocation strategies. LLMs have demonstrated strong capabilities in the understanding and generation of natural language, which has led to their adoption in various financial tasks, including, but not limited to, sentiment analyses, financial forecasting, and report generation Zhao et al. (2024). LLMs offer key advantages for financial applications, including the ability to process large volumes of real-time text and understanding nuanced financial language. Their few-shot and zero-shot capabilities also enable rapid adaptation to new tasks with minimal labeled data, making them well-suited for dynamic market environments.

Early applications of LLMs in portfolio management focused on sentiment analyses, treating LLMs as tools to extract qualitative signals from textual data. Models like FinBERT Araci (2019) were fine-tuned to extract sentiment scores from financial news and social media networks, which could then be incorporated into traditional optimization frameworks such as the Black–Litterman model to adjust the expected returns, as implemented by Colasanto et al. (2022). Their research showed that incorporating an LLM-based sentiment analysis into quantitative models such as the Black–Litterman model enhances portfolio construction and leads to higher risk-adjusted returns. Similarly, Kirtac and Germano (2025) showed that mean–variance-based proximal policy optimization (PPO) integrated with a sentiment analysis from LLaMA 3 leads to reduced drawdowns and higher Sharpe ratios. Furthermore, Iacovides et al. (2024) constructed a long–short portfolio using sentiment scores generated by FinLlama, a finance-specific LLM, and demonstrated a superior performance under volatile market conditions. These studies collectively illustrate the potential of LLMs to enhance portfolio strategies by injecting rich qualitative information in the form of financial text data into quantitative models. However, these applications commonly position LLMs as preprocessing components, extracting insights from financial text, while relying on separate optimization engines to make final allocation decisions. This reveals an important research gap: while LLMs excel at reasoning over complex financial language, their potential to directly generate portfolio weights remains underexplored.

To address this gap, our study investigated the direct use of LLMs for generating portfolio weights through prompting. To the best of our knowledge, no study has prompted LLMs directly to produce portfolio allocations without integrating it with other portfolio allocation methods. The existing approaches either constrain LLMs to supplementary roles within hybrid frameworks (e.g., optimization algorithms or multi-agent systems) or leverage them for partial tasks in the portfolio pipeline, rather than autonomous end-to-end allocation. For instance, Huang et al. (2024) used a novel LLM-based multi-objective evolutionary algorithm (MOEA) named LLM-PC-NSGA-II, where the LLM, guided by well-structured prompts, generates new solutions while non-dominated sorting and crowding distance calculations steer the evolutionary process toward optimal portfolio allocations. While their study was innovative by embedding an LLM within the NSGA-II framework, the LLM's role was confined to proposing candidate solutions rather than autonomously navigating the risk–return landscape to produce portfolio allocations. Moreover, by relegating the LLM to a "proposal engine", Huang et al. (2024) did not fully leverage the model's contextual reasoning capabilities, as each candidate vector was still evaluated solely on quantitative objectives, with no feedback loop that allowed the LLM to iteratively refine its internal understanding of trade-offs.

In addition to the work by Huang et al. (2024), recent studies have explored the use of LLMs in portfolio construction, though primarily in indirect or auxiliary roles. For instance, Ko and Lee (2024) showed that ChatGPT can assist with asset selection by applying diversification-aware heuristics, but their framework ultimately relies on a traditional mean–variance optimizer to determine the final weights. Likewise, Yuksel (2025) introduced AlphaPortfolio, an LLM-guided evolutionary framework that discovers and refines allocation strategies by combining inverse covariance weighting, entropy regularization, and volatility normalization, achieving a 71% increase in the Sharpe ratio over the equally weighted portfolio. However, this approach requires iterative generation, mutation, and scoring cycles. In a different study, Luo et al. (2025) proposed an LLM-powered multi-agent system for crypto asset allocation, where the final portfolio decisions emerge from agent negotiation rather than direct output generation. While all three methods report promising results in terms of enhanced diversification (Ko and Lee (2024)), an improved drawdown resilience Yuksel (2025), and a superior classification accuracy Luo et al. (2025), they also introduce computational complexity and intermediate optimization layers that obscure the transparency of weight decisions. In contrast, our work evaluated whether competitive allocations can be generated directly through LLM-prompt-based reasoning, without relying on downstream optimizers, evolutionary algorithms, or multi-agent coordination, thereby improving the interpretability and reducing the computational overhead.

This study employed seven medium-sized open-source LLMs to generate portfolio allocations through systematic prompt engineering, progressing from zero-shot to few-shot formats and varying temperature settings. In the zero-shot configuration, models rely solely on the descriptive content of the prompt, drawing exclusively on their pretraining knowledge. However, this often results in suboptimal outputs within financial contexts Shah and Chava (2023). To mitigate this limitation, few-shot prompting was introduced by embedding exemplar input–output pairs within the prompt to guide the model behavior more effectively. This approach has been shown to improve the task performance in specialized domains, particularly when paired with structured reasoning methods such as chain-of-thought prompting Diao et al. (2023), Agarwal et al. (2024). Notably, Kojima et al. (2022) demonstrated that zero-shot chain-of-thought prompting can elicit strong reasoning capabilities from LLMs, at times rivaling the few-shot performance, especially when the prompts are carefully designed. In addition to the prompt format, this study explored the role of temperature, a key decoding parameter that influences the diversity and consistency

of the model outputs, as emphasized in recent work on reasoning robustness and decoding dynamics Renze (2024). By jointly varying the prompt structure and temperature, this study evaluated how different configurations affect the stability, plausibility, and soundness of LLM-generated allocations, and how these effects may further depend on the underlying model architecture and training approach.

Building on this prompting framework, the analysis proceeded by evaluating the seven selected medium-sized LLMs to assess how architectural and training variations influence their performance under different prompting and temperature settings. All the models evaluated in this study build upon the foundational transformer architecture introduced by Vaswani et al. (2017), which underpins large language models through self-attention mechanisms and layered token representations. Yet, despite this shared foundation, the models diverge meaningfully in their implementation details and pretraining objectives. Recent surveys have highlighted that these differences extend to aspects such as model scaling, multi-modal integration, and parameter-efficient fine-tuning (PEFT) strategies, all of which significantly impact the downstream performance and efficiency Ding et al. (2023); Raiaan et al. (2024); Shao et al. (2024). Additionally, optimal training practices, such as balancing the model size with the training data volume, have been shown to further enhance model capabilities while reducing the computational costs Hoffmann et al. (2022).

These differences are reflected in the specific models selected for evaluation. The selected models included several adaptations of Meta's transformer-based Llama architecture Touvron et al. (2023). One such model is Jansen Adapt-Finance-Llama2-7B Cheng et al. (2023), a domain-adapted Llama 2–7B variant fine-tuned on financial texts such as securities and exchange commission (SEC) filings and earnings call transcripts. Another is QuantFactory Llama-3-8B-Instruct–Finance-RAG, an instruction-tuned model designed for financial question answering using retrieval-augmented generation (RAG) techniques QuantFactory (2024). The baseline Llama 2 (7B), which uses standard multi-head self-attention and layer normalization (LayerNorm), was also included for comparison. The other models tested included Google's Gemma-7B Team et al. (2023), which employs root mean square layer normalization (RMSNorm); DeepSeek-R1-8B DeepSeek-AI (2025), which integrates reasoning-aware pretraining objectives; Mistral-7B Chaplot et al. (2023), which incorporates grouped-query attention; and Alibaba's Qwen-7B Qwen (2024), which features sparse attention mechanisms. These models were selected to capture a range of architectural and training differences, such as the attention design, normalization schemes, and domain-specific pretraining, thereby enabling a comprehensive evaluation of how model-specific features influence the performance in portfolio allocation tasks.

Despite a growing body of work that leverages large language models as either signal generators for downstream portfolio optimizers or components within hybrid frameworks and multi-agent systems, there remains a notable absence of studies that treat LLMs as fully autonomous portfolio allocators. In existing approaches, LLMs provide sentiment scores or candidate solutions for genetic algorithms, but the ultimate translation of these outputs into precise weight vectors is invariably handled by traditional mathematical or algorithmic optimizers. Consequently, the inherent reasoning capabilities and contextual understanding embedded within LLMs have not been directly tested against the core decision problem of portfolio allocation. This paper fills that gap by systematically prompting seven medium-sized, open-source LLMs, each with distinct architectural features, attention mechanisms, and domain adaptations, to produce complete allocation vectors under varied prompt designs and temperature settings. By benchmarking these LLM-generated portfolios against classical optimization benchmarks and naive diversification strategies, we not only quantified the extent to which an LLM can internalize risk–return trade-offs, but we also identified the prompt characteristics and model attributes that most strongly

influence the allocation quality. In doing so, we established a new paradigm for integrating natural-language reasoning directly into the asset-allocation process, laying the groundwork for hybrid or fully LLM-driven portfolio strategies in future research.

3. Methodology

We designed a cost-aware, reproducible framework to test whether large language models (LLMs) can autonomously generate long-only portfolio weights from financial text and how those allocations compare to classical and AI-enhanced optimizers. The pipeline comprised the following: (i) model selection and rationale, (ii) inference-time optimization via prompt engineering and temperature tuning, and (iii) backtesting against benchmarks with turnover and transaction costs. This structure isolated LLM reasoning from downstream optimizers to directly assess its allocation capability and practical implementability.

3.1. Data and Universe

To capture market narratives, we aggregated financial news headlines and articles referencing S&P 500 constituents. These texts were sentence-tokenized, ticker-annotated, and formatted into chronological prompts for LLM input. In parallel, historical price data for the same stocks supported benchmark portfolio construction and backtesting. Both datasets spanned two non-overlapping periods: (i) prompt tuning (in-sample) from 1 January 2016 to 31 December 2021 and (ii) evaluation (out-of-sample) from 1 January 2022 to 31 March 2025. A 50-stock universe was selected by mention frequency to ensure liquidity and realistic coverage. This dual-data design enabled LLMs to infer signals from text while allowing for a rigorous performance comparison using price-based metrics.

