Language of the Markets: Deep Reinforcement Learning Approach to Algorithmic Trading with Multimodal LLM Sentiment Fusion

24159272

MSc Machine Learning Programme

Supervisors: Philip Treleaven, UCL
Dr Omer Gunes, Oxtractor

Submission date: September 2025

Abstract

This thesis presents novel research on the effect of augmenting deep reinforcement learning (DRL)-based trading agents with structured sentiment features derived from large language models (LLMs) on trading performance and risk-adjusted returns under realistic trading costs. This work builds upon the DRL-UTrans agent introduced by Yang et al. [28], which combines deep reinforcement learning with U-Net and Transformer architectures for sequential trading decisions, by systematically integrating language-derived sentiment at the feature level. An open-source LLM (`gpt-oss-120b`) is used to convert unstructured and semi-structured text into quantitative scores, ranging from -1 to 1. This thesis comprises two major pieces of work: a methodology and an experimental study, as detailed below.

Methodology: To quantify this effect, this thesis employs a rigorous ablation study designed to isolate the incremental impact of the LLM channel across contrasting market regimes.

Experiments: The study evaluates four distinct experimental configurations:

Exp I - Baseline DRL-UTrans Agent: A performance benchmark is established by reproducing and training the DRL-UTrans agent using only quantitative features derived from historical price data. This isolates the agent's effectiveness on price action alone.

Exp II - Indicator-LLM Fusion: The baseline agent is augmented with a single daily sentiment score generated by the LLM after analysing a textual summary of traditional technical indicators (RSI, MACD, etc.). This tests the ability of the agent to leverage a distilled, expert-like technical summary.

Exp III - Headline-LLM Fusion: The baseline agent is enhanced with features derived from daily news headlines. The LLM produces an individual score for each headline, and these scores are aggregated into daily statistical features, which capture the market's narrative climate.

Exp IV - Combined Fusion: The final experiment combines all available features (baseline quantitative data, indicator sentiment, and headline statistics) to evaluate the effects of providing the agent with a comprehensive view of both price and multifaceted language features.

This thesis presents the following contributions to science:

CONTRIBUTION I - A Pragmatic Pipeline for Multimodal Fusion: This dissertation introduces a novel and replicable feature-engineering pipeline that uses an LLM to translate qualitative information from news headlines and technical indicators into numerical features for reinforcement learning agents.

CONTRIBUTION II - A Systematic Quantification of Sentiment Value: The ablation study framework was designed to measure both the alpha contribution and the risk-mitigation ability of each sentiment channel, offering clear insights into the value of language-based features.

CONTRIBUTION III - An Analysis of DRL Agent Stability: This thesis provides a critical analysis of the training stability of multimodal DRL agents and highlights the challenges of run-to-run variance. It suggests that high quality sentiment features can act as a regulariser.

The code used in this project can be found at `https://github.com/shiven-taneja/dissertation`

Impact Statement

Since the inception of quantitative finance, it has been dominated by models that excel at analysing numerical data. Practitioners have utilized price history, trading volume, and their mathematical derivatives when developing algorithmic trading techniques. This paradigm is powerful and has proven itself to be highly successful. Despite this, it has ignored the vast but unstructured world of qualitative information that often precedes major shifts in asset prices. This dissertation directly addresses this gap by systematically integrating two cutting-edge technologies: Deep Reinforcement Learning (DRL), for autonomous, sequential decision-making, and Large Language Models (LLMs) for understanding and synthesizing human language. The potential impacts of this research in both academic and professional domains are significant and multifaceted.

IMPACT I - A New Paradigm for Quantitative Strategy and Risk Management: The research conducted provides a blueprint and strategic directive for practitioners in the quantitative finance industry to integrate news sentiment with DRL agents. This can lead to more sophisticated and context-aware trading systems with greater robustness, enhanced risk management, and better downside protection.

IMPACT II - Accelerating the Adoption of Advanced AI in Finance: This research provides a transparent analysis of the practical deployment challenges. This helps bridge the gap between academic potential and industry deployment by offering a realistic assessment of the steps needed to implement these models in live trading environments.

In summary, this dissertation provides a methodological framework and actionable strategic insights that can reshape how quantitative trading is approached. This research lays the groundwork for a potential paradigm shift in the industry, moving away from purely numerical models towards a new generation of language-aware systems capable of a more holistic and human-like understanding of financial markets.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Professor Philip Treleaven, whose guidance, feedback, and mentorship have been invaluable in creating this dissertation. I would also like to extend a special and sincere thank you to my external supervisor, Dr. Omer Gunes and his company, Oxtractor. Their generous provision of essential resources, including GPU access, valuable time, and an immense amount of guidance, was indispensable to the completion of this work.

On a personal note, I am extremely grateful to my family and my girlfriend. Their constant encouragement, support, and patience have been my foundation throughout this entire academic journey. This accomplishment would not have been possible without them.

I would also like to acknowledge the use of Large Language Models (LLMs) in creating this dissertation. Tools such as Gemini were used as an assistive technology for tasks such as improving grammar, editing prose, and assisting with the review, debugging, and refinement of my code.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the topic of the dissertation by establishing the motivations behind developing a sentiment-enhanced trading agent and outlines the experimentation framework employed to investigate its effectiveness. It begins by framing the research within the evolution of algorithmic trading while highlighting the gap between current quantitative models and the qualitative information present in financial news and commentary. It then defines the research objective of this dissertation and outlines the four-part experimental design, which is structured as an ablation study to isolate the incremental impact of different sentiment channels. This chapter then outlines the three key contributions of this dissertation and provides a roadmap of the structure of the rest of the thesis.

1.1 Overview of Sentiment-Aware Deep Reinforcement Learning Agents

The past two decades have drastically changed financial markets and the way trading is conducted. This is largely due to the convergence of computational power and the development of complex and sophisticated algorithms [3]. These advancements have led to the rise of algorithmic trading, where computers can execute thousands of transactions every second, and advanced risk management systems, where these algorithms can monitor global portfolios in real time. These transformations signal a fundamental shift in global financial markets towards data-driven decision-making that leverages previously untapped

information sources to gain competitive advantages in increasingly efficient markets.

Despite these advancements, these trading systems rely solely on quantitative data. Models are usually trained on historical price series, volume data, and derived technical indicators to predict market movements and execute trades. This approach completely ignores the rich qualitative information present in news articles, regulatory filings, geopolitical developments, social media posts, and market commentary. The lack of integration of this data overlooks the reality that market-moving events are often presaged in textual information sources [10].

The recent advent and rapid development of Large Language Models (LLMs) present an extraordinary opportunity to bridge this divide. For the first time, models are available that are capable of interpreting human language rapidly and at scale while also being able to convert the unstructured market narrative into a structured quantitative signal. Simultaneously, there have been massive strides in Deep Reinforcement Learning (DRL) which have created agents that excel at sequential decision-making at scale - a property that is ideal for trading in financial markets. Both of these advancements have allowed for the development of multimodal agents that can learn to trade by simultaneously "reading" the news and "watching" the charts. This intersection can create agents that possess a more holistic understanding of the market, which could lead to more robust and profitable trading strategies.

This thesis addresses the lack of research into the intersection of these domains. It directly addresses the limitations of unimodal trading models by exploring the following research question: **To what extent does augmenting a state-of-the-art Deep Reinforcement Learning trading agent with structured sentiment features, derived from Large Language Models, improve trading performance and risk-adjusted returns?** To conduct this investigation, the open-source LLM `gpt-oss-120b` was employed to generate the sentiment features. This work seeks to quantify the value of market sentiment by systematically integrating language-based sentiment features into an agent's decision-making process, using historical price data from Yahoo Finance and news headlines from a comprehensive financial news dataset.

1.2 Research Motivations

The main motivations behind this research are multifaceted, spanning practical applications in quantitative finance, the technical challenges of creating intelligent, multimodal

agents, and the economic significance of information processing in markets. The central motivation is the pursuit of a trading system that is more adaptive and informed, which could allow it to better navigate the complexities of modern financial environments.

From a practical and quantitative finance perspective, the primary motivation is the search for a persistent source of "alpha," or excess returns over the market. Research into market microstructure has shown that informed traders consistently outperform those operating solely on public price data [12]. In more efficient markets, the advantage gained from quantitative features decays as more market participants deploy similar strategies [4]. By also taking into account qualitative data, information asymmetry between market participants can be reduced. News can often act as a leading indicator that can capture shifts in investor confidence, corporate outlook, or macroeconomic trends before they are fully priced into the market. A system that is able to systematically process and act on this information could achieve a significant competitive advantage, potentially leading to improved returns and more efficient capital allocation.

Specifically, from a risk-management standpoint, augmenting agents with sentiment-aware features is motivated by the need for greater robustness. Quantitative models that are trained solely on historical price data are known for being brittle during periods of market regime shifts or exogenous shocks. Many of these events, including unexpected geopolitical developments, central bank policy shifts, or firm-specific news, can invalidate patterns learned from past data, which can lead to significant losses. An agent able to understand sentiment around these events may be able to better manage risks during these turbulent times and adapt more quickly to new market paradigms. This could lead to better risk-adjusted returns, which is a key objective in institutional finance.

The economic significance of even modest improvements in trading performance is substantial for institutional investors. When applied at scale, even a few basis points of risk-adjusted returns can lead to hundreds of millions of dollars in profits. Enhanced risk management could also provide substantial downside protection. These potential improvements justify significant research and development, as practical advances are highly valuable in the financial industry.

1.3 Research Objectives

To address the central research question introduced above, the dissertation pursues four primary research objectives. Each of these is designed to be specific, measurable, and

integral to the overall work.

1. **OBJECTIVE I - To Design and Validate a Pragmatic Multimodal Fusion Pipeline:** This objective is focused on developing an end-to-end feature-engineering pipeline, using LLMs, to translate unstructured and semi-structured text - from both news headlines and technical indicator summaries - into quantitative features. These sentiment features could then be directly integrated into a DRL agent's state representation.

2. **OBJECTIVE II - To Empirically Quantify the Value of Sentiment Features:** The goal of this objective is to systematically quantify the incremental impact of the sentiment features created. This will be measured using overall trading performance and risk metrics, including *Final Return Percentage, Compound Annual Growth Rate (CAGR), Sharpe Ratio, Sortino Ratio, and Maximum Drawdown.* An ablation study will be used to isolate the contribution of each distinct sentiment channel.

3. **OBJECTIVE III - To Validate Performance Across Diverse Market Regimes:** This objective will validate the robustness of the agent by evaluating the performance of the sentiment-augmented agent across different market conditions. These include bull markets, bear markets, and periods of high volatility across a variety of stocks. This will help determine if the value of the sentiment is context-dependent.

4. **OBJECTIVE IV - To Evaluate Production-Grade Feasibility:** The final objective aims to evaluate the practical deployment of multimodal trading systems into production-grade environments. It will evaluate computational efficiency, reproducibility, and system reliability. This involves developing and validating data engineering pipelines, including prompt engineering, result caching, and state-vector fusion, that would allow this agent to be deployed in production environments.

1.4 Research Experiments

To achieve the objectives outlined above, this thesis employs a four-part experimental design. The methodology is structured as an ablation study that allows for each of the new components to be incrementally added to the baseline system. This design allows for the incremental contribution of each source to be isolated.

1. **Experiment I - Baseline DRL-UTrans agent:** The first experiment establishes a benchmark using the DRL-UTrans agent trained and evaluated using only quantitative features derived from historical prices and volume data. This allows for the isolation of the agent's effectiveness on pure price action and provides the control against which all other experiments will be compared.

2. **Experiment II - Indicator-LLM Fusion:** The second experiment tests the value of distilled, expert-like technical analysis. A daily sentiment score is generated by the LLM from a textual summary of five technical indicators. This score augments the baseline agent and evaluates its ability to leverage a high-level summary of quantitative indicators.

3. **Experiment III - Headline-LLM Fusion:** The third experiment introduces the broader market narrative. The baseline agent is augmented with aggregated statistical features derived from the sentiment scores of daily news headlines. This experiment measures the agent's ability to react to exogenous information and capture the sentiment of the market.

4. **Experiment IV - Combined Fusion:** The final experiment integrates all the available information sources, namely the baseline quantitative data, the technical indicator sentiment and the aggregated headline features, to evaluate the comprehensive multimodal approach. This experiment is used to determine whether multiple sentiment channels provide additive benefits or whether information redundancy limits the marginal gains.

1.5 Contributions to Science

This thesis aims to make several contributions to the field of machine learning in quantitative finance. By bridging two powerful but largely disconnected methodological domains, the work attempts to provide new insights and practical tools for both researchers and practitioners. The primary contributions to science are:

CONTRIBUTION I - A Pragmatic Pipeline for Multimodal Fusion: This thesis introduces a feature-engineering pipeline that utilizes an LLM to translate qualitative information from both news headlines and technical indicators into numerical features that can be fed into a DRL agent. This is a significant contribution because it provides a modular approach for practitioners to enhance existing quantitative models without costly

architectural redesigns, lowering the barrier to entry for developing more sophisticated, market-aware agents.

CONTRIBUTION II - A Systematic Quantification of Sentiment Value: The ablation framework provides a clear and empirical measure of the marginal alpha gain and risk-mitigation properties of each distinct sentiment channel. This is important, as it moves beyond anecdotal evidence to produce data-driven conclusions about when and how language-based features add value to a quantitative trading strategy, which can inform more effective model design.

CONTRIBUTION III - An Analysis of DRL Agent Stability: This thesis not only analyses the pure performance metrics of each agent, but also provides a critical analysis of the training stability of DRL agents. The significant challenge of run-to-run variance that arises from stochastic initialisation is highlighted. This thesis presents preliminary evidence that suggests that high-quality, exogenous sentiment features may act as a regulariser that grounds the agent's policy and leads to more consistent training outcomes. This directly addresses the core issue of reliability that is paramount for any production-grade trading system.

1.6 Thesis Structure

This thesis is organised into six chapters, each providing a detailed exploration of the research problem, methodologies, experiments, results, and future implications of the study.

- **Chapter 1 - Introduction:** This chapter outlines the research motivations, core objectives and contributions while providing a roadmap for the entire thesis.
- **Chapter 2 - Background:** This chapter reviews the theoretical foundations required for this thesis. It covers Reinforcement Learning, the neural architectures used in the baseline agent and the application of LLMs in finance.
- **Chapter 3 - Methodology:** This chapter discusses the data sources, the feature engineering pipeline, the DRL-UTrans agent implementation, and the specific setup for the four research experiments.
- **Chapter 4 - Experiments and Results:** This chapter presents the empirical findings of the ablation study. It starts by establishing the baseline performance before systematically presenting the results of each sentiment-augmented agent. Finally,

it delves into a cross-experiment comparative analysis to quantify the incremental impact of each feature.

- **Chapter 5 - Discussion:** This chapter synthesizes the key findings, discusses their broader implications, addresses the limitations of the study, and proposes promising avenues for future research.
- **Chapter 6 - Conclusion and Future Work:** This chapter summarizes the research, reiterates the primary findings and contributions, and offers final concluding remarks.

Chapter 2

Background

This chapter establishes the theoretical foundations for this dissertation. It reviews relevant literature across quantitative finance and machine learning. It starts by examining algorithmic trading systems and their evolution from rule-based to learning-based systems. The chapter then analyses reinforcement learning in financial markets and trading, and even discusses the challenges of nonstationary environments and implementing realistic trading constraints. The DRL-UTrans agent is then reviewed before surveying large language models in finance. Lastly, this chapter identifies a research gap: despite advances in both domains, no existing work integrates LLM-derived sentiment features with deep reinforcement learning agents for trading.

2.1 Introduction

The landscape of financial markets has undergone a significant transformation over the past two decades. This transformation has mainly been driven by the convergence of computational power, algorithmic sophistication, and vast data availability. This has allowed computational methods in financial markets, which are broadly termed algorithmic trading, to evolve from rule-based to more sophisticated data-driven systems. Algorithmic trading systems automate the placement, sizing, and timing of orders. This advancement of algorithmic trading has been particularly accelerated by advances in deep learning, which have demonstrated unparalleled capabilities in pattern recognition, sequential decision-making, and natural language understanding. All these components are critical for successful trading strategies.

The promise of machine learning in finance extends beyond just traditional quantitative analysis. While conventional approaches have long relied on numerical indicators and statistical models, the modern financial ecosystem generates vast amounts of unstructured data. These sources include news articles, social media sentiment, earnings reports, and regulatory filings. Handling this data requires not only processing the information but also creating systems that can integrate quantitative price dynamics with qualitative market sentiment, derived from this data, to make the optimal trading decision. This chapter examines the theoretical and methodological foundations that inform this integration, which establishes the context for developing multimodal trading agents, such as the one implemented in this dissertation.

2.2 Foundations of Reinforcement Learning for Trading

Reinforcement learning has emerged as a natural framework for algorithmic trading due to its ability to model sequential decision-making under uncertainty, which is a fundamental challenge facing any trading agent. This section introduces the foundational principles of RL in algorithmic trading and looks at the application of Deep Reinforcement Learning (DRL) in this context. It then looks at the rise of RL in financial markets. Finally, it discusses the challenges of market friction and nonstationarity, which motivate our search for models that are more advanced and multimodal.

2.2.1 The Efficient Market Hypothesis Context

Before delving into the mechanics of RL, it is imperative to first frame the objective of this thesis within financial economic theory. The Efficient Market Hypothesis (EMH) was proposed by Fama in 1970 [6] and posits that asset prices fully reflect all available information. The theory is presented in three main forms:

- **Weak-form efficiency:** This posits that past market prices and data are fully reflected in securities prices. In this context, technical analysis would be of no use.
- **Semi-strong form efficiency:** All the available public information is fully reflected in securities prices in this theory. Here, neither technical nor fundamental analysis can provide traders with an edge.

- **Strong-form efficiency:** Here, all information (both private and public) is fully reflected in securities prices. Not even insider information would be able to produce excess returns.

The work presented here challenges the semi-strong form of market efficiency as it operates under the assumption that publicly available information, in the form of news headlines and market data, is not instantaneously priced into the underlying value of securities. This thesis hypothesizes that by using advanced computational models, it is possible to extract and act upon this public information more effectively than the average market participant, thereby generating "alpha" or excess risk-adjusted returns.