3.2. Model Set and Architectural Rationale

Building on this data foundation, we evaluated seven decoder-only transformer LLMs (Gemma-7B, Mistral-7B, Llama-2-7B, Adapt-Finance-Llama2-7B, DeepSeek-R1-8B, Qwen-7B, and QuantFactory Llama-3-8B-Instruct-Finance). All share masked self-attention for autoregressive generation (attention mechanism as per Vaswani et al. (2017)):

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{1}$$

where Q, K, and V are the query, key, and value matrices, and d_k is the key dimension. Despite sharing the core transformer architecture, these seven LLMs differ in ways that can materially impact the portfolio allocation performance. As summarized in Table 1, variations in attention mechanisms (e.g., multi-head versus grouped-query), normalization and activation functions, and domain-specific fine-tuning introduce meaningful heterogeneity. These design choices influence how each model interprets financial prompts and translates textual signals into weight vectors. Understanding these architectural differences is essential for interpreting the performance outcomes and benchmarking the allocation quality across models.

3.3. Prompt Engineering and Temperature Tuning

While architectural components define a model's inherent capabilities, their practical performance in financial applications depends critically on generation settings like the temperature (τ) and prompt-engineering strategies. To mitigate prompt sensitivity, we employed a systematic prompt–temperature tuning framework across multiple configurations, ensuring reproducibility and minimizing subjective bias. These inference-time

configurations work in tandem to transform raw logits $\mathbf{z} \in \mathbb{R}^V$ (where V is the vocabulary size) into controlled probability distributions through temperature-scaled softmax:

$$P(w_i|x) = \frac{\exp(z_i/\tau)}{\sum_{j=1}^{V} \exp(z_j/\tau)}, \quad (2)$$

where z_i is the logit for token i, τ is the temperature parameter, and V is the vocabulary size. The temperature parameter τ modulates the prediction entropy across a risk spectrum that directly corresponds to financial decision-making paradigms:

- $\tau \to 0^+$: Deterministic $\arg\max(\mathbf{z})$ selection (zero entropy) for regulatory-grade precision;
- $0.2 \leq \tau \leq 0.6$: Structured outputs (moderate entropy) for balanced portfolio analyses;
- $0.6 < \tau < 1.0$: Fluid generation (elevated entropy) for scenario exploration;
- $\tau \geq 1.0$: Exploratory sampling (high entropy) for stress-testing extreme cases.

Table 1. Summary of LLM architectures, training data, and domains.

Model	Parameters	Attention	Activation/Norm	Training and Fine-Tuning Domain
Gemma-7B	7B	Multi-head (16 heads)	GeGLU/RMSNorm	6T tokens (web, code, math)
Mistral-7B	7.2B	Grouped-query, sliding window	SwiGLU/RMSNorm	∼8T tokens (multilingual web, code)
Llama-2-Chat-7B	7B	Multi-head (16 heads)	SwiGLU/RMSNorm	∼2T tokens (web + code); general domain
Adapt-Finance-Llama2	7B	Multi-head (16 heads)	SwiGLU/RMSNorm	Fine-tuned on financial documents using QA-style prompts
Qwen-7B	7B	Multi-head (32 heads, NTK scaled)	SwiGLU/RMSNorm	2.4T tokens (web, books, code, math)
DeepSeek-R1-8B	8B	Qwen-like attention (32 heads)	SwiGLU/RMSNorm	Distilled on 800 K reasoning/math examples
Llama-3-8B-Instruct-Finance	8B	Multi-head (32 heads, NTK scaled)	SwiGLU/RMSNorm	Fine-tuned on ∼4 K financial Q&A prompts with retrieval

This temperature-risk continuum is operationalized through a structured prompt-engineering framework that identifies the optimal prompt–temperature pair for each LLM. Starting with a baseline zero-shot instruction (e.g., "You are a quantitative portfolio manager. Adjust weights to maximize the Sortino Ratio and factor in news"), the framework iteratively explores alternative phrasings and few-shot examples that incorporate domain-specific allocation patterns, as shown in Appendix A. The prompts were constructed from chronologically ordered headlines within each rebalancing window. Headlines were added sequentially toward each model's token limit; in practice, we did not hit the limits because the prompts used individual sentences (NLTK-extracted) rather than full paragraphs, ensuring consistency across models. Older news was not decayed, but ordering preserved the temporal context. This systematic search ensured configurations that deliver both a strong risk-adjusted performance and financially coherent outputs. Few-shot prompts anchor their responses in quantitative reasoning, while temperature tuning (τ) modulates the output variability, balancing deterministic precision with creative diversification.

To formalize this search for optimal configurations, we imposed standard portfolio constraints and defined an optimization objective over the prompt–temperature space. First, the model-generated weight distribution was normalized to satisfy valid portfolio allocation requirements:

$$\sum_{i=1}^{N} w_i = 1, \quad w_i \geq 0 \quad \forall i \in \{1, ..., N\}, \quad (3)$$

where w_i is the weight of asset i and N is the number of assets. The focus of this study was to evaluate LLMs under realistic portfolio feasibility constraints ($\sum_{i=1}^{N} w_i = 1, w_i \geq 0$), which were enforced during post-processing to ensure comparability across strategies. Although these constraints were included in the prompts, minor deviations occurred due to probabilistic generation. To guarantee strict feasibility and regulatory-grade validity, normalization was applied where necessary. The raw outputs were analyzed, and the results indicated that most LLM allocations required only minimal adjustments, with very small percentages needing normalization. This suggests that, on average, the models inherently respect the feasibility constraints, even without post-processing. Table A2 in Appendix A summarizes these diagnostics, showing that better-performing models generally required fewer corrections. Non-feasible percentages in Table A2 represent the proportion of rebalancing instances where raw weights violated the specified constraint, averaged across all out-of-sample dates. Next, the optimal prompt x^* and temperature τ^* were selected by maximizing the expected Sortino ratio:

$$(x^*, \tau^*) = \arg\max_{x \in \mathcal{P}, \tau \in \mathcal{T}} \mathbb{E}[S_{m,x,\tau}] \tag{4}$$

where x^* and τ^* are the optimal prompt and temperature, and $S_{m,x,\tau}$ is the Sortino ratio for model m under configuration (x, τ). $\mathcal{P}$ is the prompt set and $\mathcal{T} = \{0.2, 0.4, \ldots, 1.0\}$ spans the temperature grid. Once identified, these settings were applied in an out-of-sample evaluation. All the reported evaluations and comparisons were conducted under an identical prompt template for each LLM to ensure fairness. However, we acknowledge that the results are sensitive to the prompt design; alternative prompts could yield different performance rankings. This sensitivity underscores the importance of prompt engineering as a critical factor in LLM-based portfolio construction. The complete workflow, from data preprocessing through prompt–temperature tuning and evaluation, is summarized in Algorithm 1.

3.4. Portfolio Construction and Benchmarks

By leveraging the tuned configurations, monthly rebalancing converted the LLM outputs into portfolio time series. These portfolios were benchmarked against three categories:

- Classical optimizers:
  - Mean–Variance Optimization (MVO):
    We adopted a Sharpe-like objective to approximate the efficient frontier under long-only constraints:

$$\max_{\mathbf{w}} \quad \frac{\mu^\top \mathbf{w} - r_f}{\sqrt{\mathbf{w}^\top \Sigma \mathbf{w}}} \quad \text{s.t.} \quad \sum_{i=1}^{N} w_i = 1, \;\; w_i \geq 0 \;\; \forall i. \tag{5}$$

    where $\mu \in \mathbb{R}^N$ is the expected return vector, $\Sigma \in \mathbb{R}^{N \times N}$ is the return covariance matrix, r_f is the annualized risk-free rate, and $\mathbf{w}$ is the long-only, fully invested portfolio weights. The objective maximized excess return per unit risk, consistent with the Markowitz framework, and was solved via quadratic programming under the rebalancing schedule and transaction cost model in Section 3.5.
  - Black–Litterman (BL):
    The BL framework combines equilibrium returns with investor (or signal-derived) views. Let Π denote equilibrium (implied) returns, P the $K \times N$ pick matrix selecting K views over N assets, $q \in \mathbb{R}^K$ the view vector, and $\Omega \in \mathbb{R}^{K \times K}$ the

(diagonal or full) covariance of the view uncertainty. A common closed form for posterior returns is as follows:

$$\mu_{\mathrm{BL}} = ((\tau\Sigma)^{-1} + P^{\top}\Omega^{-1}P)^{-1}((\tau\Sigma)^{-1}\Pi + P^{\top}\Omega^{-1}q), \tag{6}$$

where $\tau > 0$ scales the prior (equilibrium) uncertainty and Σ is the asset-return covariance matrix. (We followed the standard BL convention in which τ down-weights the confidence in equilibrium returns relative to the views; Ω encodes the reliability of each view.) The posterior μ_{BL} is then used within the same long-only MVO routine (Equation (5)) to obtain the weights. In our AI-enhanced BL benchmark, sentiment-derived views (FinBERT) populate q, holding P, Ω, and τ fixed across strategies for comparability.

Algorithm 1: LLM portfolio allocation: tuning and evaluation.