2.2.2 Foundational Concept: Trading as a Markov Decision Process (MDP)

Trading must first be defined as a Markov Decision Process (MDP) which provides the mathematical foundation for applying reinforcement learning to financial markets. An MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ represents the state space, $\mathcal{A}$ the action space, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ the state transition probability function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function, and $\gamma \in [0, 1]$ the discount factor [21]. A graphical representation of the MDP is shown in Figure 2.1.

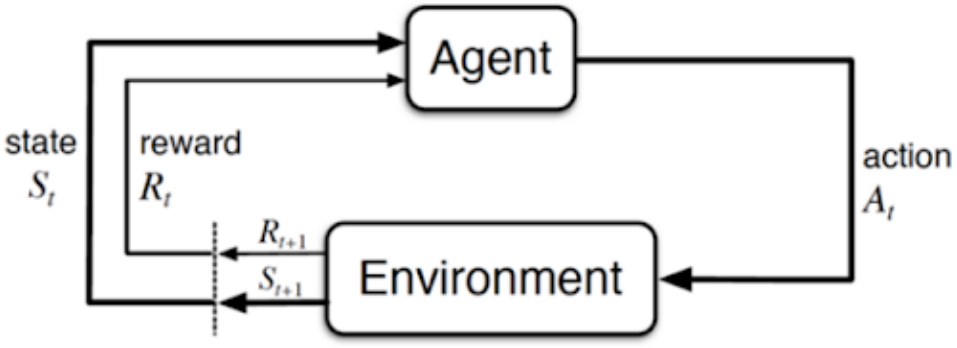


Figure 2.1: Markov Decision Process [26]

The agent's objective is to learn an optimal policy $\pi^* : \mathcal{S} \to \mathcal{A}$ that maximizes the expected cumulative discounted reward. The value function under policy π is defined as:

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right] \tag{2.1}$$

The action-value function, which represents the expected return when taking action a in state s and following policy π thereafter, is:

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[R_{t+1} + \gamma V^{\pi}(S_{t+1}) \mid S_t = s, A_t = a\right] \tag{2.2}$$

These functions satisfy the Bellman equations, which form the basis for iterative solution methods:

$$V^{\pi}(s) = \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}} P(s'|s,a)[R(s,a,s') + \gamma V^{\pi}(s')] \tag{2.3}$$

$$Q^{\pi}(s,a) = \sum_{s'\in\mathcal{S}} P(s'|s,a)[R(s,a,s') + \gamma \sum_{a'\in\mathcal{A}} \pi(a'|s') Q^{\pi}(s',a')] \tag{2.4}$$

The optimal value functions V^* and Q^* satisfy the Bellman optimality equations, which provide a recursive definition of the maximum possible return:

$$V^*(s) = \max_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} P(s' \mid s,a)\left[R(s,a,s') + \gamma V^*(s')\right] \tag{2.5}$$

$$Q^*(s,a) = \sum_{s'\in\mathcal{S}} P(s'|s,a)[R(s,a,s') + \gamma \max_{a'} Q^*(s',a')] \tag{2.6}$$

Solving these equations directly is often intractable in complex environments, which motivates the use of iterative, sample-based DRL algorithms to approximate the optimal Q-function, Q^*

Trading-Specific MDP Components

In the context of financial trading, these abstract mathematical components map to concrete market elements:

State Space ($\mathcal{S}$): The state $s_t \in \mathcal{S}$ encapsulates all the market information at time t. It contains all the necessary information for the agent to make the optimal decision. The design of the state space is critical to allow the agent to have a complete understanding of the environment, including (i) portfolio information (cash balance, current holdings), (ii) price history (Open, High, Low, Close and Volume (OHLCV) bars over a lookback window), and (iii) technical indicators [2].

Action Space ($\mathcal{A}$): The action space contains all of the decisions the agent can make. The simplest single-asset trading agents have an action space that consists of three actions:

$a_d \in \{\text{buy}, \text{sell}, \text{hold}\}$. This is expanded in more complex agents to include the quantity that is bought/sold. This is usually implemented by expanding the single discrete action space or by introducing a continuous one. The DRL-UTrans agent employs a hybrid action space with discrete decisions $a_d \in \{\text{buy}, \text{sell}, \text{hold}\}$ and a continuous position size $w \in [0, 1]$, enabling granular control over trade execution [28].

Reward Function (R): The reward function is the fundamental component of the RL process as it provides the feedback signal to the agent that then guides its learning. The way in which the reward function is designed ("Reward Engineering") dictates the trading objective of the agent. The simplest reward functions simply use the profit and loss (PnL) time-step $R_t = p_{t+1} - p_t$, where p_t can represent the portfolio value or price of a stock at time t. In order to encode more sophisticated trading strategies, that may take into account other factors such as risk, one needs to incorporate other risk-adjusted metrics, such as the Sharpe ratio, and penalize trades by incorporating transaction costs in order to better simulate real-world trading conditions.

2.2.3 Inherent Challenges: Market Friction and Non-Stationarity

The algorithms discussed have become extremely sophisticated. Despite this, real-world applications in the trading market are filled with challenges. Two of the main ones are:

1. **Market Friction:** These are the costs and the constraints that come along with operating in a trading environment. These mainly refer to the transaction costs (commission) and slippage (the difference in the price between the actual and expected execution price). Sophisticated DRL agents account for these costs. They usually do this by penalizing the reward function for each transaction which allows the agent to learn the optimal strategy despite these market frictions.

2. **Non-Stationarity:** The statistical properties of financial markets, such as volatility, correlation, and trends, are constantly changing which makes it extremely difficult to extract patterns in the markets. Agents that are trained on one regime (e.g. a low-volatility bull market) will likely perform badly on a completely new one (e.g. a high-volatility bear market). The MDP formulation assumes markets are stationary and these regime changes violate this. This is likely the hardest challenge to overcome.

This struggle to overcome the stationarity assumption exposes a major limitation with models that rely only on historical prices and volume data. Models relying solely on

technical indicators can be slow to adapt to changing market conditions as these indicators are often seen as lagging indicators of price data. This provides strong motivation to augment the state of the agent with alternative data. This forms the central theoretical justification for this dissertation's proposal to incorporate features derived from language and news sources into the DRL agent's state.

2.3 Deep Reinforcement Learning Algorithms in Finance

2.3.1 A Survey of Reinforcement Learning Algorithms in Trading

RL in trading environments was first brought to light by the work of Nevmyvaka et al. [18] who demonstrated that RL was a viable approach for trade execution using NASDAQ millisecond data. Their work introduced the low-impact state space factorization that became foundational for optimal order execution. This study worked to establish that RL agents could learn to minimize market impact while executing large trade orders. Their approach outperformed traditional volume-weighted average price (VWAP) strategies, which were the gold standard at the time.

Building on this foundation, Zhang, Zohren, and Roberts [30] presented one of the earliest and most comprehensive applications of DRL to financial markets. They trained three RL agents using Deep Q-learning Network (DQN), Policy Gradients, and Advantage Actor-Critic on the 50 most liquid futures contracts from 2011 to 2019. Their study considered both discrete and continuous action spaces and incorporated volatility scaling, allowing position sizes to adapt to market conditions. The learned policies were able to follow sustained trends while reducing exposure during consolidation phases. Their approaches outperformed classical time-series momentum benchmarks after accounting for transaction costs, which underscored DRL's viability for systematic trading.

Recent advances have greatly expanded the scope, sophistication, and ease of implementation of RL applications in finance. One of the largest drivers of this advancement was the introduction of FinRL by Liu et al. [15] which is an open-source and comprehensive library that standardized DRL research in the finance space. It allows users to implement all of the major DRL algorithms (DQN, DDPG, PPO, SAC, A2C, TD3) across multiple market environments. The authors conducted a comparative study of all these methods,

which revealed that A2C achieved superior cumulative rewards compared to the other algorithms. This provided crucial benchmarking for subsequent research. The following section provides a more detailed examination of DRL and these algorithms.

2.3.2 Deep Reinforcement Learning

DRL has grown to prominence for stock trading as it combines Deep Learning, by using deep neural networks as function approximators for the policy and/or value function, with RL, enabling scalability to high-dimensional financial state spaces. This section delves into the mathematical foundations of the key DRL model used in this thesis, Deep Q-Networks (DQN) and briefly surveys alternative approaches.

Value-Based Methods: Deep Q-Networks (DQN)

Value-based methods, such as Q-Learning and DQN, are particularly well-suited for financial trading problems with discrete action spaces. The objective of these methods is to learn the optimal action-value function $Q^*(s, a)$, which allows the agent to select the action with the greatest long-term reward in any state. This direct optimization is intuitive for financial tasks where there is a clear goal, like portfolio value, that is trying to be optimised.

DQN was the pioneering algorithm that demonstrated DRL could master complex decision-making tasks [17]. It uses experience replay and target networks in order to address the instability of combining Q-learning with neural networks.

The Q-learning update rule forms the foundation:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right] \tag{2.7}$$

DQN approximates the Q-function using a neural network with parameters θ:

$$Q(s, a; \theta) \approx Q^*(s, a) \tag{2.8}$$

The loss function minimizes the temporal difference error:

$$L(\theta) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right] \tag{2.9}$$

where $\mathcal{D}$ is the experience replay buffer and θ^- is the parameters of the target network, updated periodically as $\theta^- \leftarrow \theta$.

Experience Replay: Transitions (s_t, a_t, r_t, s_{t+1}) are stored in a buffer $\mathcal{D}$ and sampled randomly for training, breaking temporal correlations and improving sample efficiency:

$$\mathcal{D} = \{(s_1, a_1, r_1, s_2), ..., (s_N, a_N, r_N, s_{N+1})\} \tag{2.10}$$

Target Network: A separate network with parameters θ^- provides stable Q-value targets, updated periodically:

$$\theta^- \leftarrow \tau\theta + (1-\tau)\theta^-, \quad \tau \in [0, 1] \tag{2.11}$$

For trading applications, DQN's discrete action space naturally maps to buy/sell/hold decisions. However, limitations include overestimation bias and the inability to handle continuous position sizing without discretization. Extensions like Double DQN [9] and Dueling DQN [27] address limitations such as overestimation bias and have further improved the stability and performance of value-based agents.

2.3.3 Alternative Approaches: Policy Gradient and Actor-Critic Methods

While value-based methods are effective, there are other families of DRL algorithms that are widely used in financial applications.

Policy Gradient Methods, such as Proximal Policy Optimization (PPO) [23], directly learn a stochastic policy $\pi_\theta(a, s)$. Here, instead of learning a value function, it optimizes the policy parameters θ in order to maximize the expected return. PPO is specifically known for its stability and reliability, which it achieves by constraining policy updates within a trust region.

Actor-Critic Methods, such as Soft Actor-Critic (SAC) [8] and Asynchronous Advantage Actor-Critic (A3C) [16], combine the strengths of both DQN and PPO. They have two networks; one is the "actor" which learns the policy, and the other is the "critic" which is used to evaluate the actions by learning the value function. This structure can lead to more stable and efficient learning, particularly in environments with continuous action spaces (e.g. for position sizing). SAC is especially known for its use of entropy regularization, which encourages exploration and can prevent the agent from converging to suboptimal strategies.

While these actor-critic networks offer powerful alternatives, the DQN framework has extensively proven its effectiveness, and its direct value maximization approach makes it a

strong foundation for the DRL-UTrans agent and this research.

2.4 Advanced Architectures for Sequential Price Data Analysis

Having surveyed the primary family of DRL algorithms, the discussion now turns to the neural network architectures that provide the representational power for the agents. A DRL agent's performance is largely based on the ability of the underlying neural network to extract patterns and features from the raw input data. This section will explore the underlying architectures which are used to do this and will introduce the baseline for this study: the DRL-UTrans agent.

2.4.1 Neural Network Architectures for Financial Time Series

The ability of modern DRL agents to extract meaningful patterns from complex financial data is due to the neural networks underpinning their architecture. This section delves into the mathematical foundations and architectural innovations of Transformer networks and U-Nets. They will specifically be looked at in the context of sequential price modelling.

Transformer Architecture: Attention Mechanisms for Financial Data

The Transformer architecture, which was introduced in the revolutionary paper "Attention is All You Need" by Vaswani et al. (2017) [25], revolutionized sequence modeling through self-attention mechanisms. These models were able to capture long-term dependencies, without recurrent units. The scaled dot-product attention is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.12}$$

where $Q \in \mathbb{R}^{n \times d_k}$, $K \in \mathbb{R}^{m \times d_k}$, and $V \in \mathbb{R}^{m \times d_v}$ represent queries, keys, and values respectively. The scaling factor $\sqrt{d_k}$ prevents vanishing/exploding gradients in the softmax function.

Multi-head attention enables the model to attend to different representation subspaces simultaneously:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \tag{2.13}$$

where each head is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.14}$$

with learned projection matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$.

For financial applications, the attention mechanism enables the model to dynamically weight the importance of historical price points, capturing both momentum effects and mean reversion patterns. The positional encoding ensures temporal ordering is preserved:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \tag{2.15}$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \tag{2.16}$$

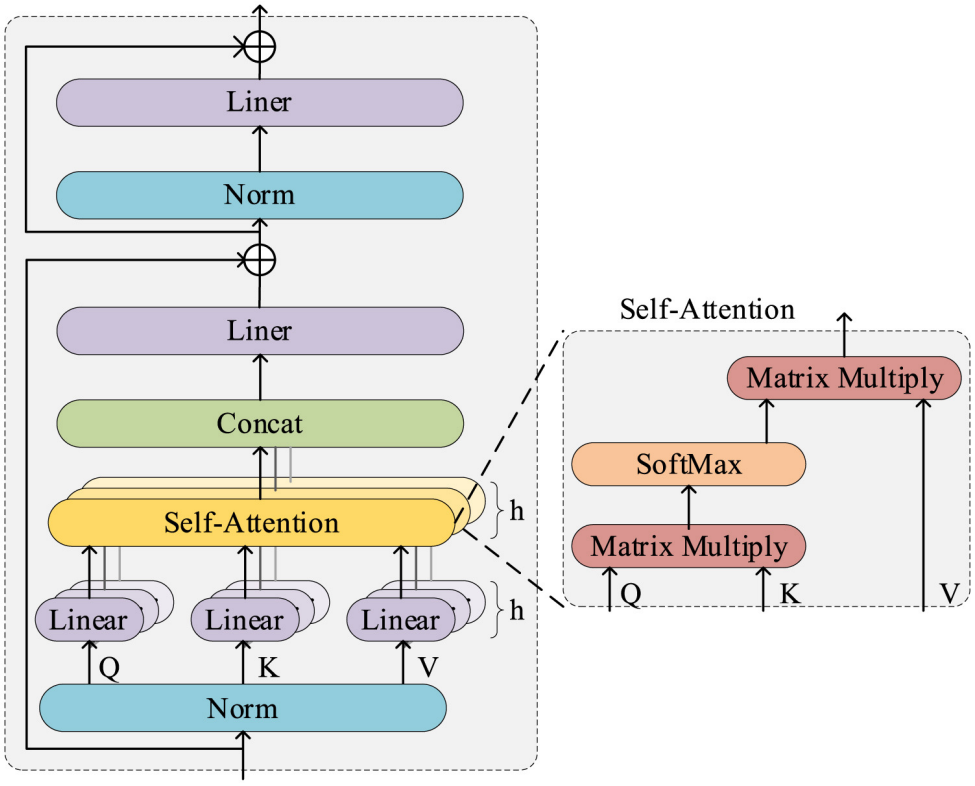


Figure 2.2: Internal structure of the Transformer Layer [28]

U-Net Architecture: Multi-Scale Feature Extraction

The U-Net architecture, originally developed for biomedical segmentation [22], excels at capturing multi-scale temporal patterns in financial data. The encoder-decoder structure with skip connections preserves both high-frequency trading features and long-term trends.

The encoder progressively downsamples the input through convolutional layers:

$$h_l = \text{ReLU}(\text{Conv}_{k\times k}(h_{l-1})) \tag{2.17}$$

followed by max pooling:

$$h'_l = \text{MaxPool}_{2\times 2}(h_l) \tag{2.18}$$

The decoder upsamples while concatenating skip connections:

$$d_l = \text{ReLU}(\text{Conv}_{k\times k}([\text{UpConv}(d_{l+1}), h_l])) \tag{2.19}$$

where $[\cdot, \cdot]$ denotes concatenation along the channel dimension.

This architecture's ability to preserve fine-grained details while capturing global context makes it particularly suitable for identifying chart patterns, support/resistance levels, and other technical indicators that operate across multiple timescales.

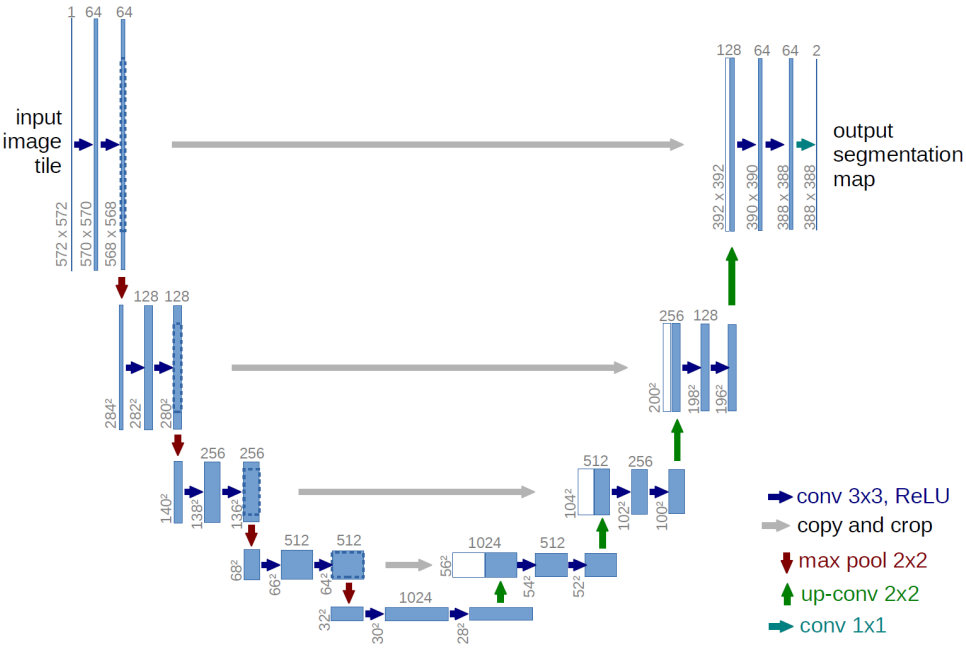


Figure 2.3: The original U-Net architecture [22]

2.4.2 The DRL-UTrans agent

The DRL-UTrans architecture, which was proposed by Yang *et al.* (2023) [28] combines a U-Net + Transformer and uses this as the backbone for a DRL agent for single-stock trading. The core idea behind the model is that it allows the UTrans network to jointly decide **(i)** what to do {buy, sell, or hold} and **(ii)** the quantity to transact via a continuous action weight $w \in [0, 1]$. This action weight allows the agent to scale the position size.