Input : News corpus D; model set $\mathcal{M}$; prompt set $\mathcal{P}$; temperature grid $\mathcal{T}$; tuning window $\mathcal{W}_{\mathrm{in}}$; evaluation window $\mathcal{W}_{\mathrm{out}}$; constraints $\mathcal{C}$ (long-only, budget, caps); cost rate λ_{tc}; rebalancing schedule $\mathcal{R}$

Output: For each $m \in \mathcal{M}$: portfolio weights $\{w_t^{(m)}\}$ and performance metrics

Data preparation

Format D into chronological prompts; define the 50-stock universe; align price series for backtesting;

for *each model* $m \in \mathcal{M}$ **do**

  Stage A: Prompt–Temperature tuning on $\mathcal{W}_{\mathrm{in}}$

  for *each prompt* $p \in \mathcal{P}$ **do**

    for *each temperature* $\tau \in \mathcal{T}$ **do**

      for *each rebalance date* $t \in \mathcal{W}_{\mathrm{in}}$ *with schedule* $\mathcal{R}$ **do**

        $w_t^{\mathrm{raw}} \leftarrow \mathrm{LLMAllocate}(m, p, \tau, \mathrm{prompt}(t))$;

        $w_t \leftarrow \mathrm{NormalizeToFeasible}(w_t^{\mathrm{raw}}, \mathcal{C})$;

      end

      $S_{m,p,\tau} \leftarrow \mathrm{Sortino}(\{w_t\}_{t \in \mathcal{W}_{\mathrm{in}}})$;

    end

  end

  $(p^{\star}, \tau^{\star}) \leftarrow \arg\max_{(p,\tau) \in \mathcal{P} \times \mathcal{T}} S_{m,p,\tau}$

  Stage B: Out-of-sample evaluation on $\mathcal{W}_{\mathrm{out}}$

  for *each rebalance date* $t \in \mathcal{W}_{\mathrm{out}}$ *with schedule* $\mathcal{R}$ **do**

    $w_t^{\mathrm{raw}} \leftarrow \mathrm{LLMAllocate}(m, p^{\star}, \tau^{\star}, \mathrm{prompt}(t))$;

    $w_t \leftarrow \mathrm{NormalizeToFeasible}(w_t^{\mathrm{raw}}, \mathcal{C})$;

    $\mathrm{Turnover}_t \leftarrow \sum_i |w_{t,i} - w_{t-1,i}|$;

    $TC_t \leftarrow \lambda_{\mathrm{tc}} \cdot \mathrm{Turnover}_t$;

    $r_t^{\mathrm{net}} \leftarrow r_t^{\mathrm{gross}} - TC_t$;

  end

  Summarize {Sharpe, Sortino, Omega, CVaRSharpe, Turnover} over $\mathcal{W}_{\mathrm{out}}$;

end

Benchmarks

Construct MVO, BL, Deep-Learning MVO, FinBERT-BL, EW, CapW with identical universe, schedule, and costs;

Compare

Apply Leung–Wong tests to Sharpe ratios across all strategies; rank by post-cost metrics;

return portfolios and comparative statistics

- AI-enhanced optimizers:
  - Deep Learning MVO: A GRU-based model predicts μ_t and Σ_t dynamically, optimizing the Sharpe ratio under turnover constraints.
  - FinBERT BL: FinBERT sentiment scores replace q in the BL framework, injecting text-derived signals into the expected returns.
- Naive baselines: Equally weighted (EW) and market-cap-weighted (CapW) portfolios.

All strategies shared the same universe and rebalancing schedule for comparability.

3.5. Transaction Cost Model

We applied a simple, implementable transaction cost model based on portfolio turnover. Let $\mathbf{w}_t$ denote the portfolio weights after rebalancing at time t, and $\mathbf{w}_{t-1}$ the weights before rebalancing. One-period turnover is defined as the ℓ_1 change in weights:

$$\text{Turnover}_t = \sum_{i=1}^{N} |w_{t,i} - w_{t-1,i}|. \tag{7}$$

Given a per-unit cost rate λ_{tc} (we used 10 bps per trade), the transaction cost haircut at time t is as follows:

$$TC_t = \lambda_{tc} \cdot \text{Turnover}_t. \tag{8}$$

The net portfolio returns are computed as follows:

$$r_t^{\text{net}} = r_t^{\text{gross}} - TC_t, \tag{9}$$

where r_t^{gross} is the pre-cost portfolio return for the period. We report the annualized turnover as the time-average of Turnover_t scaled to the annual frequency under the chosen rebalancing schedule (monthly in our study). This model ensures comparability across strategies and directly links the trading intensity to the realized performance.

3.6. Backtesting and Metrics

Finally, we assessed the real-world viability through cost-aware backtesting. The performance was summarized using the Sharpe, Sortino, Omega, and CVaR Sharpe ratios. The statistical significance of the differences was tested via Leung–Wong Sharpe equality tests. (The full details and p-value matrices appear in Section 4) This integrated design directly evaluated the LLMs' ability to convert textual reasoning into actionable allocations, while accounting for feasibility and trading frictions. It highlights three dimensions: (a) reasoning quality, (b) sensitivity to the prompt and temperature, and (c) implementability under realistic cost constraints, factors often overlooked in gross performance comparisons. Figure 1 summarizes the overall methodology, illustrating how the LLM-driven portfolios were constructed and benchmarked against traditional and AI-enhanced strategies.

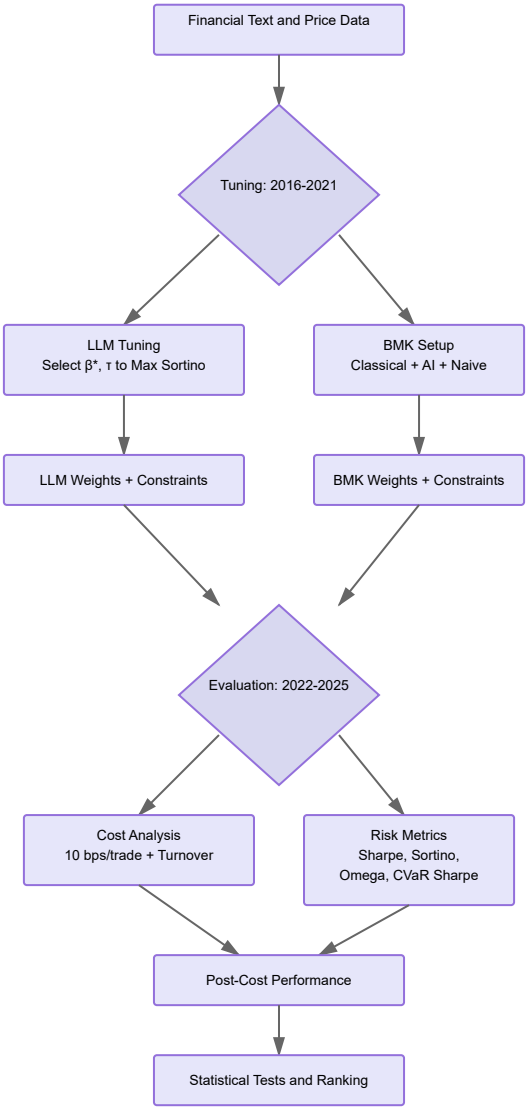


Figure 1. Cost–aware LLM portfolio allocation framework.

4. Empirical Analysis

This section conducts an in-depth empirical evaluation of LLM-prompt-based portfolio strategies, covering data preparation, prompt optimization, and performance benchmarking. Using a multi-stage approach, it assesses how different prompt and temperature configurations affect the portfolio outcomes, comparing LLM-generated allocations to optimized strategies in terms of both pre-cost and post-cost risk-adjusted metrics.

4.1. Preliminary Data Analysis

This study used two different types of data; text data and numerical data. Text data were in the form of financial news and headlines, while numerical data were in the form of historical share prices.

4.1.1. Financial Text Data

The financial text data were collected from various sources, including the Ashraq Financial Headlines Dataset (accessed: 26 May 2025) (https://huggingface.co/datasets/ashraq/financial-news) and the FNSPID Financial News Dataset (accessed: 26 May 2025) (https://github.com/Zdong104/FNSPID_Financial_News_Dataset), and more recent news

was collected from NewsAPI (accessed: 26 May 2025) (https://newsapi.org/). A total of 41,523 financial text data points were obtained from 1 January 2016, to 31 March 2025 (an average of 18 financial text data points per trading day) and used by the LLMs as the input for their asset allocation prompts. Although the dataset was curated for consistency, this design choice prioritized reproducibility over raw realism. Future research should validate the findings on live, heterogeneous news streams.

To provide some insight into the composition of the financial text data, Figure 2 displays the word frequency for the top nine portfolio stocks by market value. A darker blue shading of a word indicates a higher frequency of mentions, while lighter blue signifies a lower frequency of mentions of the word in relation to that specific stock. The figure reveals that words such as "positive", "sentiment", "stock", "risk", "market", "earnings", and "potential" appeared more frequently in sentences or headlines mentioning these stocks. This pattern may be due to the market's focus on evaluating company performance, investor sentiment, and growth prospects when discussing high-value stocks. The pattern also indicates a tendency in financial news to emphasize optimistic outlooks, potential risks, and stock- and market-related themes, reflecting the broader discourse around these prominent stocks.

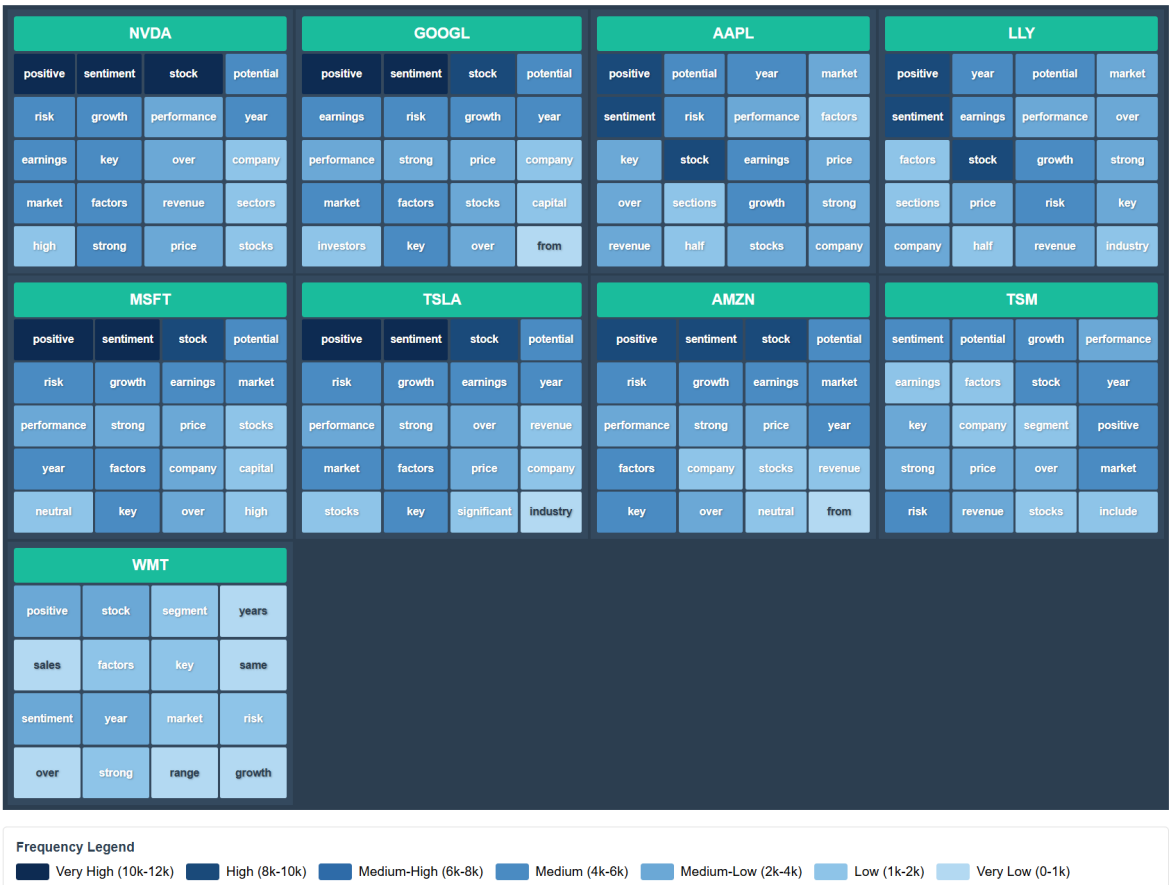


Figure 2. Word frequency by stock (top 9 portfolio stocks by market cap).

4.1.2. Financial Text Data Preparation for LLM Integration

As shown above, the financial text data were well-suited for LLM prompts due to their comprehensive coverage, market sentiment insights, and mentions of stocks in the portfolio. Before the LLMs processed these financial text data, they underwent a four-stage preparation pipeline before being utilized in the LLM prompts:

Stage 1: Data Augmentation. While the Ashraq Financial Headlines Dataset and FNSPID Financial News Dataset provided a foundation for our dataset, their outdated nature (last updated in January 2024) necessitated supplementation with more recent financial news. To achieve this, we utilized the NewsAPI service to collect the most up-to-date financial news, extending our dataset into the year 2025.

Stage 2: Sentence Extraction. The second stage involved segmenting articles into sentences using Python (Version 3.10.11) Natural Language Toolkit (NLTK) tokenizers. We also utilized custom regular expression (RegEx) patterns to identify stock tickers and company names in a case-insensitive manner. For pre-annotated datasets (e.g., Ashraq Financial Headlines Dataset), this step was skipped.

Stage 3: Data Aggregation. In the third stage, we combined the extracted sentences from unstructured sources with pre-annotated headlines. The resulting unified dataset spanned from 1 January 2016 to 31 March 2025 and comprised 41,523 text entries, averaging approximately 18 entries per trading day. All the entries were standardized with stock code annotations to ensure consistency.

Stage 4: LLM Input Preparation. The final preprocessing stage formatted the dataset for LLM compatibility through chronological ordering and model-specific prompt templates. Entries were standardized with timestamps in days and validated for reproducible results across different LLMs. This structured approach ensured fair benchmarking while accommodating each model's unique tokenization and formatting requirements.

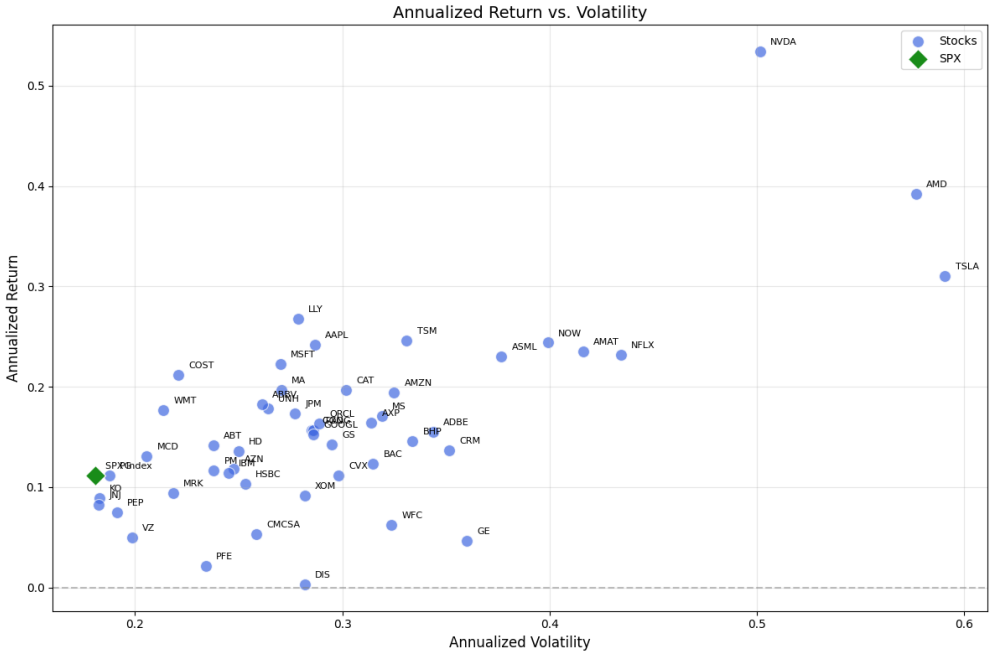
This preprocessing pipeline ensured that the LLMs received comprehensive and equity-specific text data for use in asset allocation prompts. The time-based coverage (2016–2025) provided both recent market sentiment and long-term financial context for more robust asset allocation decisions.

4.1.3. Numerical Data

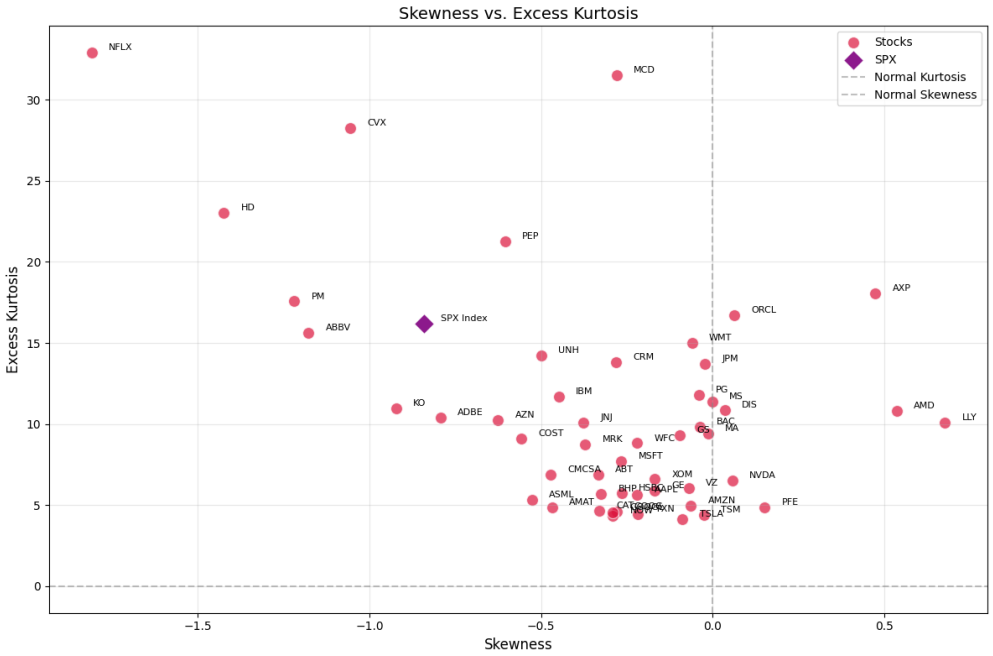
While the LLM-prompt-based portfolio strategies did not rely on numerical data in the form of historical share prices for their initial construction, historical share price data played a crucial role in backtesting their performance and in constructing benchmark portfolios such as the mean–variance optimization (MVO) and Black–Litterman models. We used a portfolio of 50 stocks selected based on their mention frequency in our combined dataset (Ashraq, FNSPID, and NewsAPI). While this approach naturally weights toward large-cap stocks that dominate financial news, the portfolio size ensures meaningful diversification and the 50-stock threshold prevents overconcentration while providing statistically robust backtesting results. These selection criteria allowed the LLMs to effectively leverage the prevailing market sentiment while maintaining a balanced representation of the investment universe. Figure 3 illustrates the distribution characteristics of the 50 portfolio stocks, emphasizing key properties such as their skewness, kurtosis, and central tendency measures in their return distributions.