Feature extraction (UTrans network)

A sliding window of historical prices and technical indicators is fed into the UTrans network. This is first processed by the U-Net encoder, which progressively downsamples the input, allowing the model to capture the multi-scale temporal patterns. The downsampled latent representation of the input passes through a Transformer layer whose multi-head self-attention captures the long-range dependencies across the sliding window. Finally, the decoder upsamples the features, while skip-connections from the corresponding encoder levels are added to it in order to recover high-resolution details. The network ends with two lightweight heads:

- a **categorical head** (softmax) that outputs the discrete action $A_t \in \{\text{buy}, \text{sell}, \text{hold}\}$;
- a **continuous head** (sigmoid) that outputs the action weight w_t.

These two values represent the agent's action at each time-step.

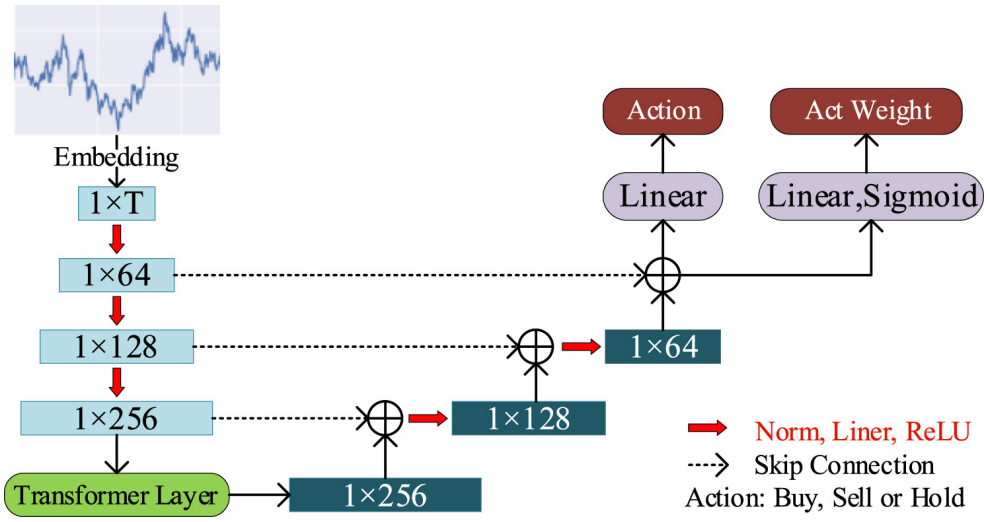


Figure 2.4: The UTrans feature-extractor with dual outputs (from [28]).

From supervised backbone to DRL agent

To stabilise the learning, two identical UTrans networks are instantiated using a DQN-based training algorithm:

1. the **policy network** which produces the trading decision that is then fed into the environment
2. the **target network** is a lagged copy of the policy network, which provides a fixed bootstrapped target that is softly updated only every C steps ($\theta_t \leftarrow \theta_p$).

After an action is executed by the agent, the environment returns a reward R_t, quantifying the effectiveness of this move, along with the next state. A tuple of the transition $(S_t, A_t, w_t, R_t, S_{t+1})$ is appended to a replay memory of fixed capacity. This is used to train the network by sampling a mini-batch from the replay memory, which is then used to perform gradient descent.

Reward design

The reward introduced in the paper is position-aware. When a *buy* or *sell* action is taken, the reward is based on the realised profit or loss on the executed shares only. For a *hold* action, it marks to market the current position, allowing learning signals even in flat markets. The authors implemented a rule where all of the trades must be executed in lots of 100 shares. The reward function R_t is given below where C_t is the transaction cost price, P_t is the price of the stock, B_t is the amount bought, S_t is the amount sold, and H_t is the amount held.

$$R_t = \begin{cases} (C_t - P_t)\ B_t, & \text{if action = 'buy'}, \\ (P_t - C_t)\ S_t, & \text{if action = 'sell'}, \\ (P_t - C_t)\ H_t, & \text{if action = 'hold'}. \end{cases} \tag{2.20}$$

This design of the DRL-UTrans network enables risk-controlled trading by allowing the agent to express confidence levels in its decisions. This allows it to reduce its position sizes during periods of uncertainty. This model was validated against seven baseline approaches and was able to outperform all of them. It achieved higher returns while also having lower investment risk than six of the baseline agents. The agent particularly performed well in periods of volatile markets and crashes. This is likely due to its reward function, which is volatility-sensitive.

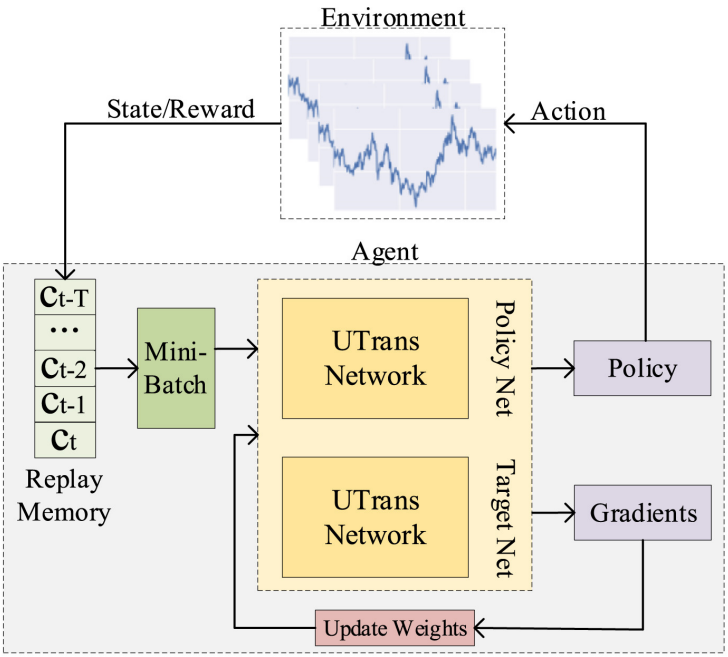


Figure 2.5: Full DRL-UTrans agent showing the policy/target networks, replay buffer and training loop (from [28]).

Justification for Use as a Baseline

There were several key factors that justified the choice of the DRL-UTrans agent as the baseline. Firstly, it represents a state-of-the-art architecture that is specifically designed for time-series data. It has demonstrated superior performance against multiple benchmarks in the original publication. Secondly, its hybrid architecture, which effectively captures both short-term, high-frequency patterns (via U-Net) and long-range temporal dependencies (via Transformer), provides a robust and challenging foundation. Using and building upon such a strong baseline agent allows the true incremental impact of adding qualitative sentiment features to be observed while ruling out that observed gains are merely due to a weak baseline.

Limitations of the Baseline Agent

While the DRL-UTrans agent has been shown to be a powerful agent, it is not without potential limitations. The agent uses a fixed-size lookback window for its input into the UTrans network which may limit the ability for the agent to capture macro-level regime shifts over longer time horizons than the window. Furthermore, the agent's architectural complexity demands significant computational resources for training and backtesting. The

original paper used 2x NVIDIA GeForce RTX 2080 Ti (11 GB) graphics cards to train the agent. It may also be prone to overfitting on specific market regimes without careful regularization and hyperparameter tuning. While its reward mechanism is sensitive to volatility, it could potentially reward inaction during strong trends.

2.5 Large Language Models in Financial Applications

Parallel to advances in reinforcement learning, the financial sector has witnessed a revolution in natural language processing through the introduction and rapid development of Large Language Models (LLMs). This section examines the evolution, methodologies, and applications of LLMs and their ability to extract actionable features from textual financial data.

2.5.1 Evolution from Domain-Specific to General-Purpose Models

Early work in financial NLP relied on domain-specific adaptations. One of the earliest, and most widely used models, is FinBERT which was introduced by Araci in 2019 [1]. This was a BERT variant that was specifically fine-tuned on financial corpora and was able to achieve 97% accuracy on Financial PhraseBank, a 15-percentage-point improvement over previous methods. This demonstrated the value of domain-specific pre-training for better capturing financial terminology and context.

The emergence of larger and more powerful general models shifted this paradigm. Recent studies have compared GPT-4 class models and found that, when properly prompted, they were able to exceed FinBERT's performance without further domain-specific training or fine-tuning. [24]. This shift has profound implications for production systems, eliminating the need for maintaining and creating specialized models.

2.5.2 Sentiment Extraction Methodologies

Various techniques can be used for financial sentiment extraction using modern LLMs:

Prompt Engineering: Zero-shot and few-shot prompting strategies enable consistent and simple sentiment scoring. A prompt using a template along the lines of the following is fed into the model:

> Analyze the financial sentiment of the following text. Task: Score the directional sentiment for the named instrument over the next few trading days. Return ONLY valid JSON: {{"score": <float in [-1, 1]>}}

Instruction Tuning: Recent studies, such as that performed by Zhang et al. (2023) [29], demonstrated that instruction-tuned models had the ability to significantly outperform base models in numerical reasoning tasks, which are crucial for earnings analysis and quantitative assessments.

Retrieval-Augmentation Generation (RAG): RAG approaches combine LLMs with external knowledge bases. This allows for real-time incorporation of market context and historical patterns [13].

2.5.3 Integration Challenges and Production Considerations

There are certain challenges that are present with deploying LLMs in production trading systems:

Latency Constraints: Financial markets require sub-second response times. In order to meet these challenges, current solutions incorporate model distillation, caching strategies, and edge deployment.

Consistency and Reliability: Stochastic generation can produce varying sentiment scores even with identical inputs. Production systems implement ensemble voting, temperature control, and deterministic decoding to try to bolster reproducibility.

Temporal Alignment and Lookahead bias: One of the critical challenges is ensuring that the temporal integrity of information is kept. One of the main obstacles is ensuring that the sentiment features are aligned with the price data. This requires sophisticated timestamp matching and aggregation strategies to ensure there is no lookahead bias at any of the data points. A second, more subtle but equally important issue is that of using pre-trained LLMs where they possess implicit knowledge of world events up to their training cutoff date. The model used in this thesis, the gpt-oss-120b, was trained on data up until 2025, and while analysing a headline from 2022, it has the potential of processing it with "knowledge of the future," which can contaminate the backtest and lead to unrealistically optimistic results. This study acknowledges this inherent limitation and thus the results should be looked at as an upper-bound representation of the performance achievable with sentiment features.

2.5.4 Empirical Evidence and Performance Metrics

Recent studies have begun integrating LLMs into trading studies and have provided compelling evidence for their implementation. A study by Jiang et al. (2025) [11] combined FinBERT sentiment with LSTM price prediction, which achieved a 23% improvement in directional accuracy. Another study conducted by Fatouros et al. (2023) [7] demonstrated that ChatGPT-derived features were able to achieve a 36% higher correlation with market returns compared to traditional sentiment dictionaries.

Despite these exploratory studies, critical limitations still persist. LLMs struggle with understanding novel market-moving events that are outside their training data. They have also been found to exhibit bias towards bullish sentiment and could possibly amplify market noise rather than extract genuine signals. These limitations motivate the careful ablation and integration approach proposed in this thesis. This allows for the isolation of the sentiment scoring to be studied, where LLM-derived features augment rather than replace price-based features.

2.6 Identifying the Research Gap

Great strides have been made in both RL and LLM domains, especially in financial applications. Despite this, there lies a critical gap in the intersection of the two.Analysis of the current literature shows that sophisticated DRL agents are capable of learning complex trading strategies from price data. Powerful LLMs have also shown promise in sentiment extraction from financial text. These two modalities have evolved almost exclusively in parallel, and there has been limited exploration of their direct integration in a multimodal trading system.

In the current multimodal approaches discussed, namely the Cross-Modal Temporal Fusion framework [19] and the Stock Movement Prediction with Multimodal Stable Fusion system [31], they have shown consistent improvement ($\sim$5-30%) over single-modality baselines. It was discovered in these studies that text sentiment contributed to more predictive power than first anticipated. These studies solely focused on predictive tasks rather than sequential decision-making tasks, and none have integrated language-derived sentiment into the state space representation of an RL agent.

This gap in the direct fusion of DRL trading agents and LLM sentiment analysis presents a significant research opportunity. The DRL-UTrans framework established a strong foundation for processing historical price sequences; however, it currently operates

on quantitative market data alone. This completely ignores the features that exist solely in the rich qualitative information available through news sentiment, market commentary, and financial reports. Conversely, LLM-based sentiment analysis approaches have proven highly effective for financial predictive tasks; these models have operated mainly in an isolated environment, without the integration into sequential decision-making frameworks which can utilize their features.

This thesis addresses this gap by proposing a multimodal architecture that enhances the DRL-UTrans framework by integrating LLM-derived sentiment features. This research examines whether a sentiment-aware RL trading agent can achieve greater risk-adjusted returns compared to its price-only counterparts, when language-based features are integrated into its state representation. This investigation represents a novel contribution to both the RL and NLP literature in the financial domain. This bridges two powerful, but previously disconnected, methodological domains to attempt to create a more sophisticated and market-aware trading system.

Chapter 3

Methodology

This chapter outlines the methodology used in all four investigations. It begins by detailing the data sources, collection procedures and the stock selection criteria. The chapter then outlines the multimodal feature engineering pipeline, which includes the generation of the baseline technical indicators, and the two-pronged approach for generating sentiment features from news headlines and technical indicator summaries. The chapter then specifies the implementation of the DRL-UTrans agent architecture and the training algorithm. The simulated trading environment is then described and the chapter concludes by discussing the four-part ablation study, the training protocol and the evaluation metrics used to assess performance.

3.1 Data Sources and Collection

Data quality and breadth are the foundations of every financial machine learning system. This research uses two distinct types of data: quantitative market data and qualitative text data. This provides the agent with a holistic view of the market environment.

3.1.1 Financial Price Data

All financial price data in this thesis was sourced from Yahoo Finance using the `yfinance` Python library [20]. This library provides daily Open, High, Low, and Close (OHLC) prices along with trading volume. To ensure consistency, all price data is adjusted using the library's `auto_adjust = True` parameter, which mitigates the impact of corporate

actions such as stock splits and dividend payments. This provides a clean and continuous price series that is suitable for calculating technical indicators and for evaluating portfolio performance. All data were cached locally to maintain consistency across experiments.

3.1.2 News Headline Dataset

News headlines were sourced from the FNSPID (Financial News and Stock Price Impact Dataset) which is a large-scale corpus that contains over 1.2 million headlines for stocks listed in the S&P 500 [5]. The data were stored and queried from a local PostgreSQL database. For each entry in the `fnspid_news` table, three columns were extracted: the stock ticker (`stock_symbol`), the headline text (`headline`), and the publication date (`date`). This dataset provides the qualitative information used to generate sentiment scores with a Large Language Model (LLM).

3.1.3 Stock Selection Criteria

A universe of ten stocks from diverse sectors was selected to test the methodology across different market dynamics. This selection process was governed by two primary constraints to ensure the validity and generalisability of the results:

1. **Data Richness:** Each stock that was selected needed to have a minimum of 8000 associated news headlines. This was to ensure there was a sufficiently dense textual signal for sentiment analysis.

2. **Historical Depth:** A minimum of five years of continuous price data was required. This ensures the training and testing periods encompass various market regimes (e.g., bull, bear, and volatile markets) and provides sufficient data for model training.

The universe of stocks chosen, and their key statistics are detailed in table 3.1. A 70/30 temporal split was enforced in order to partition the data into train and test sets. The first 70% was used to train the agent while the last 30% is reserved for out-of-sample evaluation. This strict temporal separation prevents lookahead bias and simulates more realistic trading scenarios.

Table 3.1: Stock Universe and Dataset Characteristics

Ticker	Company	Headlines	Date Range	Test Start Date
BABA	Alibaba Group	11,625	2014-2023	2021-04-05
GE	General Electric	8,680	2012-2023	2020-06-26
GOOG	Alphabet Inc.	9,930	2018-2023	2022-06-15
KO	The Coca-Cola Co.	10,521	2009-2023	2019-08-21
MRK	Merck & Co.	10,774	2009-2023	2018-08-30
MS	Morgan Stanley	9,458	2010-2023	2019-10-23
NVDA	NVIDIA Corp.	11,862	2011-2023	2020-02-25
QQQ	Invesco QQQ Trust	11,813	2011-2023	2020-02-28
T	AT&T Inc.	9,463	2016-2023	2021-09-24
WFC	Wells Fargo & Co.	11,301	2011-2023	2020-05-20

3.2 Feature Engineering Pipeline

Transforming the raw data into a structured feature set that the DRL agent can interpret is one of the most critical stages of the methodology. The pipeline used in this thesis was designed to create a rich, multimodal state representation by combining traditional quantitative indicators with the LLM-derived sentiment features. The entire feature engineering pipeline is depicted in figure 3.1

3.2.1 Baseline Technical Indicators

To establish a quantitative baseline, a set of four widely used technical indicators was calculated from the historical OHLCV data. These indicators were chosen to capture different aspects of price dynamics, including momentum, trend, and volatility:

- **Moving Average Convergence Divergence (MACD):** A trend following momentum indicator.
- **Stochastic Oscillator (KDJ_K):** A momentum indicator that compares the closing price of a security to its prices over a certain period of time.
- **Open-Close Difference:** A measure of the intra-day price movement.
- **Relative Strength Index (RSI_14):** A momentum oscillator that measures both the speed and change of price movements over a 14-day period.

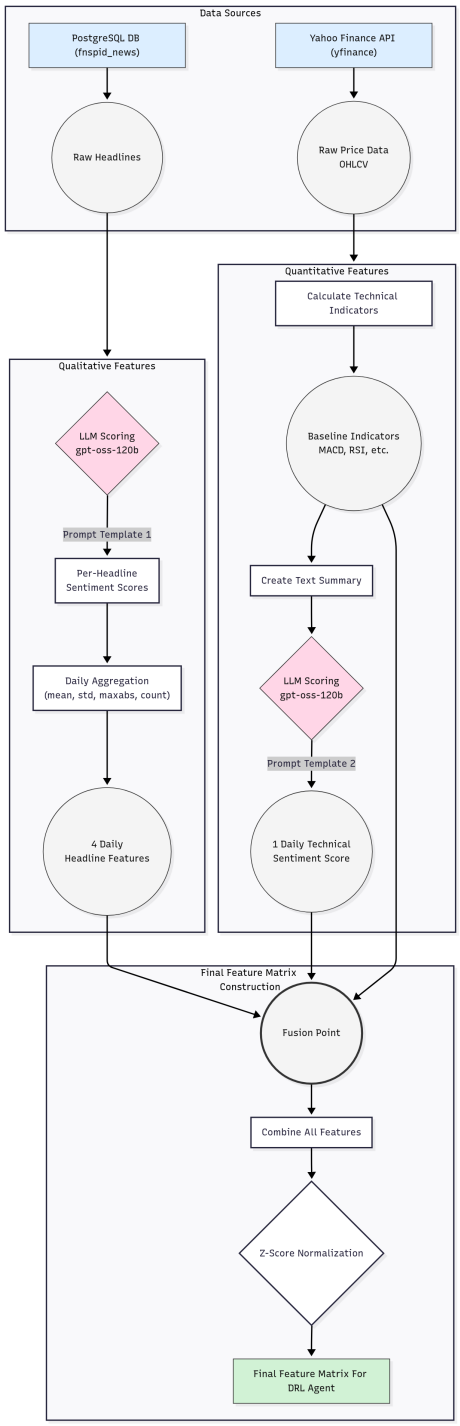


Figure 3.1: The end-to-end feature engineering pipeline, illustrating the parallel processing of quantitative price data and qualitative textual data, which are then fused and normalized to create the final feature matrix for the DRL agent.

3.2.2 LLM-Based Sentiment Scoring

One of the central contributions of this work is the systematic conversion of unstructured text into quantitative sentiment features. This is achieved using the open-source LLM `gpt-oss-120b`, which was accessed using the Groq API. When prompting the model, a `temperature` of 0.0 was used in order to ensure deterministic and reproducible outputs. To handle the high volume of requests, robust engineering practices were implemented, including incremental processing, local caching of all LLM responses, and a retry mechanism with exponential backoff.

Two distinct sentiment channels were created through intricate prompt engineering.

1. Headline Sentiment Scoring: Each individual headline was scored for its potential impact on the associated stock price over the next few days. The model was instructed to return a valid JSON with a single element containing a `score` key with a floating point value between -1.0 and 1.0. The exact prompt template that was used is shown below.

```
You are a finance assistant. Return only valid JSON.
You will be given a single financial news headline about
a public company or ETF.
Task: Score the directional sentiment for the named
instrument over the next few trading days.
Return ONLY valid JSON: {{"score": <float in [-1, 1]>}}

Headline: "{headline}"
```

2. Technical Indicator Sentiment: This channel was used to create a high-level qualitative interpretation of a quantitative state. For each trading day, a textual summary of the technical indicators was created. This summary was then passed into the LLM which was instructed, similarly to above, to return a single sentiment score between -1.0 and 1.0. This process distils the state of the asset's technical indicators into a single, expert-like judgment. The prompt template used is as follows.

```
You are a quantitative trading assistant. Return ONLY
valid JSON with keys 'score' and 'rationale'.
We summarize the current technical state of a security.
```

```
Provide a directional sentiment score in [-1, 1] for
the next 1-5 trading days.
Use the summary; do not invent facts.
Return ONLY valid JSON: {{"score": <float in [-1, 1]>}}

Ticker: {ticker}
Date: {date}
Summary: {summary}
```

3.2.3 Feature Aggregation and Normalization

Since multiple headlines can occur on a single day, raw sentiment scores were aggregated into daily statistical features. Four features were created to describe the daily news landscape:

- `news_sent_mean_1d`: The average sentiment score of all the headlines for the day.
- `news_sent_std_1d`: The standard deviation of scores capturing the level of consensus or disagreement in the news.
- `news_sent_maxabs_1d`: The maximum absolute sentiment score. This highlights the most significant news event of the day.
- `news_count_1d`: This is the total number of headlines for the day. This serves as a proxy for news volume.

Finally, all of the baseline technical indicator features were standardized to have zero mean and a unit variance using Z-score normalization. This step is crucial to ensure the stability of the neural network training. The mean (μ) and standard deviation (σ) were calculated exclusively on the training data and then applied to both the training and test sets. This is done to prevent any lookahead bias. The formula for feature x is:

$$x_{normalized} = \frac{x - \mu_{train}}{\sigma_{train}} \tag{3.1}$$

The features derived from the LLM are not normalized as their native [-1,1] or positive integer range is already well scaled for the neural network.

3.3 DRL Agent and Architecture

The DRL agent is the core decision-making component. This work adapted the DRL-UTrans agent [28], and introduced specific implementation choices where the original paper was not explicit.

3.3.1 DRL Implementation Choices

In this implementation of the DRL-UTrans agent, two main implementation choices were chosen in order to better process time-series data and train the model:

1. **Global Average Pooling:** The final output of the decoder is passed through a global average pooling layer. This operates across the entire sequence length dimension. This detail was implemented in order to create a stable and holistic representation of the entire look-back window which helps reduce the sensitivity to noise in the most recent time step.

2. **Dependent Action Head:** The output of the weight prediction is concatenated with the pooled feature vector. This occurs before it is fed into the action (Q-value) prediction head. This makes the Q-value estimates conditional on the trade size. Doing this allows the agent to learn the expected value of an action given the capital commitment.

3.3.2 Dual-Head Design for Action and Position Sizing

The model used employs a dual-head output structure. This provides more nuanced control over trading decisions:

- **Action Head:** A fully connected linear layer that outputs the raw logits for all three of the actions (Buy, Sell, Hold). During inference time, an argmax is applied to the logits which selects the action with the highest estimated Q-value.

- **Weight Head:** A fully connected linear layer followed by a `Sigmoid` activation function which output a continuous weight $w \in [0, 1]$. This represents the desired position size as a fraction of the maximum tradeable size.

This design decouples the classification task of choosing an action from the regression task of choosing the associated weight.

3.3.3 Training Algorithm and Hyperparameters

The agent is trained using a DQN algorithm. This approach maintains two separate networks: the `policy_net` that is actively trained and selects the actions, and the `target_net` whose weights are only updated periodically to provide stable Q-value targets during training.

DQN training is facilitated by an Experience Replay buffer, which is used to store past transitions (state, action, reward, next_state). At each training step, a mini-batch of these transitions is sampled to break temporal correlations and improve sample efficiency. The RAdam optimizer [14] is used to update the model parameters. The loss is calculated using the Q-learning loss shown below:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim D}[L_{Huber}(Q_{target}, Q_{pred}(s, a; \theta))] \tag{3.2}$$

where $Q_{target} = r + \gamma \max_{a'} Q(s', a'; \theta_{target})$ and L_{Huber} is the Smooth L1 loss. The weight head is trained effectively through the gradient flowing from the action head without an explicit regression target. Key hyperparameters used in training the agent are listed in Table 3.2.

Table 3.2: DRL Agent and Training Hyperparameters

Hyperparameter	**Value**
Optimizer	RAdam
Learning Rate	1e-3
Loss Function (Q-value)	Huber (Smooth L1)
Discount Factor (γ)	0.99
Batch Size	20
Replay Memory Size	10,000
Target Network Update Frequency	Every 500 steps
Epsilon Start (ϵ_{start})	1.0
Epsilon End (ϵ_{end})	0.1
Epsilon Decay Steps	50 (linear)
Lookback Window Size (L)	12
Transformer Heads	8
Transformer Layers	1

3.4 Trading Environment

In order to train a robust DRL agent, the trading environment must be realistic and well defined. The environment, which was implemented in Python, simulates the mechanics of trading a single stock, including a representation of the state and executing trading actions with appropriate transaction costs.

3.4.1 State Representation

At every time step t, a state S_t is provided to the agent with shape $L \times F$, where $L = 12$ is the lookback window size and F is the total number of features. This vector of features is a concatenation of two components:

1. **Market Features:** The set of technical indicators and LLM sentiment features for the day
2. **Portfolio Features:** A representation of the agent's current portfolio. It is a vector of three real-time metrics: `position_frac` ($\frac{\text{shares held}}{\text{investment capacity}}$), `cash_frac` ($\frac{\text{cash}}{\text{portfolio value}}$), and `remaining_capacity` ($\frac{\text{(capacity - shares held)}}{\text{capacity}}$). These features are tiled across the L time steps. It ensures that the agent is aware of its portfolio state when making decisions.

3.4.2 Action Space and Execution

The agent's action space is a hybrid, containing both discrete and continuous components. At each time step, the outputs from the agent are: a discrete action $a \in \{0 : \text{Buy}, 1 : \text{Sell}, 2 : \text{Hold}\}$ and a continuous weight $w \in [0, 1]$. The action is executed according to the following rules:

- **If Buy:** The number of shares to buy is calculated based on the weight w, the cash available, and the remaining investment capacity.
- **If Sell:** The number of shares to sell is calculated based on the weight w, and the number of shares held.
- **If Hold:** No transaction takes place

All trades are rounded to the nearest lot of 100 to model realistic market constraints. If an invalid trade is attempted (e.g. buying with insufficient cash) then the action is overwritten to "Hold".

3.4.3 Reward Function Design

The reward function utilized in this paper is adapted from the original DRL-UTrans agent [28]. It is designed to reflect the outcome of the agent's actions. The reward function is outlined in equation 2.20. The structure of the reward function gives immediate feedback on the realised gains and losses from both the executed trade, and if the current positions are held.

3.4.4 Transaction Cost Modelling

To simulate a realistic trading environment, a transaction cost is implemented. A commission fee of 0.1% (10 basis points) is deducted from the total value of every transaction. By including this cost, it penalises frequent trading and further ensures that profitable strategies overcome this frictional cost, similarly to a real trading environment.

3.5 Experimental Design

This thesis systematically evaluates the research question by employing a rigorous experimental protocol based on an ablation study. This enables the isolation and quantification of the incremental benefit of each sentiment channel.

3.5.1 Ablation Study Framework

An ablation study is a process of investigation where components from a system are systematically removed/added in order to understand its individual contribution to the overall performance. As outlined in Figure 3.2, this thesis utilises an additive ablation study. It initially starts with the baseline agent, consisting of only baseline features, and then incrementally adds the LLM-derived feature sets, both individually and in combination.

3.5.2 Four Experimental Configurations

The four configurations are detailed below in Figure 3.3 with the features used.

3.5.3 Training Protocol

A strict training protocol, outlined below, was followed in order to ensure the robustness and reliability of the findings.

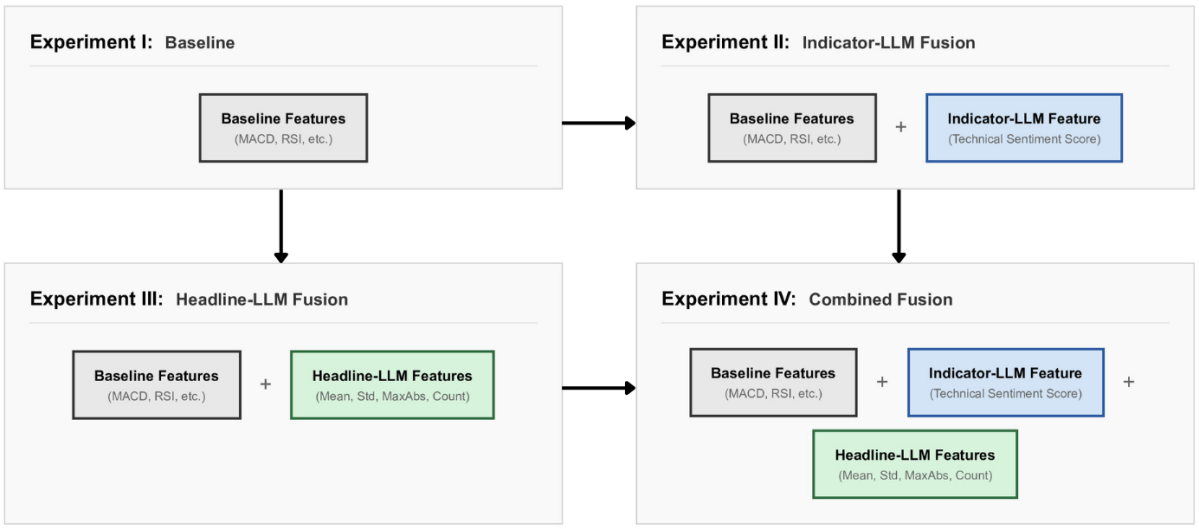


Figure 3.2: The four-part ablation study design. Each experiment incrementally adds a new feature set to the baseline agent, allowing for a systematic analysis of the marginal contribution of each information channel.

Table 3.3: Experimental Configurations and Associated Feature Sets

Experiment	**Name**	**Features Used**
I	Baseline DRL-UTrans	Baseline Technical Indicators
II	Indicator-LLM Fusion	Baseline + Technical Indicator Sentiment Score
III	Headline-LLM Fusion	Baseline + Aggregated Daily Headline Features
IV	Combined Fusion	Baseline + All LLM-Derived Features

- **Data Split:** A fixed 70/30 chronological train/test split was used for all stocks
- **Seeding:** Each of the four experiments was conducted for each of the 10 stocks using three random seeds $\{8, 26, 1111\}$. This helps mitigate the effects of random weight initialization and stochasticity in the training process
- **Model Selection:** The single best run out of the three was chosen for the final analysis. The selection criteria used was the final return percentage achieved on the out-of-sample test set.
- **Training Duration:** All agents were trained for 50 epochs

3.5.4 Evaluation Metrics

A comprehensive suite of metrics was used to assess both the returns and risk-adjusted performance.

- **Final Return Percentage:** The total percentage gain or loss over the test period.
- **Compound Annual Growth Rate (CAGR):** Annualised average rate of return for the test period.
- **Sharpe Ratio:** Measures the risk adjusted returns. It calculates the excess return over the risk-free rate per unit of volatility (standard deviation of returns). A risk-free rate of 2% was assumed.
- **Sortino Ratio:** A variation of the Sharpe ratio that only uses the standard deviation of negative returns in the denominator.
- **Maximum Drawdown (MDD):** The largest peak-to-trough decline in the portfolio value. It is seen as an indicator of the maximum downside risk.

3.6 Implementation Details

This section will detail the software and hardware environments used to conduct the research, facilitating reproducibility.

3.6.1 Software and Hardware

The implementation was developed entirely in `Python (3.13.5)`. The deep learning components were built using `PyTorch (2.8.0)` while the data manipulation and numerical operations were handled by `Pandas (2.3.1)` and `NumPy (2.3.2)`. Data visualisation was performed using `Matplotlib (3.10.5)`. The LLM was accessed via the `Groq (0.31.0)` client, and the financial data was downloaded using `yfinance (0.2.65)`.

The model training was conducted on a workstation equipped with an Nvidia A10G GPU with 24 GB of VRAM.

3.6.2 Data Caching

All of the downloaded price data and LLM-generated sentiment was cached locally. This ensures that the same feature set is used between experiments. This further eliminated variability from external data sources or API changes.

3.7 Summary

This chapter provided the architectural and methodological blueprint for the entire dissertation. All of the design choices, including the data sources, feature-engineering pipeline, DRL agents architecture, and the simulated environment, were outlined and made to ensure the rigour, reproducibility, and practical relevance of the study. This framework provided a consistent foundation which will enable a systematic comparison of the experimental configurations as outlined in the following chapter.

Chapter 4

Experiments and Results

This chapter presents the four-part ablation study and details the setup, results and analysis for each configuration. It begins by establishing the baseline agent, then sequentially introduces two distinct sentiment channels before a final model combines them. The chapter then synthesizes these findings through a comparative analysis and an evaluation of model performance across different market regimes.

4.1 Introduction

This chapter details the four-part ablation study to systematically quantify the incremental impact of the sentiment features derived from the LLM. This ablation framework allows for the precise measurement of each component's contribution to trading performance. The subsequent section presents the setup and a descriptive analysis of the results for each experiment. Finally, a comprehensive cross-model comparison is conducted to extract key highlights from the research.

Reference figures and tables are included in this chapter, while the comprehensive results are included in the Appendices (7). This includes the detailed performance metrics in Table 7.1 while the corresponding equity curve plots are in Appendix 7.2

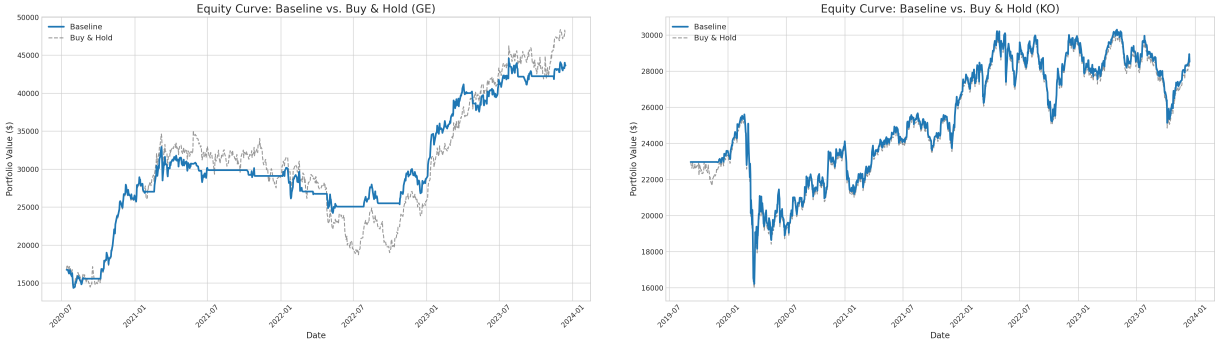
4.2 Exp I - Baseline DRL-UTrans agent:

This initial experiment establishes a baseline performance benchmark by training the DRL-UTrans agent exclusively on the technical indicators. This agent serves as the control,

against which all the other sentiment-augmented agents are measured.