The figure reveals deviations from normality and different dispersion patterns across the portfolio stocks. The annualized return versus annualized risk scatter plot reveals a positive relationship between annualized returns and volatility, with most stocks having an annualized return of less than 30% and an annualized volatility of less than 40%. The positive slope of the overall relationship confirms that investors are generally compensated for bearing additional risk, although the outliers demonstrate how non-systematic factors such as sector momentum (NVDA and TSLA, for example) can dominate individual stock behavior. The skewness versus kurtosis plot reveals that portfolio stocks exhibit excess kurtosis between 5 and 15 on average, indicating returns with fatter tails and a

sharper peak than a normal distribution. Additionally, most stock returns display negative skewness, suggesting that investors may experience frequent small gains, but occasional large losses. Notably, stocks like NFLX, CVX, and MCD show an extremely high excess kurtosis (exceeding 25), signaling a heightened tail risk. NFLX also stands out with a pronounced negative skew (below −1.5), reflecting a higher likelihood of severe downside moves compared to the upside potential.



(**a**) Risk–return characteristics.



(**b**) Higher-moment analysis.

Figure 3. Distributional characteristics of portfolio constituents: (**a**) demonstrates the risk–return trade-off across assets, while (**b**) highlights non-normal distribution properties through skewness and kurtosis measures.

4.2. Choosing the Best LLM Prompt: Multi-Prompt Temperature Tuning

The initial phase of employing large language models (LLMs) for portfolio allocation involved the selection of optimal prompt formulations and model parameters to maximize the output quality. Among the key parameters, particular attention was given to the model temperature, which governs the trade-off between deterministic behavior and creativity in the model's generated responses. To systematically evaluate these factors, we employed a dedicated prompt-tuning dataset encompassing multiple prompt variations and temperature settings across diverse LLM architectures. Our evaluation framework included seven prominent and finance-fine-tuned models: Gemma-7B, Mistral-7B, Jansen Adapt-Finance-Llama2-7B, DeepSeek-R1-8B, QuantFactory's Llama-3-8B-instruct-Finance, Qwen-7B, and Llama2-7B. This comprehensive approach enabled the identification of optimal configurations, specifically the ideal prompt version coupled with its corresponding temperature setting, while maintaining rigorous methodological standards through the strict segregation of prompt tuning and evaluation datasets to prevent overfitting.

Six distinct prompt templates (prompt descriptions provided in Appendix A) were evaluated across a temperature spectrum ranging from 0.2 to 1.0, enabling a comprehensive assessment of how varying degrees of stochasticity affect the model performance. This evaluation was conducted on the prompt-tuning dataset, which spanned from 1 January 2016 to 31 December 2021, as shown in Figure 4 below.

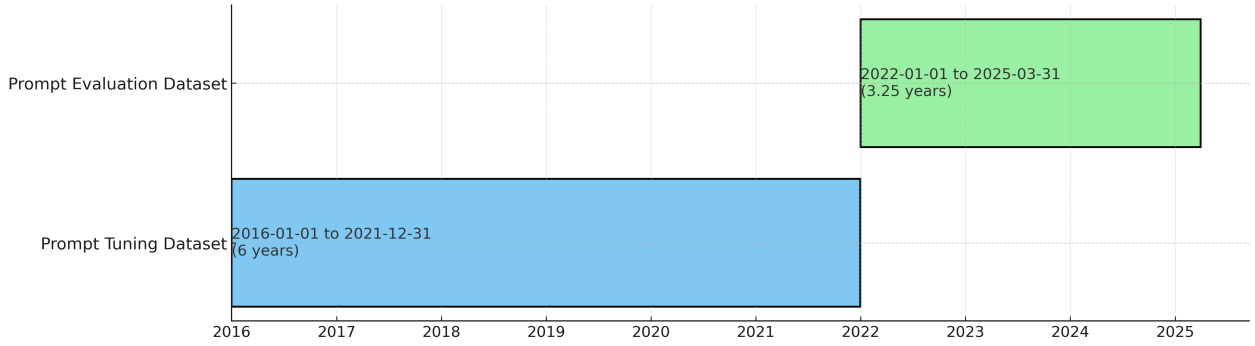


Figure 4. Dataset split: prompt tuning and prompt evaluation.

The empirical results of these experiments, presented in 3-D plots of Sortino ratios against the prompt–temperature combinations for each LLM, are shown in Figure A1 in Appendix A. These plots provide evidence for selecting the most effective prompt–temperature combinations for each LLM, which are then be used to construct LLM-prompt-based portfolios. Figure A1 shows that lower temperature settings, specifically within the range of 0.2 to 0.6, yield higher risk-adjusted returns, as measured by the Sortino ratio. Additionally, Prompt 3 and Prompt 5 resulted in higher average Sortino ratios compared to the other prompts. The superior performance of Prompt 3 and Prompt 5 stemmed from two key factors: (1) their incorporation of the macroeconomic context, enabling better market-wide reasoning, and (2) structured instructions (e.g., "identify key themes" and apply "chain-of-thought" reasoning) that forced the systematic decomposition of news into investment factors. These elements transformed LLMs from passive processors into analytical portfolio constructors, mirroring professional investors' decision frameworks. The combination of a broader economic context with step-by-step reasoning requirements proved particularly effective for generating risk-adjusted allocations.

Among the evaluated models, the Mistral-7B, QuantFactory's Llama-3-8B-instruct-Finance, and Gemma-7B LLMs demonstrated a superior performance, achieving the highest Sortino ratios on average. In contrast, Deepseek-R1-8B underperformed across multiple prompts and temperature settings, indicating its relative inefficiency in the tested scenarios.

These findings suggest that model selection, prompt design, and temperature tuning play critical roles in optimizing risk-adjusted returns. Table 2 shows the best-performing combinations of prompts and temperatures across all models, determined by analyzing both the highest average Sortino ratios per prompt at specific temperatures and the highest average Sortino ratios per temperature for specific prompts. The configurations presented represent the optimal performance conditions for each model.

The resulting portfolios, constructed using these optimal combinations, were evaluated on out-of-sample data (1 January 2022–31 March 2025, as shown Figure 4) against benchmark portfolios. A comparative performance analysis follows in the next section.

Table 2. Optimal prompt and temperature combinations by model.

Model	Temperature	Prompt	Sortino
Mistral-7B	0.40	Prompt 3	1.6252
QuantFactory_Llama-3-8B-Instruct-Finance	0.60	Prompt 3	1.5876
Gemma-7B	0.60	Prompt 1	1.5812
Llama2-7B	0.40	Prompt 5	1.5286
Jansen_adapt-finance-Llama2-7B	0.20	Prompt 5	1.4325
Qwen-7B	0.40	Prompt 6	1.3505
Deepseek-R1-8B	0.60	Prompt 3	1.2247

4.3. Results

The performance of the LLM-prompt-based portfolios, constructed using optimized prompt versions and temperature configurations, was evaluated against three distinct benchmark categories: AI-optimized portfolios, classical optimization approaches, and naive diversification strategies. The AI-based benchmarks included a deep learning MVO portfolio that employed a gated recurrent unit (GRU) neural network trained to maximize the Sharpe ratio through dynamic asset allocation with monthly rebalancing, and a FinBERT Black–Litterman portfolio that generates asset views using sentiment scores derived from the FinBERT large language model. Classical benchmarks consist of a standard Black–Litterman model using historical return means as views and a traditional mean–variance optimization approach set to maximize the Sharpe ratio. Naive diversification benchmarks, such as equally weighted and market-cap-weighted portfolios, were included as fundamental baselines to assess whether more sophisticated methods provide meaningful advantages over simple diversification strategies. This comprehensive comparison framework allowed for a robust evaluation of whether LLM-generated portfolios can outperform established quantitative optimization approaches and naive diversification portfolios.

4.3.1. Portfolio Performance: Pre- and Post-Transaction Cost

Table 3 summarizes both the pre- and post-transaction-cost results, offering a comprehensive view of the risk-adjusted performance across all portfolio categories. A clear hierarchy emerged: AI-optimized portfolios, particularly deep learning MVO and FinBERT Black–Litterman, achieved the highest Sharpe and Sortino ratios, outperforming both classical optimizers and all LLM-generated strategies. Among the classical benchmarks, the Black–Litterman model continued to outperform mean–variance optimization (MVO), while both exceeded the naïve diversification baselines. Within the LLM-prompted group, QuantFactory Llama-3-8B-Instruct-Finance delivered the strongest pre-cost performance (Sharpe = 0.741; Sortino = 1.087). More importantly, it retained most of its advantage after the transaction costs (Sharpe = 0.708; Sortino = 1.038) due to its exceptionally low turnover (0.238). Similarly, Gemma-7B and Mistral-7B, characterized by limited rebalancing, displayed a relatively stable post-cost performance, highlighting the advantage of cost efficiency through lower trading frequency. By contrast, Qwen-7B and DeepSeek-R1-8B

experienced the steepest deterioration in their Sharpe ratios once the costs were applied, falling by more than 20%, a direct consequence of their higher turnover and reactive allocation behavior.

Transaction costs exert a material and redistributive effect on portfolio rankings. While all the strategies experienced reduced Sharpe and Sortino ratios once costs were included, the magnitude of this decline varied sharply with turnover. High-frequency or naïve strategies such as cap-weighted and equally weighted portfolios suffered the largest deterioration (Sharpe drops of 14% and 19%, respectively). In contrast, low-turnover LLM portfolios—notably QuantFactory, Gemma-7B, and Mistral-7B—preserved much of their gross performance, narrowing the performance gap with classical optimization methods. The resulting post-cost ranking shifted slightly as turnover-efficient LLMs climbed closer to mid-tier classical models. Deep learning MVO remained the overall leader (Sharpe = 1.319 post-cost), followed by FinBERT Black–Litterman (0.921) and classical Black–Litterman (0.760). However, QuantFactory Llama-3-8B achieved a net Sharpe ratio comparable to classical MVO, underscoring how the transaction-cost sensitivity can materially influence the realized performance.