- **Experimental Setup:** The agent's state representation consists solely of the four baseline technical indicators (`MACD`, `KDJ_K`, `Open-Close Difference` and `RSI_14`) along with the features outlining the current state of the portfolio. No LLM-derived sentiment features are used.

- **Results and Analysis:** On average, the baseline agent proved to be a robust foundation. As detailed in the aggregate results (Table 4.1), the agent achieved a CAGR of 15.26% and a Sharpe ratio of 0.51. The level of risk-adjusted performance is identical to that of the Buy & Hold strategy, which also achieved a Sharpe ratio of 0.51. The critical initial finding is that the framework provides inherent risk management, with the baseline agent achieving an Average MDD of -36.04%, which was significantly lower than the -43.81% MDD experienced by the Buy & Hold strategy. A representative equity curve for the baseline agent is presented in Figure 4.1, and this superior risk management is evident in Figure 4.1a where the agent held no stocks during the most turbulent periods.



(a) Performance on GE

(b) Performance on KO

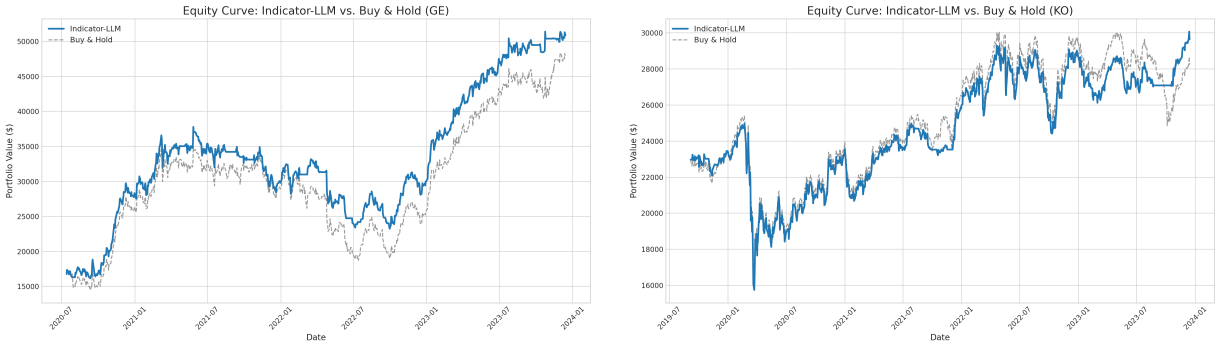
Figure 4.1: Equity curves of the Baseline DRL-UTrans agent on GE and KO. This agent, operating on quantitative features only, serves as the performance benchmark.

4.3 Exp II - Indicator-LLM Fusion

The second experiment assesses the marginal impact of augmenting the baseline agent with a daily sentiment score, which is derived from an LLM when fed a textual summary of

technical indicators. The objective of this was to determine if an abstracted and qualitative interpretation of quantitative features could enhance performance.

- **Experimental Setup:** The baseline agent is augmented with a single daily sentiment score from $[-1, 1]$ which is derived from the LLM's analysis of the four technical indicators. This score is added to the baseline agent's state representation.
- **Results and Analysis:** The results indicate that the technical indicator sentiment score provides no discernable benefit and actually degrades performance. The indicator-LLM agent had an average CAGR of 15.16% and an average Sharpe ratio of 0.45. Both of these are slightly lower than the baseline (Table 4.1). This configuration produced the lowest risk-adjusted returns of any of the models tested. This finding suggests that an LLM-based summary of quantitative technical indicators may introduce noise or redundancy rather than a valuable signal.



(a) Performance on GE

(b) Performance on KO

Figure 4.2: Equity curves of the Indicator-LLM Fusion agent on GE and KO. This agent augments the baseline with sentiment derived from technical indicator summaries.

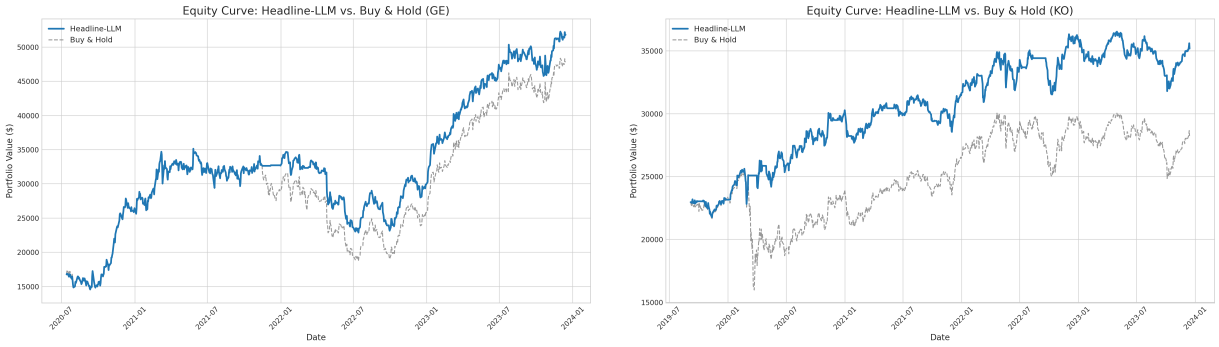
4.4 Exp III - Headline-LLM Fusion

The third experiment measures the contribution of the sentiment features derived from daily financial news headlines. This configuration enhances the baseline agent's state representation with aggregated statistics that capture the narrative climate of the market.

- **Experimental Setup:** In this experiment, the baseline agent is augmented with four aggregated statistical features derived from the LLM's analysis of daily news headlines. These features are: `news_sent_mean_1d`, `news_sent_std_1d`, `news_sent_maxabs_1d`,

and `news_count_1d`. The technical indicator sentiment feature used in Exp II is not included in the agent's state.

- **Results and Analysis:** This agent emerged to be the top-performing strategy, achieving the highest average Sharpe Ratio of 0.59 and Sortino Ratio of 0.87. This is a notable 15% improvement in the Sharpe Ratio compared to the baseline agent. On certain stocks, the impact of this feature was particularly transformative. For instance, the agent more than doubled the Sharpe Ratio for KO (from 0.34 to 0.77) while dramatically reducing its maximum drawdown from -36.7% to -13.0%. This suggests that news-derived sentiment can capture unique, non-price information that is highly valuable to a DRL agent's decision making process.



(a) Performance on GE

(b) Performance on KO

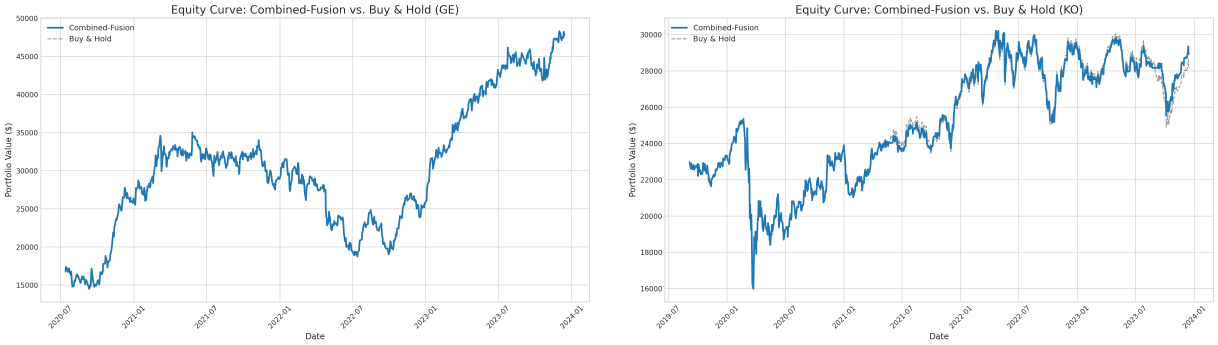
Figure 4.3: Equity curves of the Headline-LLM Fusion agent on GE and KO, demonstrating the impact of augmenting the baseline agent with news-derived sentiment features.

4.5 Exp IV - Combined-Fusion Agent

The final experiment evaluates an agent trained with all of the available features. It combines the baseline technical indicators with both the indicator-derived and the headline-derived LLM sentiment features. This experiment aims to determine if two distinct sentiment channels offer synergistic benefits over a single channel.

- **Experimental Setup:** The agent is trained on the combined feature sets from Exp I, II and III. This represents the most feature-rich state representation.

- **Results and Analysis:** The Combined-Fusion agent outperformed the baseline, achieving a CAGR of 17.95% and a Sharpe Ratio of 0.55. Although the agent achieved the highest CAGR, its risk-adjusted performance was lower than that of the Headline-LLM Fusion. This outcome suggests that there may be a degree of information redundancy between the feature sets and further supports the negative or noise impact of the indicator-LLM signal. This indicates that the agent's performance is an aggregate of its components rather than a synergistic improvement.



(a) Performance on GE

(b) Performance on KO

Figure 4.4: Equity curves of the Combined-Fusion agent on GE and KO. This agent incorporates all available features: quantitative, indicator sentiment, and headline sentiment.

4.6 Cross-Experiment Comparative Analysis

This section synthesises the results from all of the four experiments and the Buy & Hold strategy to provide a holistic overview of the findings.

4.6.1 Aggregate Performance Metrics

The average performance across all 10 stocks is detailed in Table 4.1. The data suggest that augmenting the DRL agent with LLM-derived sentiment may be a viable source of alpha. Both the Headline-LLM and the Combined-Fusion agents outperform the Baseline and Buy & Hold strategies on key risk-adjusted metrics. Despite this, the Baseline agent had the lowest MDD of any model. The Headline-LLM model was the top performer, achieving the highest Sharpe Ratio (0.59) and Sortino Ratio (0.87). The Combined-Fusion agent achieved the greatest CAGR across all the models. Critically, all of the DRL agents

managed risk more effectively than the passive benchmark as seen by their lower average MDDs.

Table 4.1: Aggregate Performance Metrics Across All Stocks

Model / Strategy	**CAGR (%)**	**Sharpe Ratio**	**Sortino Ratio**	**Max Drawdown (%)**
Baseline	15.26	0.51	0.75	**-36.04**
Indicator-LLM	15.16	0.45	0.68	-38.33
Combined-Fusion	**17.95**	0.55	0.80	-42.49
Headline-LLM	17.78	**0.59**	**0.87**	-38.28
Buy & Hold	16.46	0.51	0.75	-43.81

4.6.2 Visual Comparison

In order to provide a more intuitive understanding of the comparative performance between agents, the key aggregated metrics are visualised as bar charts. Figure 4.5 shows the average CAGR, where the Combined-Fusion and Headline-LLM agents exhibited the highest growth. Figure 4.6 displays the average Sharpe ratios. This visually confirms the superior performance achieved by the Headline-LLM agent. Finally, Figure 4.7 compares the average MDD, which highlights the risk-management advantage of all DRL agents over the Buy & Hold strategy and shows the Baseline achieved the greatest performance.

4.6.3 Win-Rate Analysis

As evidenced in the win-rate table below (Table 4.2), the top-performing agents consistently outperformed the others. The Headline-LLM agent not only had the best average performance, but was also the most frequent top performer on a stock-by-stock basis. This consistency emphasized its robustness across different market dynamics.

Table 4.2: Win-Rate Analysis of Agents Across 10 Stocks

Agent	**Sharpe Ratio Wins**	**CAGR Wins**
Headline-LLM	**4**	**6**
Combined-Fusion	3	3
Baseline	2	1
Indicator-LLM	1	0

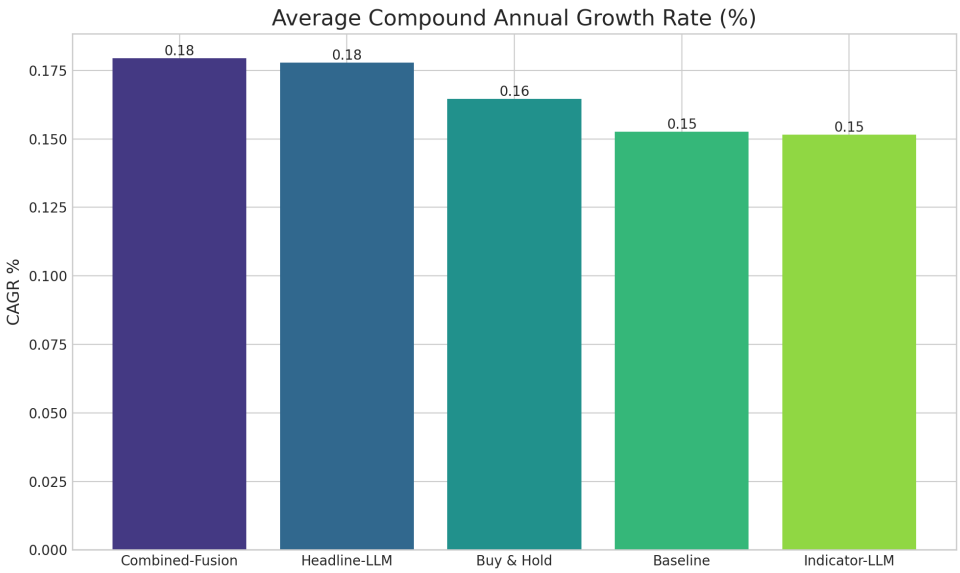


Figure 4.5: Comparison of the average Compound Annual Growth Rate (CAGR) across all agents and the Buy & Hold benchmark. The values represent the mean performance over the ten-stock universe.

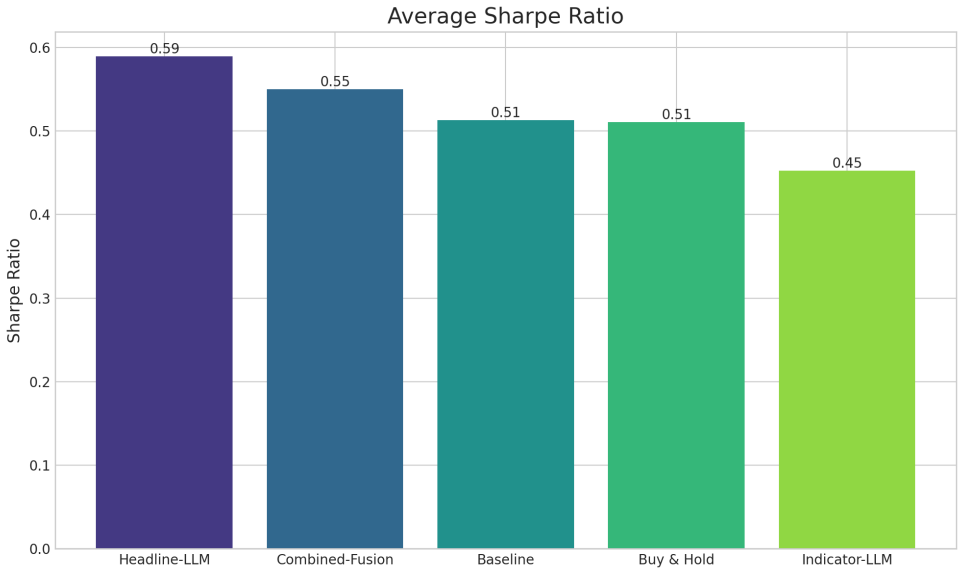


Figure 4.6: Comparison of the average Sharpe Ratio across all agents and the Buy & Hold benchmark. This metric evaluates risk-adjusted returns, with higher values indicating superior performance.

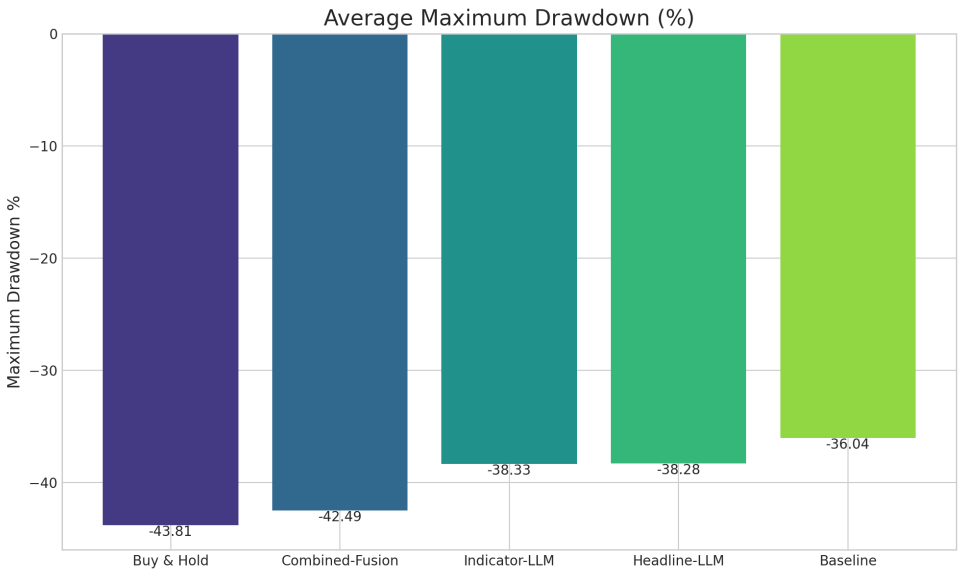


Figure 4.7: Comparison of the average Maximum Drawdown (MDD) across all agents and the Buy & Hold benchmark. Lower (more negative) values indicate greater peak-to-trough portfolio decline.

4.7 Performance Across Market Regimes

In order to evaluate the robustness of the agents under different market conditions (Objective III), their performance was analysed across three distinct, pre-identified periods: a bull market, bear market, and a volatile recovery phase. The results of this analysis are presented in Figure 4.8.

These results show that the Headline-LLM and Combined-Fusion agents outperformed all the others during the Bull market and the Volatile/Recovery phase. Despite this, they had the worst performance during the Bear market. This indicates that the headline sentiment may be less reliable during periods of decline in the stock market.