Table 3. Performance metrics by optimization strategy (pre- and post-transaction costs).

Portfolio/Model	Type	Sharpe	Sharpe (Post-Cost)	Sortino	Sortino (Post-Cost)	Turnover
Deep Learning MVO	AI-Optimized Benchmark	1.361	1.319	2.002	1.937	0.488
FinBERT Black–Litterman	AI-Optimized Benchmark	0.998	0.921	1.480	1.365	0.534
Classical Black–Litterman	Classical Benchmark	0.827	0.760	1.193	1.097	0.491
Classical MVO	Classical Benchmark	0.751	0.713	1.143	1.084	0.502
QuantFactory Llama-3-8B-Instruct-Finance	LLM Prompt	0.741	0.708	1.087	1.038	0.238
Gemma-7B	LLM Prompt	0.609	0.580	0.869	0.827	0.189
Mistral-7B	LLM Prompt	0.602	0.576	0.858	0.822	0.161
Cap-Weighted	Naive Benchmark	0.615	0.530	0.881	0.759	0.695
Llama2-7B	LLM Prompt	0.588	0.522	0.836	0.742	0.450
Jansen Adapt-Finance-Llama2-7B	LLM Prompt	0.572	0.510	0.825	0.736	0.440
Qwen-7B	LLM Prompt	0.611	0.490	0.856	0.687	0.845
Equally Weighted	Naive Benchmark	0.575	0.466	0.818	0.663	0.761
DeepSeek-R1-8B	LLM Prompt	0.427	0.383	0.661	0.592	0.473

These findings emphasize that turnover control is a decisive determinant of real-world portfolio viability. In low-turnover environments, LLM-driven strategies can offer cost-resilient allocations that remain competitive with traditional optimizers, even when their pre-cost efficiency appears inferior. To illustrate this turnover–allocation relationship, Figure 5 presents the composition of the best-performing LLM portfolio, QuantFactory Llama-3-8B-Instruct-Finance, during the November 2024 rebalancing period.

As shown in Figure 5, the portfolio exhibited substantial concentration in NVDA, AAPL, and JPM (collectively exceeding 31%), indicating correlated exposures and limited diversification. Minimal weights (approximately 0.7%) across more than 30 assets contributed little to efficiency while adding rebalancing frictions that can erode the net performance. This allocation pattern reflects the model's selective bias toward highly capitalized technology and financial equities; this is an LLM-driven tendency that mirrors the sentiment intensity in financial headlines.

Overall, risk-adjusted metrics decline across all strategies when costs are applied, but the magnitude of this decline varies sharply with turnover. AI-enhanced optimizers such as deep learning MVO and FinBERT Black–Litterman remain resilient despite moderate turnover, whereas classical optimizers experience a steeper deterioration. Naïve diversification methods perform the worst due to a persistently high turnover. In contrast,

several LLM portfolios—notably QuantFactory, Gemma-7B, and Mistral-7B—maintained a competitive post-cost performance, in some cases surpassing the cap-weighted benchmarks through their implicit cost-efficiency and restrained trading intensity.

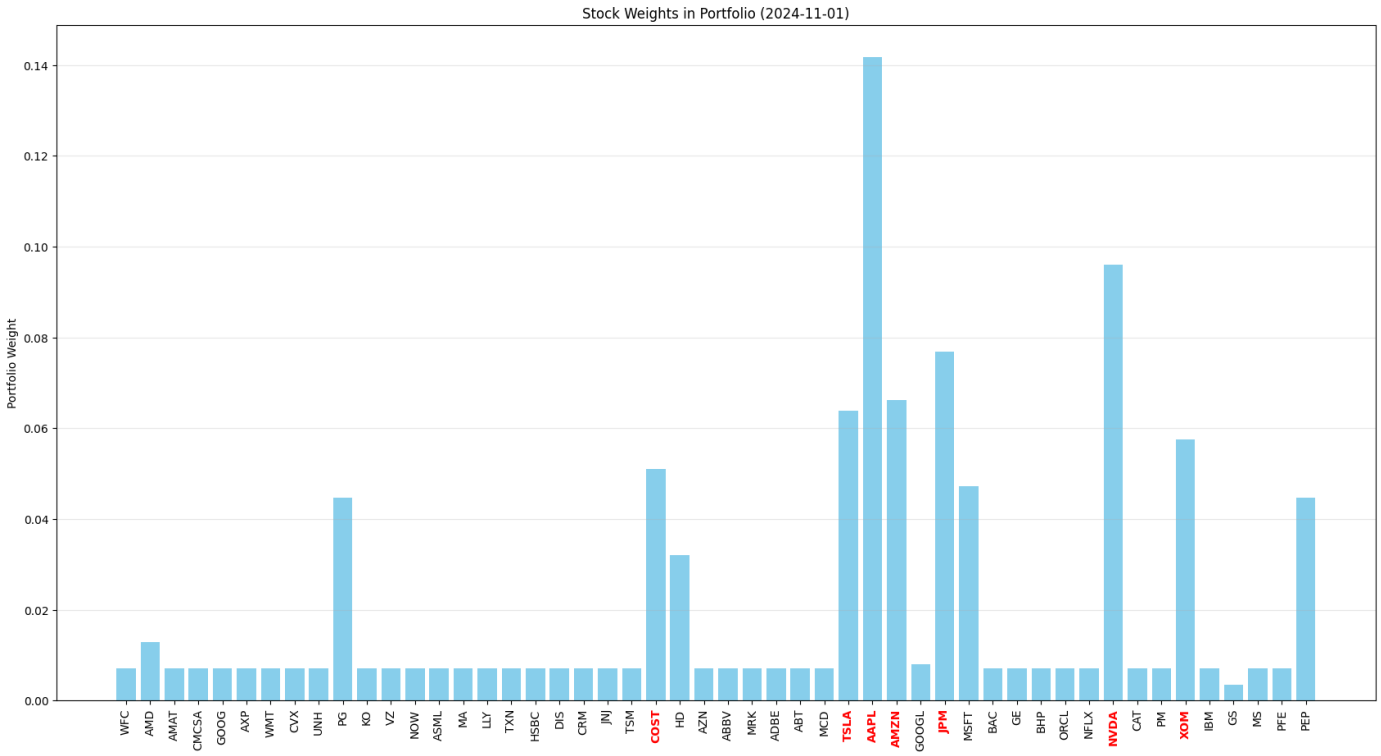


Figure 5. Portfolio weights for QuantFactory Llama-3-8B-Instruct-Finance during November 2024 rebalancing.

4.3.2. Statistical Significance

While the results show that some LLM prompt portfolios perform better than naive diversification portfolios while underperforming compared to AI-optimized and classical optimization benchmarks on a risk-adjusted basis, these comparisons of performance do not tell us if the differences are statistically significant. The Leung–Wong test addresses this by testing if the difference in the Sharpe ratios between any two portfolios is statistically meaningful. Figure 6 below shows the results of the Leung–Wong Sharpe ratio equality test, which reveals statistically significant differences in the Sharpe ratios across portfolios.

Nearly all pairwise comparisons, including those between benchmarks and LLM prompt portfolios, showed overwhelming significance ($p \leq 0.01$), indicating that these performance gaps are highly unlikely to arise by chance. For comparisons with marginally higher p-values, such as the 0.019 difference between Qwen-7B and Gemma-7B, we applied the conventional 5% threshold ($p \leq 0.05$) to assess significance. While these cases represent weaker evidence than the $p \leq 0.01$ results, they still meet the standard for statistical significance in financial research, suggesting meaningful performance distinctions. Together, these findings robustly confirm that some LLM prompt portfolios outperform naive diversification portfolios while being outperformed by AI-optimized and classical optimization benchmark portfolios, highlighting the importance of explicit mathematical optimization in portfolio allocation, something that LLM prompt portfolios currently lack.