4.8 Summary

The empirical results presented in this chapter provide several key, data-driven insights into the performance of each agent. The DRL-UTrans agent was a robust baseline that matched passive benchmark returns, albeit with superior risk management. Augmenting this agent with sentiment features derived from a LLM was shown to be a viable method for enhancing performance, but this was highly dependent on the source of the sentiment.

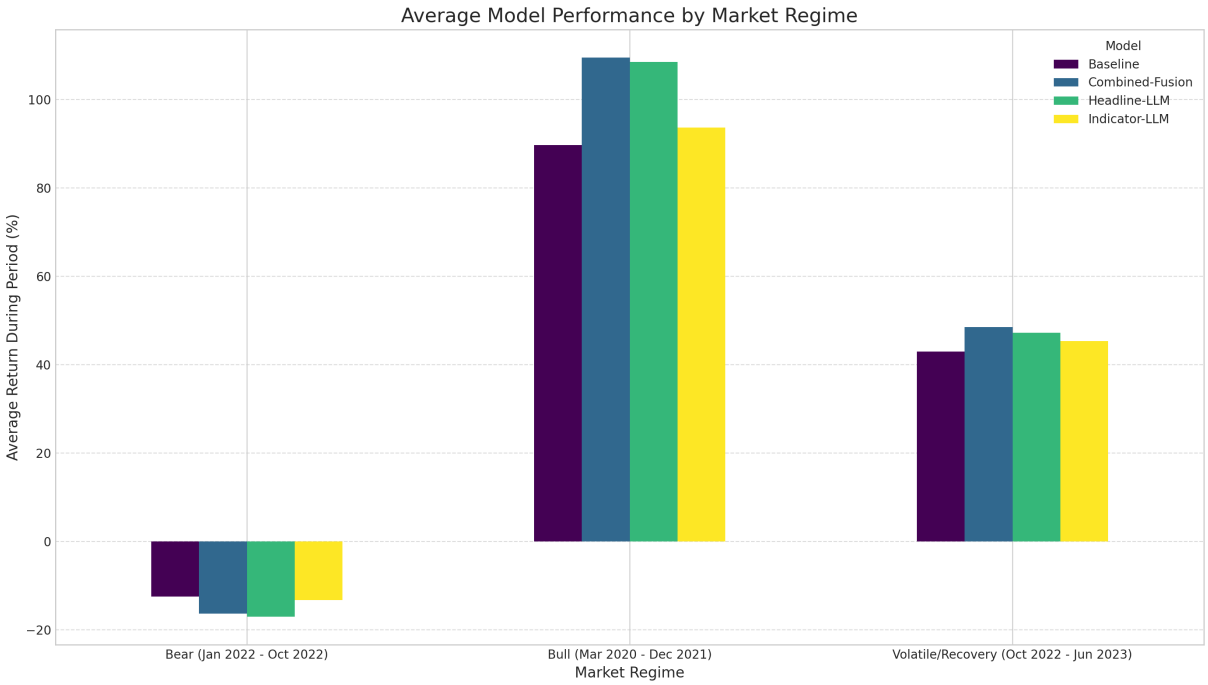


Figure 4.8: Average agent returns during distinct market regimes. The performance of each agent is evaluated during pre-defined bull, bear, and volatile/recovery periods to assess its adaptability.

The analysis demonstrates that the headline derived sentiment provides significant and consistent improvements in risk adjusted returns across a variety of stocks and market conditions. On the other hand, sentiment derived from technical indicators appears to be redundant and offers no discernible benefit over the baseline. Finally, the combined agent showed evidence of information redundancy instead of synergy. This suggests that feature selection is critical. Augmenting an agent with headline sentiment proved to be the superior strategy, which validated the central hypothesis of this dissertation.

Chapter 5

Discussion

This chapter interprets the empirical findings from the preceding chapter to answer the central research questions of the dissertation. It begins by examining the results of the ablation study and proposes theoretical explanations for these outcomes. The chapter then discusses these findings within the broader academic context, and considers their implications for the Efficient Market Hypothesis and the future of multimodal alpha generation. It then discusses the limitations of the study, before finally proposing avenues for future research that directly address the limitations of this work and build upon the results of the thesis.

5.1 Introduction

The preceding chapter presented the empirical results of the ablation study and objectively quantifies the performance of the DRL agents. This chapter moves from description to interpretation in order to answer the central research question: *To what extent does augmenting a state-of-the-art Deep Reinforcement Learning trading agent with structured sentiment features, derived from Large Language Models, improve trading performance and risk-adjusted returns?.*

The core finding of Chapter 4 was that not all sentiment is equally valuable. While sentiment derived from news headlines provided a substantial and consistent improvement to risk-adjusted returns, the sentiment derived from technical indicators decreased performance. This chapter dissects these outcomes, first by interpreting the findings and then by situating them within the broader context of financial theory and practice. Finally, the

chapter will evaluate the study's limitations before proposing avenues for future research.

5.2 Interpretation of Key Findings

The ablation study was designed to isolate the marginal contribution of each of the sentiment channels. The results of this experiment provide a clear narrative about the nature of information in financial markets and how agents can effectively utilise it.

5.2.1 The Primacy of Exogenous Information: Why News Succeeded

The most significant finding of this dissertation is the superior performance of the Headline-LLM Fusion agent. As seen in Chapter 4, the agent had an average of 15% improvement in Sharpe Ratio over the baseline, which indicates that news headlines contain novel, orthogonal information that is not fully latent in historical price data or technical indicators. This "alpha" likely stems from the ability of news to capture forward-looking or fundamental events, such as earnings surprises, product announcements, or macroeconomic shifts, which can act as a leading indicator for price movements. This is contrasted by technical indicators, which are derived from historical prices and thus lagging or coincident. News provides an exogenous signal that can break from historical patterns and inform the agent of impending regime changes. The improvement present in both returns and risk for a stock like KO, whose valuation is heavily influenced by brand perception and consumer trends often reflected in the news, underscores this point.

5.2.2 The Peril of Redundancy: Why Indicator Sentiment Failed

In stark contrast, the Indicator-LLM Fusion agent was the worst performing strategy. This result supports the hypothesis that using an LLM to summarise existing quantitative indicators is a lossy and noisy compression of information. The raw numerical technical indicators are precise inputs, while the LLM summary is an imprecise abstraction that can lose critical nuances and introduce ambiguity. The DRL-UTrans agent is designed to learn complex, non-linear relationships from the raw technical indicators and gains nothing from this redundant and less precise feature. From this, it can be seen that abstracting already-quantified information can be actively harmful.

5.2.3 Information Dilution: Why the Combined Agent did not Excel

The fact that the Combined-Fusion agent failed to outperform the Headline-LLM agent in both the Sharpe and Sortino ratio points toward information redundancy and feature overload. It's possible that the high-value, orthogonal signal from headlines was diluted by the noisy, redundant signal from the indicator sentiment. The agent may have struggled to assign the correct weighting to the valuable information with the larger and more complex state space, which led to suboptimal, yet still profitable, performance. From this, it can be inferred that a critical property of multimodal systems is that signal quality is far more important than signal quantity. Adding more features, especially those that are correlated or of low quality, does not guarantee improved performance and can instead hinder the learning process.

5.3 Implications of this Study

The findings of this research have significant implications for both academic theory and for applications in quantitative finance. This directly addresses the contributions from Chapter 1

5.3.1 For Academia: Challenging Market Efficiency and Defining Multimodal Alpha

The results of this study present evidence that challenges the semi-strong form of the Efficient Market Hypothesis (EMH) as introduced in Chapter 2. The clear performance improvement resulting from the inclusion of news headline sentiment supports the hypothesis that it is possible to extract alpha from publicly available information. This finding implies that such information is not instantaneously and perfectly reflected in asset prices. This leaves a window of opportunity for these agents to exploit this transient inefficiency.

This work further contributes to the academic setting by extending the findings of prior work in using NLP in finance. Previous studies have explored the predictive power of sentiment in directional forecasting, and this study builds on this by integrating that signal into a sequential decision-making framework. This moves beyond prediction to autonomous action, which offers a more complete model of an intelligent trading agent. The results strongly suggest that the frontier of 'alpha' generation is shifting to incorporate

quantitative analysis and natural language understanding. There is likely alpha to be found by extending this research and developing superior methods to fuse disparate, multimodal information sources.

5.3.2 For Practitioners: A Blueprint for Integrating LLMs

This dissertation provides evidence for quantitative trading firms to invest in the infrastructure to process high-quality, exogenous data streams like news feeds. The observed 15% improvement in the Sharpe Ratio represents a significant competitive edge that is difficult to achieve by refining and improving existing quantitative methods. This validates the premise of Contribution I by providing a pragmatic pipeline for this integration.

The results of this experiment also serve as a cautionary tale against indiscriminate inclusion of features, as evident in the Combined-Fusion agent's failure to achieve synergistic benefits. This underscores the importance of rigorous feature selection and the pursuit of truly orthogonal features. The takeaway is to focus on high-quality, uncorrelated feature sources rather than simply increasing the volume of inputs.

5.4 Limitations of the Study

While the findings of the dissertation are compelling, it is essential to acknowledge its limitations to contextualize the results and ensure their appropriate interpretation.

5.4.1 Stochasticity and Training Stability

One of the most significant limitations of this study is the inherent randomness of the DRL agent and the training process. This is evidenced by the run-to-run variance data as the performance of any given agent is not a single point but a distribution of possible outcomes. A full table outlining this variation is available in the Appendix (Section 7.3). Some agents exhibited a large range in CAGR and Sharpe Ratio across the runs. For instance, on the baseline agent on MRK, there was a Sharpe Ratio range of 1.11 across the three seeds. This reveals the inherent instability of the DRL agent. The methodology of selecting the single best run for the main analysis demonstrates the potential of a given architecture; however, it must be acknowledged that the average performance across all runs would be lower and the variance higher.

Interestingly, a key hypothesis emerges from the variance data: both of the agents augmented with the headline sentiment exhibited lower run-to-run variance than purely quantitative models. This is shown explicitly in Table 5.1. As shown, the Headline-LLM agent had the least variation, and this difference is quite drastic, with it achieving almost 3 times less variation across both the standard deviation and range for both the CAGR and Sharpe Ratio. This suggests that the external, narrative-driven signal may act as a regulariser that grounds the agent's policy and prevents convergence to wildly different local optima. This directly addresses Contribution III. Nonetheless, the stochasticity remains a major hurdle for real-world deployment where consistency is paramount.

Table 5.1: Analysis of Agent Stability (Averaged Across All Tickers)

Experiment	**Avg. CAGR Std. Dev.**	**Avg. CAGR Range (Δ)**	**Avg. Sharpe Std. Dev.**	**Avg. Sharpe Range (Δ)**
Baseline	7.522	16.800	0.257	0.599
Indicator-LLM	7.262	17.020	0.248	0.559
Headline-LLM	**2.410**	**5.629**	**0.083**	**0.194**
Combined	5.140	12.297	0.111	0.261

Note: Lower values indicate greater stability and less sensitivity to random initialization during training.

5.4.2 Methodological and External Validity Constraints

Several of the methodological choices made may limit the generalisability of the findings. The study was conducted on only ten US equities, and the performance of this model could differ across different asset classes (e.g., commodities, forex), international stocks, or broader market indexes. The DRL-UTrans agent is just one of the many possible architectures, and the results may be specific to the combination of the U-Net and Transformer layers.

Furthermore, one of the critical methodological constraints is the potential of lookahead bias from the pre-trained LLM. As noted in Chapter 2, the LLM was trained on data up to 2025, and when analysing a headline from anytime in the past, it may process that information with implicit knowledge of future events, which could inflate its performance. Thus, these results should be interpreted as an upper bound of the current performance achievable.

A further limitation pertains to the nature and source of the textual data. The study utilized the publicly available FNSPID dataset [5]. While quite extensive, the dataset in-

troduces uncontrolled variables, including potential selection bias from their specific choice of news outlets, a lack of guaranteed temporal fidelity regarding the precise time a story is said to have been published, and other potential quality issues. A production-grade system would need to instead leverage a proprietary and low-latency news feed to ensure the timeliness and breadth of the information are sufficient.

Moreover, only the headline of the news story was used to produce the sentiment score, which would reduce the informational depth. Headlines are, by design, a condensed and often sensationalized summary of the article. This can lead to them often lacking the critical nuance, specific quantitative data (e.g. revenue figures, earnings-per-share), and the broader context that can be found within the article's full text. The decision of the agent was based on a compressed and potentially incomplete representation of the available news, which could have limited its performance.

5.5 Avenues for Future Research

Both the findings and the limitations of this study give rise to several promising directions for future research.

Firstly, future work should address the limitation of training instability. Research could explore ways to mitigate the run-to-run variance. One potential approach is to use ensemble techniques where the policies from multiple agent/training runs are averaged to create a more powerful and robust meta-agent. Other DRL architectures, such as PPO or SAC, could also be a valuable next step as they are known for their stability.

Secondly, the feature engineering pipeline could be expanded to include more advanced or alternative LLMs. There could also be work done to combine the scores from multiple LLMs in order to have a more robust predictor of the true sentiment. Another crucial extension would be to incorporate the numerical data from news articles alongside the sentiment score. Additionally, a proprietary and robust news dataset from a diverse range of sources with precise sub-second timestamps could create a higher-fidelity and robust signal. The use of the entire news article to produce the sentiment score should also be explored.

Finally, this study's single-agent and single-asset framework could be expanded into both a portfolio-level or multi-agent setting. A future agent could be trained and tasked with allocating capital across a universe of assets, which would be a more realistic and complex task. There is potential that incorporating sentiment derived from news headlines

and articles could potentially improve the robustness of these models as they could better allocate resources according to broader market dynamics.

Chapter 6

Conclusion and Future Work

This chapter concludes the thesis by synthesizing the research objectives, key findings, and the broader implications of the study. It begins by presenting a summary of the investigation's purpose and ablation study used to address it. The chapter then reiterates the principal findings from the experimental results, highlighting their significance. Following this, it formally states the main contributions of the work. Finally, the chapter offers concluding reflections on the study's significance, positioning this dissertation as a foundational step that establishes a clear and compelling agenda for future work at the intersection of artificial intelligence and quantitative finance.

6.1 Summary of the Investigation

This dissertation set out to quantify the impact of structured sentiment features derived from LLMs on the performance of a DRL-based trading agent. In order to address this question, a four-part ablation study was designed and implemented to measure the marginal benefit of the sentiment channels. This work built upon the DRL-UTrans framework introduced by Yang et al. [28]. The primary objectives of this thesis were to design a feature-engineering pipeline for multimodal fusion, empirically quantify the value of each sentiment signal, and validate the performance of the agents. The study explored the effects of integrating sentiment from technical indicator summaries and news headlines into the agent's state space. A comprehensive comparative analysis of each configuration was undertaken to investigate the marginal benefit of each channel.

6.2 Principal Findings

The core findings of this investigation are multifaceted and provide compelling preliminary evidence of the efficacy of multimodal DRL agents in quantitative finance.

Firstly, the baseline DRL agent proved to be a robust and effective foundation that was able to match the risk-adjusted returns of a Buy & Hold strategy while exhibiting superior risk management using only quantitative features. This success fulfills Objective III as it demonstrated an inherent ability to navigate market risk.

Secondly, the ablation study provided initial evidence of a clear difference in value between the two sentiment sources. The Headline-LLM agent emerged as the top performer as it achieved a 15% improvement in Sharpe Ratio over the baseline. This suggests that the sentiment score derived from news sources contains novel, exogenous information that is not fully captured by traditional price-based indicators. In contrast, the indicator-LLM proved to be the worst-performing strategy, which demonstrates that using an LLM to summarise existing quantitative indicators introduces information redundancy that can hinder performance.

Finally, the study found that improved performance cannot be guaranteed by simply adding more features. The Combined-Fusion agent underperformed the Headline-LLM agent on a risk-adjusted basis. This finding points to a degree of information dilution where the signal quality is more critical than the signal quantity.

6.3 Contributions of the Study

This dissertation makes several key contributions at the intersection of machine learning and quantitative finance.

- **A Pragmatic Pipeline for Multimodal Fusion:** This thesis designed and implemented a successful and replicable feature engineering pipeline. This provides a methodological blueprint for integrating LLM-derived sentiment into the state representation of DRL agents without significant architectural overhauls.
- **A Systematic Quantification of Sentiment Value:** The rigorous ablation study provided preliminary evidence on the additive value of sentiment sources. These findings provide data-driven conclusions on how, and when, language-based features can enhance a trading strategy.

- **An Analysis of DRL Agent Stability:** This research went beyond analysing pure performance metrics and investigated the inherent stochasticity in the training process of the DRL agent. This thesis highlights the issue of run-to-run variance and provides initial evidence that high-quality headline sentiment can act as a regularising signal that can improve training consistency. This is a key hurdle for real-world deployment.

6.4 Final Reflections and Future Outlook

This dissertation presents promising, yet foundational, findings in the realm of autonomous agents that can effectively fuse language and quantitative data to trade in financial markets. The work has shown that a DRL agent has the potential to extract valuable alpha from news headlines and can utilise that information to improve risk-adjusted returns and manage downside risk.

Despite these positive initial findings, these results should not be viewed as a definitive conclusion. The high run-to-run variance and the information redundancy in the combined agent underscore the need for continuous and rigorous research in the field. Furthermore, the methodological limitations, such as the use of a single DRL architecture, a finite set of stocks, and a public news dataset, point to vast areas for further research.

Ultimately, this dissertation serves as a starting point and a methodological blueprint for future explorations in the field. It lays the groundwork for a new multimodal trading system and offers a strong call to action for researchers to address the remaining challenges outlined. The journey to creating truly intelligent, multimodal, and robust financial agents has only just begun.

Bibliography

[1] Dogu Araci. *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*. arXiv:1908.10063 [cs]. Aug. 2019. DOI: 10.48550/arXiv.1908.10063. URL: http://arxiv.org/abs/1908.10063 (visited on 08/13/2025).

[2] Souradeep Chakraborty. *Capturing Financial markets to apply Deep Reinforcement Learning*. arXiv:1907.04373 [q-fin]. Dec. 2019. DOI: 10.48550/arXiv.1907.04373. URL: http://arxiv.org/abs/1907.04373 (visited on 08/01/2025).

[3] Dave Cliff, Dan Brown, and Philip Treleaven. *Technology Trends in the Financial Markets: A 2020 Vision*. UK Government Office for Science, Sept. 2011.