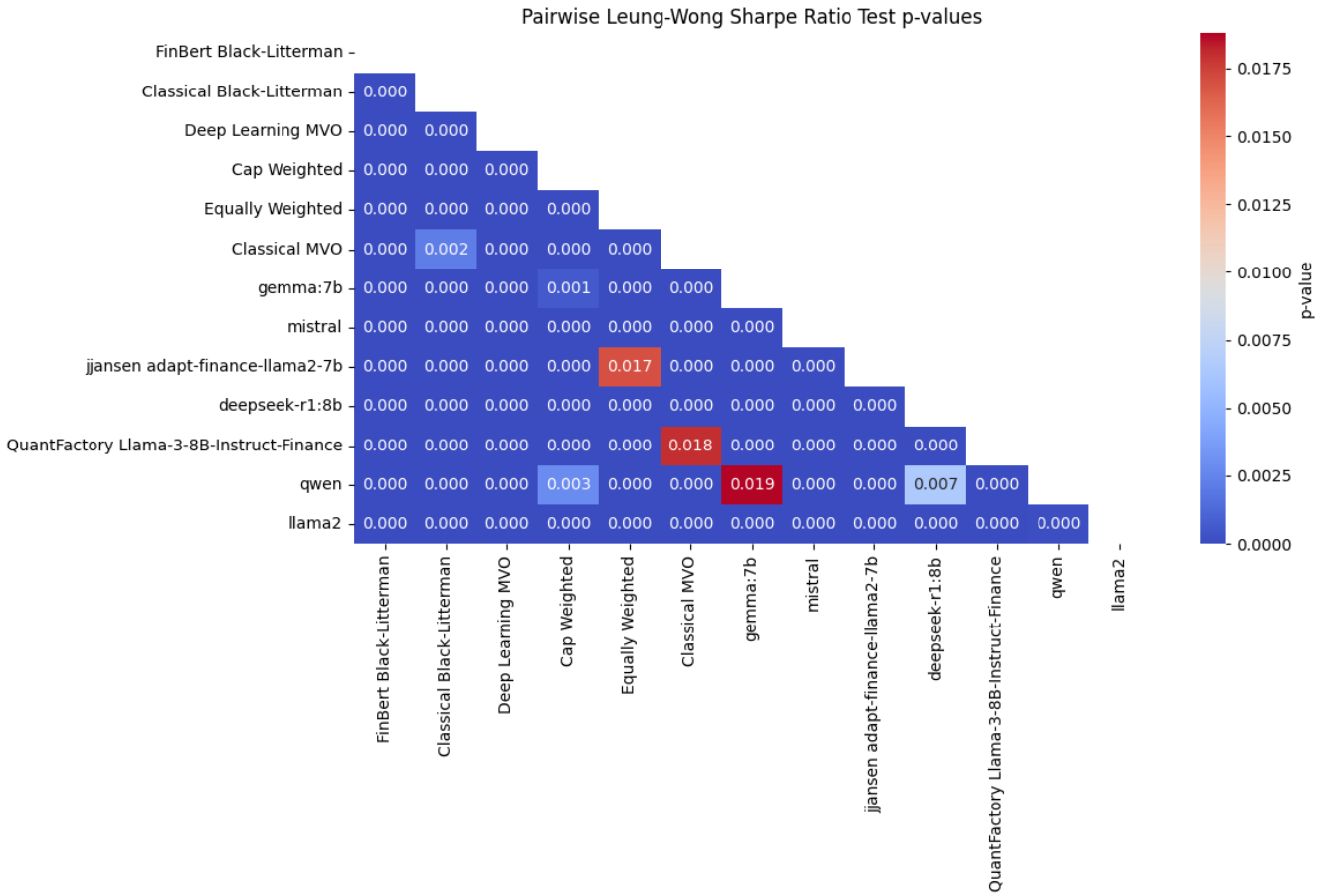


Figure 6. Leung–Wong Sharpe ratio significance test.

4.4. Discussion

The empirical results reveal a clear performance hierarchy: AI-enhanced optimizers outperformed classical methods and LLM-based portfolios, while LLM strategies consistently surpassed naive diversification. This pattern underscores the importance of explicit mathematical optimization, which LLMs currently lack. Unlike traditional optimizers that solve for efficient frontiers, LLMs rely on probabilistic text generation, leading to plausible, but occasionally suboptimal, allocations and concentration risk. This performance differential arose from fundamental limitations in LLM-based portfolio construction. First, the absence of explicit optimization mechanics means that LLMs cannot mathematically balance risk and return, resulting in allocations that may appear reasonable, but lack the rigor of mean–variance or utility-based optimization. Second, the probabilistic nature of generation introduces the risk of hallucination, producing plausible, yet inefficient, weight distributions. These shortcomings explain why even the best-performing LLM prompt strategy may underdeliver in real-world portfolio designs without quantitative grounding.

Despite these limitations, LLM portfolios demonstrate two notable strengths. First, their ability to extract contextual signals from unstructured financial text enables them to outperform simple diversification heuristics. Second, their relatively low turnover—particularly in finance-tuned models such as QuantFactory Llama-3-8B-Instruct-Finance—mitigates the transaction cost drag, preserving competitiveness in cost-sensitive environments. This suggests that, although LLM-based strategies may underperform in gross terms, their simplicity and rebalancing efficiency can partially shield them from real-world frictions. The observed turnover efficiency likely stemmed from structural factors: prompt design that emphasizes the macroeconomic context and chain-of-thought reasoning, and domain-specific fine-tuning that reduces overreactions to short-term noise. Finance-tuned

models like QuantFactory Llama-3-8B-Instruct-Finance and Jansen Adapt-Finance-Llama2-7B may have learned to prioritize consistency and resist overfitting to transient signals, yielding smoother allocation paths. Conversely, general-purpose models such as Qwen-7B exhibit highly reactive and volatile allocations, which erode the net performance once the transaction costs are applied. Ultimately, these findings highlight turnover and related transaction costs as critical differentiators in performance and suggest that any realistic evaluation of portfolio construction methods must incorporate cost-aware metrics to reflect the real-world implementability.

Beyond raw performance, the viability of LLM-driven portfolios depends on how well they align with regulatory expectations for transparency and risk governance. Frameworks such as the Dodd–Frank Act, MiFID II, and UCITS impose stringent requirements on the auditability, disclosure, and robust risk controls. While LLM-generated allocations offer interpretability advantages by removing opaque optimization layers, they introduce new compliance challenges. For example, inference-time parameters like temperature directly affect output variability; deterministic settings ($\tau \to 0$) can be enforced to guarantee reproducibility for audit trails under MiFID II. In addition, logging prompt templates, inference parameters, and model versions provide traceability for compliance audits. Explainability metrics, such as attribution scores linking allocations to specific news items, can support UCITS disclosure requirements by clarifying decision drivers. Scenario-based prompts for stress testing (e.g., risk-off macro shocks) can further align with the regulatory expectations for risk governance. Finally, maintaining version-controlled prompt libraries and model checkpoints ensures governance consistency. Our results show that low-turnover LLM portfolios can achieve competitive Sortino ratios post-costs, making them attractive in cost-sensitive contexts. However, the absence of explicit optimization logic limits their ability to satisfy regulatory-grade risk–return constraints. These findings reinforce the need for hybrid approaches that integrate LLM reasoning with quantitative optimization, ensuring strategies that balance interpretability, profitability, and compliance.

From a practical perspective, these results underscore the trade-off between interpretability and optimization rigor. LLM-driven allocations enhance transparency by removing intermediate optimization layers, but this comes at the expense of mathematical precision. While benchmarks differ in their underlying information sets, our evaluation emphasizes practical implementability over theoretical parity. To bridge this gap, future work should explore hybrid designs that integrate text and price data for more equitable comparisons. We intentionally excluded numeric features from the LLM prompts to isolate text-driven reasoning. Future research should assess hybrid prompts that combine text with minimal numeric indicators (e.g., volatility, recent returns) to improve the comparability with price-based benchmarks. In particular, blending LLM reasoning with quantitative optimization could leverage the strengths of both approaches. Additionally, scaling to larger, domain-adapted models and incorporating multi-modal signals (e.g., structured data and text) may further enhance the robustness.

5. Conclusions

This study uniquely evaluated LLMs as autonomous portfolio allocators, diverging from prior work that confined them to sentiment extraction or hybrid optimization pipelines. By prompting seven medium-sized LLMs to generate complete weight vectors without downstream optimizers, we established a novel approach for integrating natural-language reasoning into portfolio construction. Through a rigorous evaluation of seven medium-sized open-source LLMs against six benchmark strategies, including AI-optimized, classical optimization, and naive diversification methods, this research provides insights into the strengths and limitations of LLM-prompted portfolio construction. The results

reveal a clear performance hierarchy: LLM-generated portfolios generally underperformed against AI-optimized and classical benchmarks in gross risk-adjusted returns, primarily due to the absence of explicit mathematical optimization. Nevertheless, certain LLMs, such as QuantFactory Llama-3-8B, demonstrated a competitive performance relative to naïve diversification, suggesting that LLMs can produce plausible allocations, even without embedded optimization logic.

When transaction costs were considered, the performance landscape shifted significantly. High-turnover strategies, including some naïve diversification and LLM portfolios, experienced substantial degradation in net returns. In contrast, low-turnover LLMs such as Gemma-7B and Mistral-7B retained their competitiveness, even outperforming cap-weighted benchmarks post-costs. This underscores the turnover efficiency as a critical factor in real-world portfolio viability. A statistical analysis using the Leung–Wong test confirmed that the observed performance differences were statistically significant. This study was constrained by a 50-stock universe, reliance on financial news headlines, and medium-sized LLMs. The portfolios were long-only and rebalanced monthly, which may limit the generalizability. The results could differ with larger universes, alternative data sources, or multi-modal inputs.

Importantly, regulatory considerations, such as transparency and reproducibility, necessitate deterministic configurations and compliance-aware designs. Future research should advance hybrid models that integrate LLM reasoning with optimization algorithms, enabling strategies that satisfy profitability objectives while adhering to risk and regulatory standards. This contribution lays the groundwork for a new paradigm in portfolio management, bridging narrative-driven insights with quantitative rigor. Scaling to larger, domain-adapted models and integrating structured and unstructured data may enhance the robustness. Additionally, cost-aware evaluations across diverse market regimes and the use of reinforcement learning or retrieval-augmented generation could further improve the allocation quality.

Ultimately, this study bridges natural language reasoning and quantitative finance, demonstrating both the potential and limitations of LLMs as portfolio allocators. Although the current models failed to outperform traditional optimization techniques, their ability to generate stable, low-turnover allocations makes them attractive in cost-sensitive environments. As financial LLMs mature, they can redefine investment decision-making, offering strategies that harmonize data-driven precision with narrative-aware adaptability.

Author Contributions: Conceptualization, L.A.M. and J.W.M.M.; methodology, L.A.M.; software, L.A.M.; validation, L.A.M. and J.W.M.M.; formal analysis, L.A.M.; investigation, L.A.M.; resources, L.A.M.; data curation, L.A.M.; writing—original draft preparation, L.A.M.; writing—review and editing, J.W.M.M.; visualization, L.A.M.; supervision, J.W.M.M.; project administration, L.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available in the Ashraq Financial Headlines Dataset (accessed: 26 May 2025) at https://huggingface.co/datasets/ashraq/financial-news, the FNSPID Financial News Dataset (accessed: 26 May 2025) at https://github.com/Zdong104/FNSPID_Financial_News_Dataset, NewsAPI (accessed: 26 May 2025) at https://newsapi.org/, and Yahoo Finance at https://finance.yahoo.com/ (accessed: 26 May 2025).

Acknowledgments: The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Prompt engineering variants used in the LLM portfolio optimization framework.

Prompt Version	*Description*
Prompt v1	Zero-shot—no examples: *"You are a quantitative Portfolio Manager. Adjust weights to maximize Sortino Ratio and factor news."*
Prompt v2	Few-shot—2 examples (risk-on and risk-off): *"You are a quantitative Portfolio Manager. Adjust weights to maximize Sortino Ratio and factor news."* Includes examples with earnings beats and sector performance.
Prompt v3	Few-shot—6 examples that include the impact of macroeconomic news on individual companies.
Prompt v4	Few-shot—10 examples that include macroeconomic and earnings news. Includes instructions for "Chain-of-thought" reasoning.
Prompt v5	Few-shot—6 examples with macroeconomic news. Adds background context and instructions for "Chain-of-thought" reasoning and theme extraction.
Prompt v6	Few-shot—6 examples with macroeconomic news. Instructs LLMs to apply mean–variance optimization.

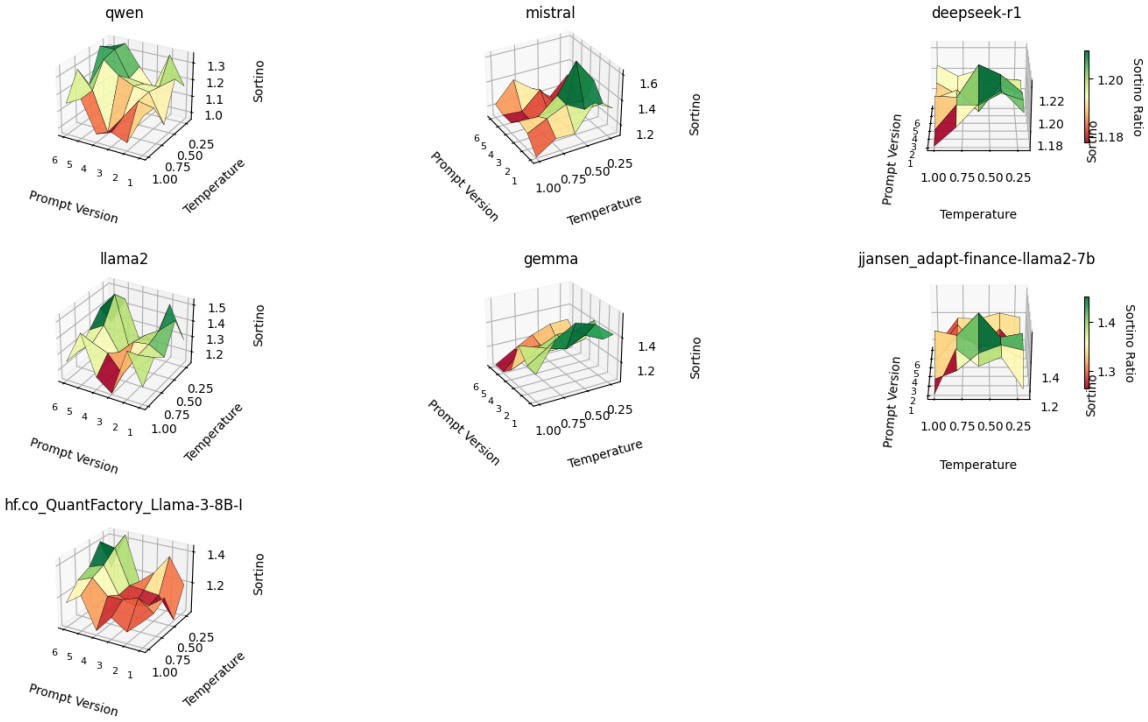


Figure A1. Performance heatmap across prompt versions and temperature settings. The color gradient represents the Sortino ratios, with darker green hues indicating higher risk-adjusted returns.

Table A2. Normalization diagnostics by model (lower percentages indicate a better raw feasibility). Percentages are computed over all assets and rebalancing dates.

Model	Non-Negative Not Feasible (%)	Sum-to-One Not Feasible (%)
Mistral-7B	0.00	1.53
QuantFactory_Llama-3-8B-Instruct-Finance	0.81	2.01
Gemma-7B	0.93	2.24
Llama2-7B	1.24	3.39
Jansen_adapt-finance-Llama2-7B	1.32	3.11
Qwen-7B	1.54	3.98
Deepseek-R1-8B	1.85	3.87

References

Agarwal, R., Singh, A., Zhang, L., Bohnet, B., Rosias, L., Chan, S., Zhang, B., Anand, A., Abbas, Z., Nova, A., & Co-Reyes, J. (2024). Many-shot in-context learning. *Advances in Neural Information Processing Systems*, *37*, 76930–76966.

Araci, D. (2019). FinBERT: Financial sentiment analysis with pre-trained language models. *arXiv*, arXiv:1908.10063.

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

Chaplot, D. S., Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., & Sayed, W. E. (2023). Mistral 7B. *arXiv*, arXiv:2310.06825. [CrossRef]

Cheng, D., Huang, S., & Wei, F. (2023, May 7–11). *Adapting large language models via reading comprehension*. The Twelfth International Conference on Learning Representations, Vienna, Austria.

Colasanto, F., Grilli, L., Santoro, D., & Villani, G. (2022). BERT's sentiment score for portfolio optimization: A fine-tuned view in Black and Litterman model. *Neural Computing and Applications*, *34*(20), 17507–17521. [CrossRef]

DeepSeek-AI. (2025). DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Deepseek*. Available online: https://arxiv.org/abs/2501.12948 (accessed on 5 April 2025).

Diao, S., Wang, P., Lin, Y., Pan, R., Liu, X., & Zhang, T. (2023). Active prompting with chain-of-thought for large language models. *arXiv*, arXiv:2302.12246.

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C. M., Chen, W., & Yi, J. (2023). Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, *5*(3), 220–235. [CrossRef]

Heng, R. Q. W., Vittori, E., Ong, K., Mao, R., Cambria, E., & Mengaldo, G. (2025). Leveraging LLMS for top-down sector allocation in automated trading. *arXiv*, arXiv:2503.09647. [CrossRef]

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. D. L., Hendricks, L. A., Welbl, J., Clark, A., & Hennigan, T. (2022). Training compute-optimal large language models. *arXiv*, arXiv:2203.15556. [CrossRef]

Huang, Z., Zhang, Z., Hua, C., Liao, B., & Li, S. (2024). Leveraging enhanced egret swarm optimization algorithm and artificial intelligence-driven prompt strategies for portfolio selection. *Scientific Reports*, *14*(1), 26681. [CrossRef]

Iacovides, G., Konstantinidis, T., Xu, M., & Mandic, D. (2024, November 14–17). *FinLlama: LLM-based financial sentiment analysis for algorithmic trading*. Proceedings of the 5th ACM International Conference on AI in Finance (pp. 134–141), Brooklyn, NY, USA.

Kirtac, K., & Germano, G. (2025, April 24–28). *Leveraging LLM-based sentiment analysis for portfolio allocation with proximal policy optimization*. ICLR 2025 Workshop on Machine Learning Multiscale Processes, Singapore.

Ko, H., & Lee, J. (2024). Can ChatGPT improve investment decisions? From a portfolio management perspective. *Finance Research Letters*, *64*, 105433. [CrossRef]

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, *35*, 22199–22213.

Luo, Y., Feng, Y., Xu, J., Tasca, P., & Liu, Y. (2025). LLM-powered multi-agent system for automated crypto portfolio management. *arXiv*, arXiv:2501.00826.

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

QuantFactory. (2024). Llama-3-8B-Instruct-Finance-RAG-GGUF. *Quant Factory*. Available online: https://huggingface.co/QuantFactory/Llama-3-8B-Instruct-Finance-RAG-GGUF (accessed on 25 May 2025).

Qwen, T. (2024). Qwen2 technical report. *arXiv*, arXiv:2412.15115.

Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M., Sakib, S., Mim, M. M. J., Ahmad, J., Ali, M. E., & Azam, S. (2024). A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*, *12*, 26839–26874. [CrossRef]

Renze, M. (2024). The effect of sampling temperature on problem solving in large language models. In *Findings of the association for computational linguistics: Emnlp 2024* (pp. 7346–7356). Association for Computational Linguistics.

Romanko, O., Narayan, A., & Kwon, R. H. (2023). Chatgpt-based investment portfolio selection. In *Operations research forum* (Vol. 4, p. 91). Springer International Publishing.

Shah, A., & Chava, S. (2023). Zero is not hero yet: Benchmarking zero-shot performance of llms for financial tasks. *arXiv*, arXiv:2305.16633. [CrossRef]

Shao, M., Basit, A., Karri, R., & Shafique, M. (2024). Survey of different large language model architectures: Trends, benchmarks, and challenges. *IEEE Access*, *12*, 188664–188706. [CrossRef]

Team, G., Anil, R., Borgeaud, S., Alayrac, J. B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., & Silver, D. (2023). Gemini: A family of highly capable multimodal models. *arXiv*, arXiv:2312.11805. [CrossRef]

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., & Bikel, D. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv*, arXiv:2307.09288. [CrossRef]

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *arXiv*, arXiv:1706.03762.

Yuksel, K. A. (2025). *AlphaPortfolio: Discovery of portfolio optimization and allocation methods using llms*. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5118317 (accessed on 6 April 2025).

Zhao, H., Liu, Z., Wu, Z., Li, Y., Yang, T., Shu, P., Xu, S., Dai, H., Zhao, L., Mai, G., & Liu, N. (2024). Revolutionizing finance with llms: An overview of applications and insights. *arXiv*, arXiv:2401.11641. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.