[4] Rick Di Mascio, Anton Lines, and Narayan Y. Naik. *Alpha Decay and Institutional Trading*. en. SSRN Scholarly Paper. Rochester, NY, Nov. 2017. DOI: 10.2139/ssrn.2580551. URL: https://papers.ssrn.com/abstract=2580551 (visited on 08/16/2025).

[5] Zihan Dong, Xinyu Fan, and Zhiyuan Peng. *FNSPID: A Comprehensive Financial News Dataset in Time Series*. arXiv:2402.06698 [q-fin]. Feb. 2024. DOI: 10.48550/arXiv.2402.06698. URL: http://arxiv.org/abs/2402.06698 (visited on 08/18/2025).

[6] Eugene F. Fama. "Efficient Capital Markets: A Review of Theory and Empirical Work". In: *The Journal of Finance* 25.2 (1970). Publisher: [American Finance Association, Wiley], pp. 383–417. ISSN: 0022-1082. DOI: 10.2307/2325486. URL: https://www.jstor.org/stable/2325486 (visited on 08/17/2025).

[7] Georgios Fatouros et al. "Transforming sentiment analysis in the financial domain with ChatGPT". In: *Machine Learning with Applications* 14 (Dec. 2023), p. 100508. ISSN: 2666-8270. DOI: 10.1016/j.mlwa.2023.100508. URL: https://www.sciencedirect.com/science/article/pii/S2666827023000610 (visited on 08/13/2025).

[8] Tuomas Haarnoja et al. *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*. arXiv:1801.01290 [cs]. Aug. 2018. DOI: 10.48550/arXiv.1801.01290. URL: http://arxiv.org/abs/1801.01290 (visited on 08/02/2025).

[9] Hado van Hasselt, Arthur Guez, and David Silver. *Deep Reinforcement Learning with Double Q-learning*. arXiv:1509.06461 [cs]. Dec. 2015. DOI: 10.48550/arXiv.1509.06461. URL: http://arxiv.org/abs/1509.06461 (visited on 08/13/2025).

[10] Jacob Boudoukh et al. *Which News Moves Stock Prices? A Textual Analysis*. URL: https://www.nber.org/system/files/working_papers/w18725/w18725.pdf (visited on 08/15/2025).

[11] Tingsong Jiang and Qingyun Zeng. *Financial sentiment analysis using FinBERT with application in predicting stock movement*. arXiv:2306.02136 [q-fin]. June 2025. DOI: 10.48550/arXiv.2306.02136. URL: http://arxiv.org/abs/2306.02136 (visited on 08/13/2025).

[12] Albert S. Kyle. "Continuous Auctions and Insider Trading". In: *Econometrica* 53.6 (1985). Publisher: [Wiley, Econometric Society], pp. 1315–1335. ISSN: 0012-9682. DOI: 10.2307/1913210. URL: https://www.jstor.org/stable/1913210 (visited on 08/16/2025).

[13] Patrick Lewis et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. arXiv:2005.11401 [cs]. Apr. 2021. DOI: 10.48550/arXiv.2005.11401. URL: http://arxiv.org/abs/2005.11401 (visited on 08/13/2025).

[14] Liyuan Liu et al. *On the Variance of the Adaptive Learning Rate and Beyond*. arXiv:1908.03265 [cs]. Oct. 2021. DOI: 10.48550/arXiv.1908.03265. URL: http://arxiv.org/abs/1908.03265 (visited on 08/20/2025).

[15] Xiao-Yang Liu et al. *FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance*. arXiv:2011.09607 [q-fin]. Mar. 2022. DOI: 10.48550/arXiv.2011.09607. URL: http://arxiv.org/abs/2011.09607 (visited on 08/12/2025).

[16] Volodymyr Mnih et al. *Asynchronous Methods for Deep Reinforcement Learning*. arXiv:1602.01783 [cs]. June 2016. DOI: 10.48550/arXiv.1602.01783. URL: http://arxiv.org/abs/1602.01783 (visited on 08/13/2025).

[17] Volodymyr Mnih et al. *Playing Atari with Deep Reinforcement Learning*. arXiv:1312.5602 [cs]. Dec. 2013. DOI: 10.48550/arXiv.1312.5602. URL: http://arxiv.org/abs/1312.5602 (visited on 08/02/2025).

[18] Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. "Reinforcement learning for optimized trade execution". In: *Proceedings of the 23rd international conference on Machine learning*. ICML '06. New York, NY, USA: Association for Computing Machinery, June 2006, pp. 673–680. ISBN: 978-1-59593-383-6. DOI: 10.1145/1143844.1143929. URL: https://doi.org/10.1145/1143844.1143929 (visited on 08/12/2025).

[19] Yunhua Pei et al. *Cross-Modal Temporal Fusion for Financial Market Forecasting*. arXiv:2504.13522 [cs]. Aug. 2025. DOI: 10.48550/arXiv.2504.13522. URL: http://arxiv.org/abs/2504.13522 (visited on 08/13/2025).

[20] Ran Aroussi. *yfinance: Download market data from Yahoo! Finance API*. URL: https://github.com/ranaroussi/yfinance (visited on 08/18/2025).

[21] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. 2nd ed. Bradford Books, Nov. 2018. ISBN: 0-262-03924-9. URL: https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf (visited on 08/13/2025).

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv:1505.04597 [cs]. May 2015. DOI: 10.48550/arXiv.1505.04597. URL: http://arxiv.org/abs/1505.04597 (visited on 08/04/2025).

[23] John Schulman et al. *Proximal Policy Optimization Algorithms*. arXiv:1707.06347 [cs]. Aug. 2017. DOI: 10.48550/arXiv.1707.06347. URL: http://arxiv.org/abs/1707.06347 (visited on 08/02/2025).

[24] Yanxin Shen and Pulin Kirin Zhang. "Financial Sentiment Analysis on News and Reports Using Large Language Models and FinBERT". In: *2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS)*. ISSN: 2834-8567. July 2024, pp. 717–721. DOI: 10.1109/ICPICS62053.2024.10796670. URL: https://ieeexplore.ieee.org/abstract/document/10796670 (visited on 08/13/2025).

[25] Ashish Vaswani et al. *Attention Is All You Need*. arXiv:1706.03762 [cs]. Aug. 2023. DOI: 10.48550/arXiv.1706.03762. URL: http://arxiv.org/abs/1706.03762 (visited on 08/13/2025).

[26] Vijay Kanade. *Markov Decision Process Definition, Working, and Examples - Spiceworks*. en-US. Dec. 2022. URL: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-markov-decision-process/ (visited on 07/30/2025).

[27] Ziyu Wang et al. *Dueling Network Architectures for Deep Reinforcement Learning*. arXiv:1511.06581 [cs]. Apr. 2016. DOI: 10.48550/arXiv.1511.06581. URL: http://arxiv.org/abs/1511.06581 (visited on 08/13/2025).

[28] Bing Yang et al. "Deep reinforcement learning based on transformer and U-Net framework for stock trading". In: *Knowledge-Based Systems* 262 (Feb. 2023), p. 110211. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2022.110211. URL: https://www.sciencedirect.com/science/article/pii/S0950705122013077 (visited on 07/30/2025).

[29] Boyu Zhang et al. *Enhancing Financial Sentiment Analysis via Retrieval Augmented Large Language Models*. arXiv:2310.04027 [cs]. Nov. 2023. DOI: 10.48550/arXiv.2310.04027. URL: http://arxiv.org/abs/2310.04027 (visited on 08/13/2025).

[30] Zihao Zhang, Stefan Zohren, and Stephen Roberts. *Deep Reinforcement Learning for Trading*. arXiv:1911.10107 [q-fin]. Nov. 2019. DOI: 10.48550/arXiv.1911.10107. URL: http://arxiv.org/abs/1911.10107 (visited on 08/14/2025).

[31] Chang Zong and Hang Zhou. *Stock Movement Prediction with Multimodal Stable Fusion via Gated Cross-Attention Mechanism*. arXiv:2406.06594 [q-fin]. Dec. 2024. DOI: 10.48550/arXiv.2406.06594. URL: http://arxiv.org/abs/2406.06594 (visited on 08/13/2025).

Chapter 7

Appendices

7.1 Detailed Performance Metrics

This appendix provides the complete performance metrics for each of the four DRL agent configurations across the ten-stock universe.

Table 7.1: Detailed Performance Metrics by Stock and Agent

Ticker	Agent	CAGR (%)	Sharpe Ratio	Sortino Ratio	MDD (%)
BABA	Baseline	**-20.91**	**-0.36**	**-0.66**	-58.23
	Indicator-LLM	-24.30	-0.49	-0.90	**-54.59**
	Combined-Fusion	-32.38	-0.54	-1.01	-67.41
	Headline-LLM	-34.19	-0.47	-0.87	-73.25
GE	Baseline	32.27	1.31	1.82	**-26.40**
	Indicator-LLM	38.34	**1.31**	1.83	-38.51
	Combined-Fusion	35.84	1.09	1.61	-46.52
	Headline-LLM	**39.04**	1.26	**1.84**	-34.90
GOOG	Baseline	14.74	0.59	0.78	-27.85
	Indicator-LLM	14.61	0.57	0.75	-32.03
	Combined-Fusion	14.61	0.57	0.75	-32.03
	Headline-LLM	**15.29**	**0.61**	**0.77**	**-26.71**
KO	Baseline	5.19	0.34	0.30	-36.66

Continued on next page

Table 7.1 – continued from previous page

Ticker	Agent	CAGR (%)	Sharpe Ratio	Sortino Ratio	MDD (%)
	Indicator-LLM	6.15	0.39	0.35	-36.92
	Combined-Fusion	5.54	0.36	0.32	-36.95
	Headline-LLM	**10.49**	**0.77**	**0.84**	**-12.95**
MRK	Baseline	6.63	0.40	0.41	-26.74
	Indicator-LLM	1.16	0.16	0.09	-33.71
	Combined-Fusion	10.64	0.56	0.65	-24.08
	Headline-LLM	**10.92**	**0.57**	**0.65**	**-23.88**
MS	Baseline	15.46	0.61	0.75	**-47.07**
	Indicator-LLM	20.21	0.68	0.90	-49.01
	Combined-Fusion	**26.70**	**0.86**	**1.10**	-51.32
	Headline-LLM	19.61	0.69	0.89	-51.28
NVDA	Baseline	73.21	1.33	2.16	-66.18
	Indicator-LLM	73.70	1.33	2.19	-65.18
	Combined-Fusion	71.97	1.31	2.01	-65.51
	Headline-LLM	**74.72**	**1.40**	**2.31**	**-63.14**
QQQ	Baseline	6.05	0.92	0.81	-8.31
	Indicator-LLM	4.77	0.68	0.60	**-6.65**
	Combined-Fusion	**26.74**	**1.06**	**1.46**	-35.10
	Headline-LLM	25.01	1.01	1.37	-35.10
T	Baseline	1.37	0.18	0.13	**-29.26**
	Indicator-LLM	0.60	0.14	0.07	-30.23
	Combined-Fusion	**2.34**	**0.22**	0.18	-31.12
	Headline-LLM	-2.31	0.04	-0.04	-31.26
WFC	Baseline	18.47	**0.75**	**1.03**	-33.76
	Indicator-LLM	16.22	0.64	0.90	-36.43
	Combined-Fusion	17.25	0.68	0.98	-34.83
	Headline-LLM	**18.98**	0.74	0.95	**-30.35**

7.2 Equity Curve Plots

This appendix provides the complete set of equity curve plots for each of the ten stocks in the experimental universe, organized by the four experimental configurations. It also includes the combined equity curve for each stock, showing the performance of all four DRL agents simultaneously against the Buy & Hold benchmark.

7.2.1 Baseline Agent Performance (Experiment I)

The following figures display the out-of-sample performance of the Baseline DRL-UTrans agent, which was trained solely on quantitative technical indicators, for each stock in the test universe.

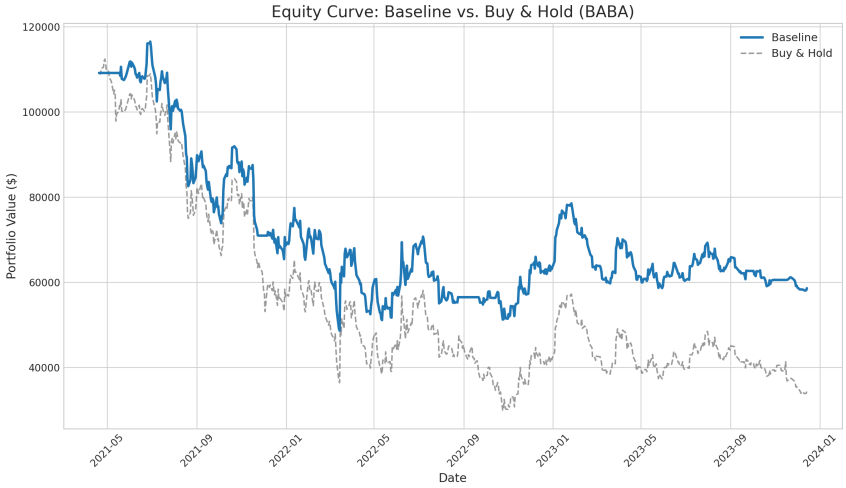


Figure 7.1: Baseline Agent Equity Curve for BABA.

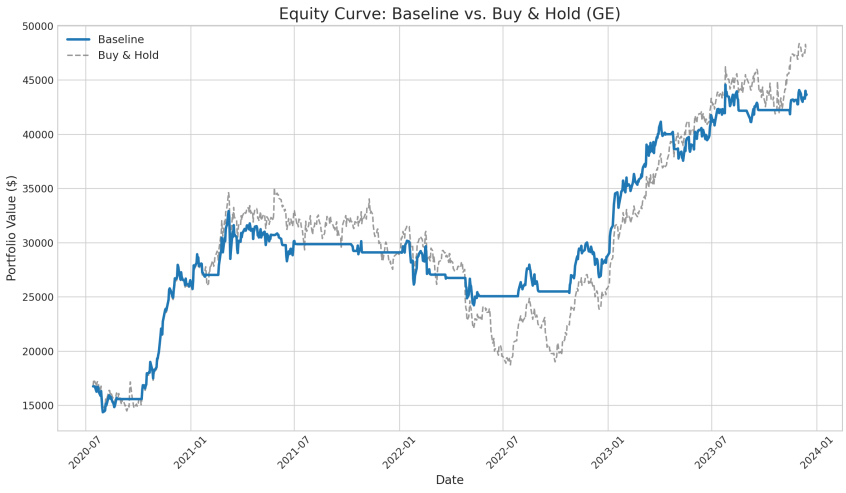


Figure 7.2: Baseline Agent Equity Curve for GE.

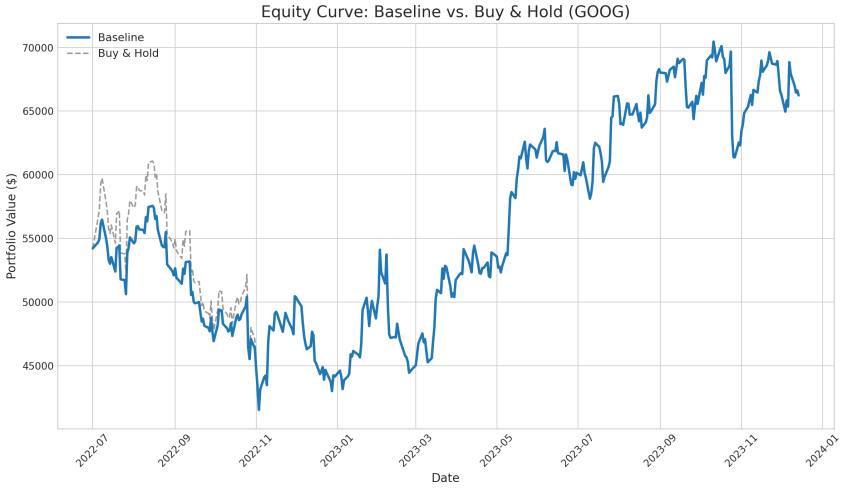


Figure 7.3: Baseline Agent Equity Curve for GOOG.

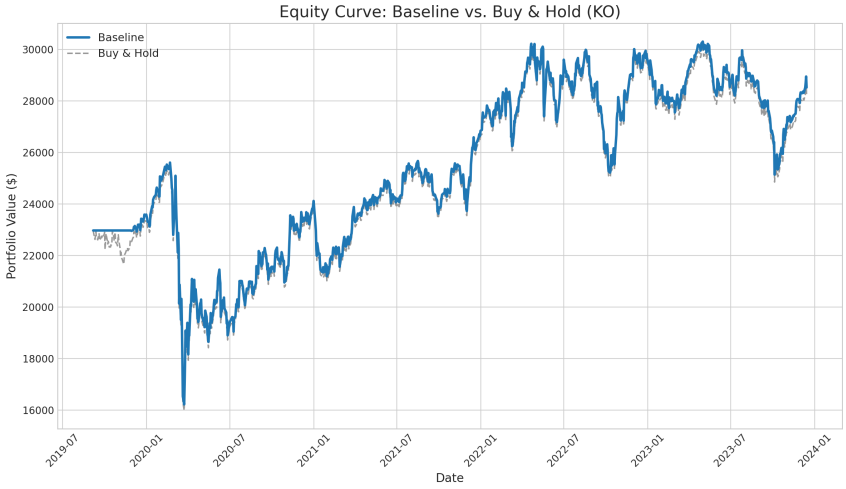


Figure 7.4: Baseline Agent Equity Curve for KO.

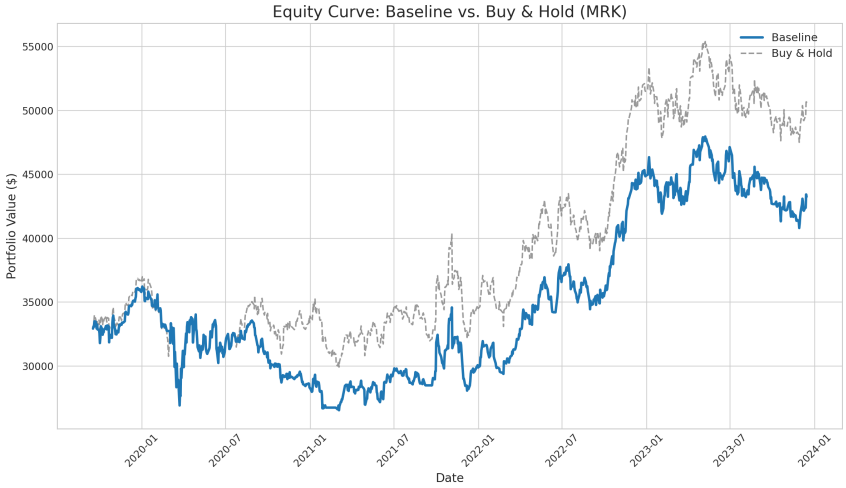


Figure 7.5: Baseline Agent Equity Curve for MRK.

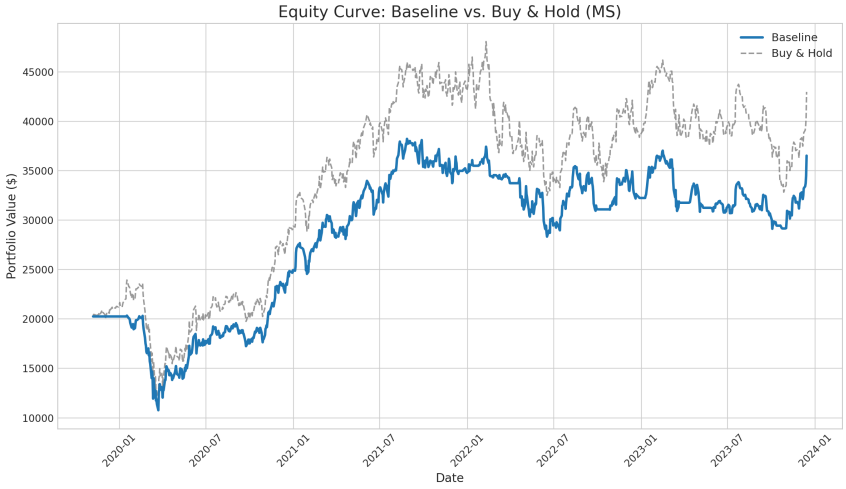


Figure 7.6: Baseline Agent Equity Curve for MS.

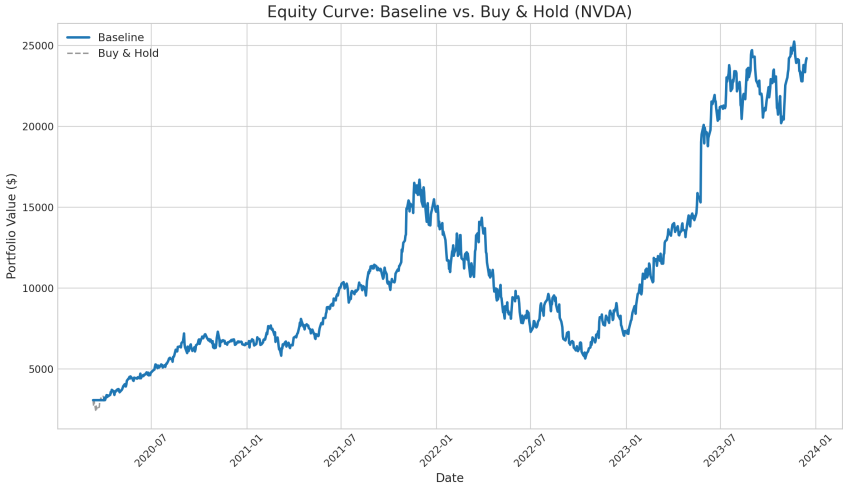


Figure 7.7: Baseline Agent Equity Curve for NVDA.

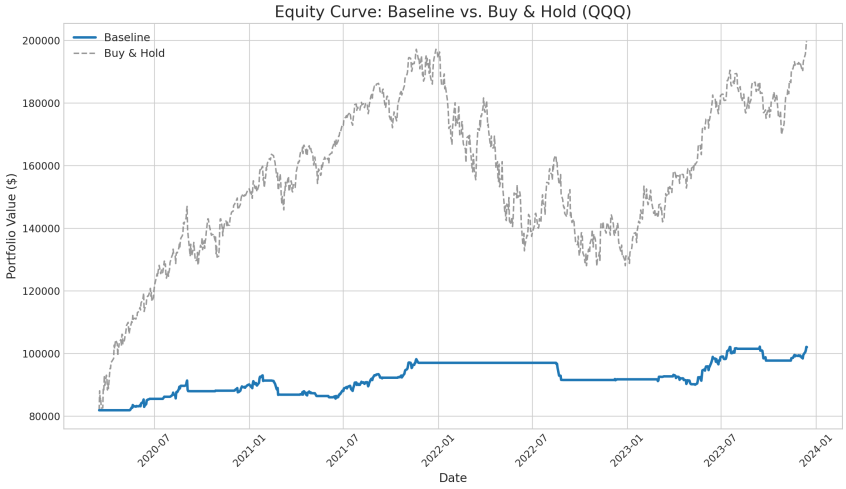


Figure 7.8: Baseline Agent Equity Curve for QQQ.

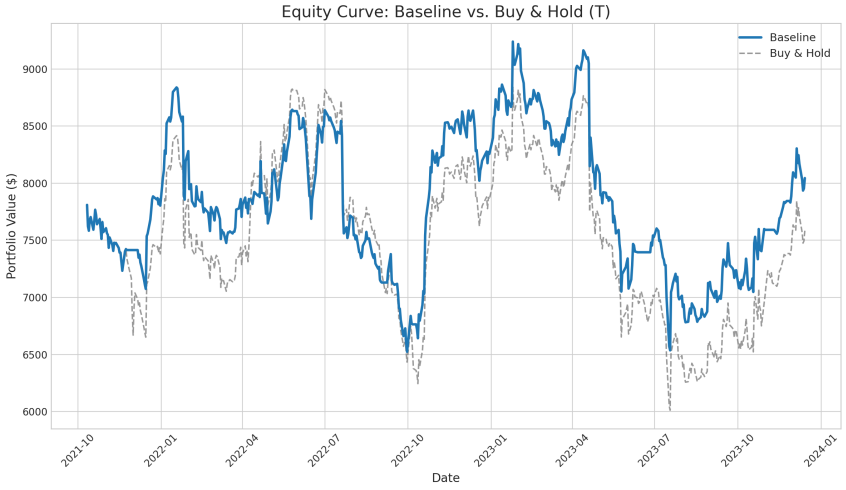


Figure 7.9: Baseline Agent Equity Curve for T.

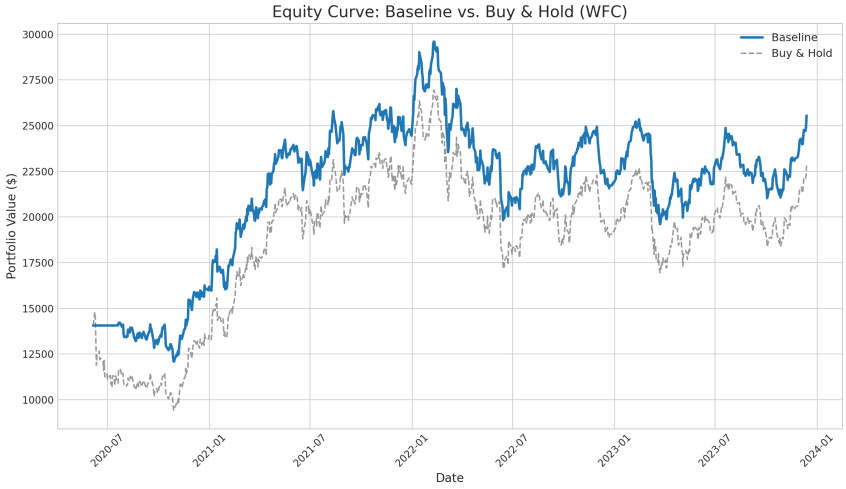


Figure 7.10: Baseline Agent Equity Curve for WFC.

7.2.2 Indicator-LLM Fusion Performance (Experiment II)

The following figures display the performance of the Indicator-LLM Fusion agent, which augmented the baseline Agent with a sentiment score derived from technical indicator summaries.

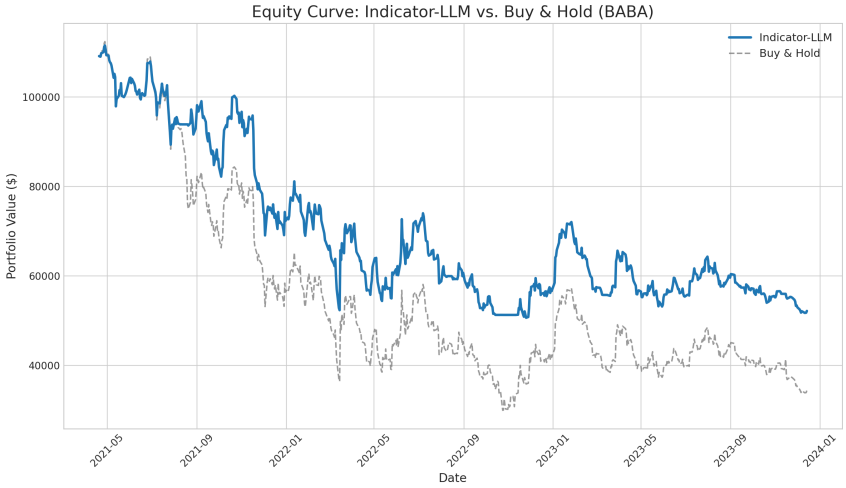


Figure 7.11: Indicator-LLM Agent Equity Curve for BABA.

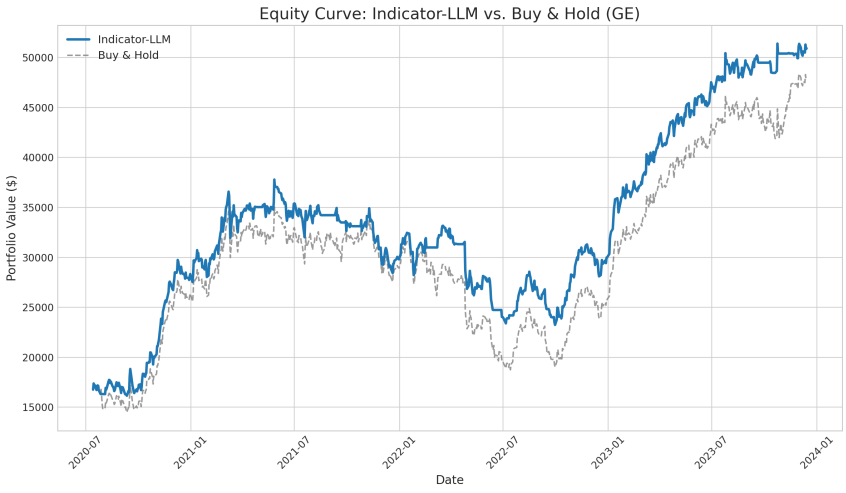


Figure 7.12: Indicator-LLM Agent Equity Curve for GE.

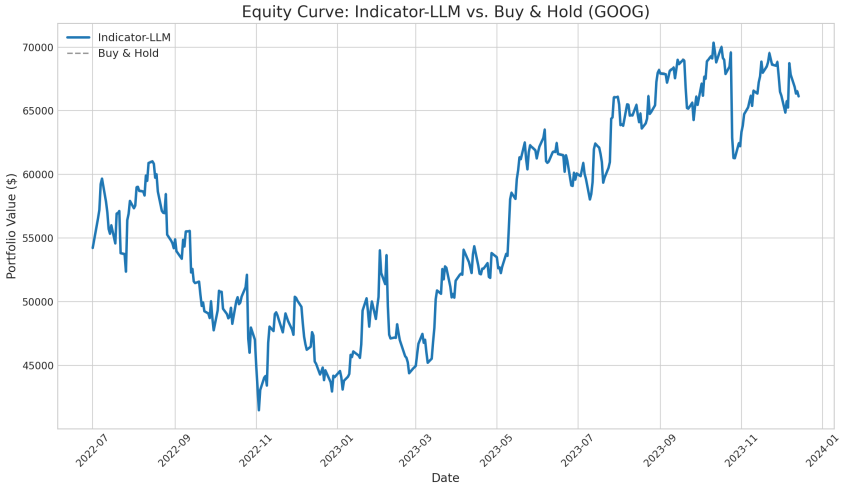


Figure 7.13: Indicator-LLM Agent Equity Curve for GOOG.

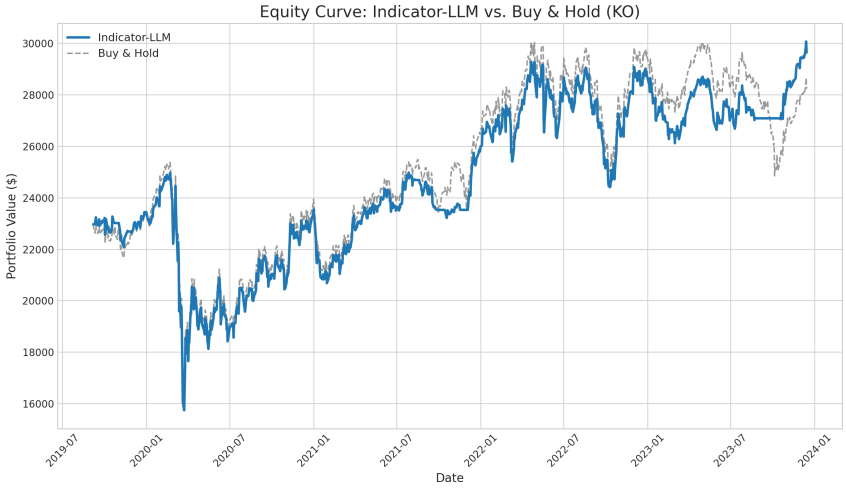


Figure 7.14: Indicator-LLM Agent Equity Curve for KO.

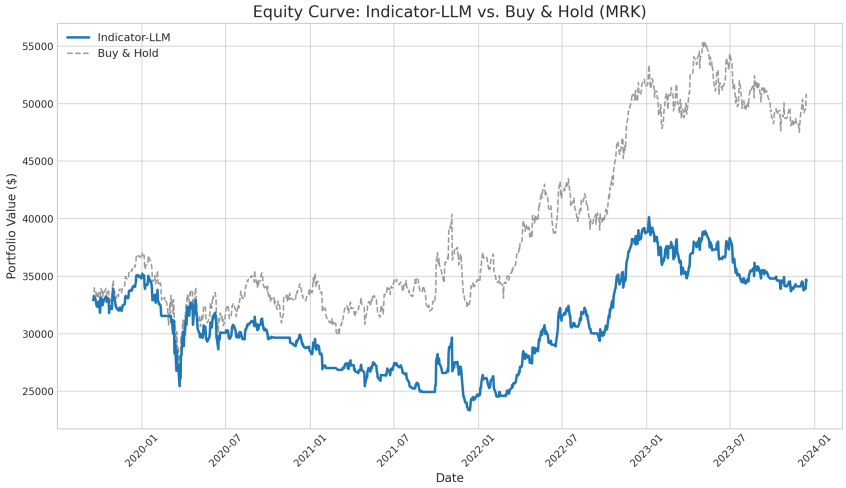


Figure 7.15: Indicator-LLM Agent Equity Curve for MRK.

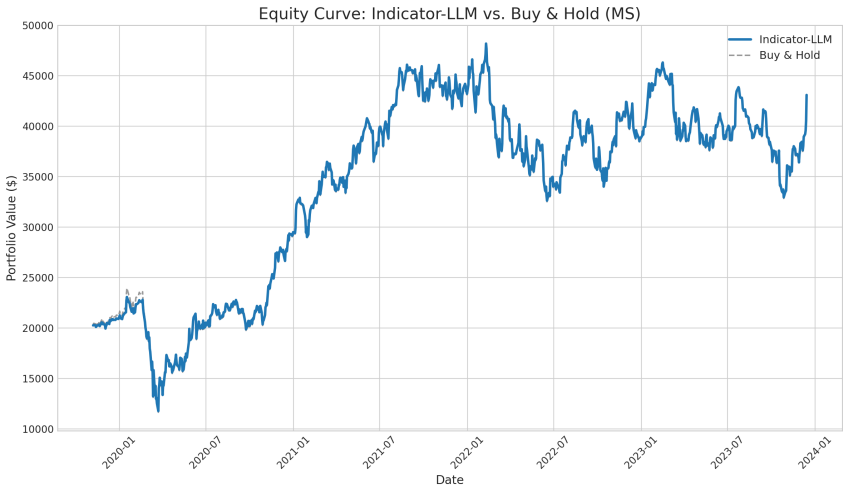


Figure 7.16: Indicator-LLM Agent Equity Curve for MS.

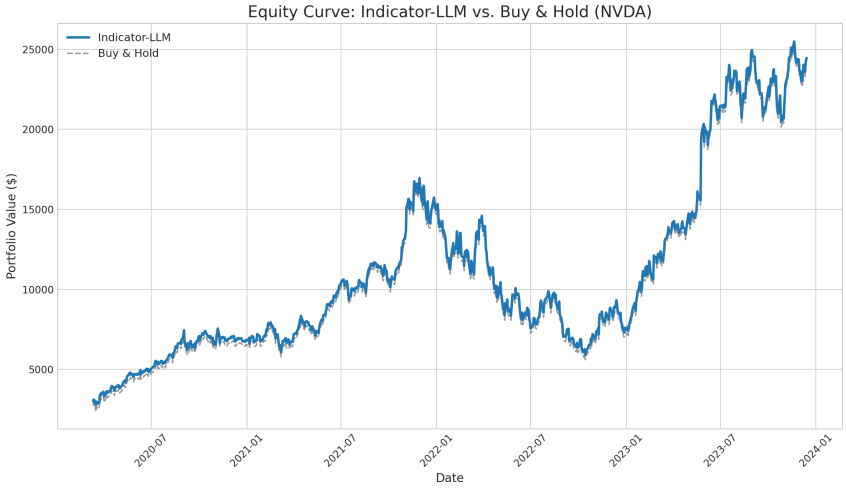


Figure 7.17: Indicator-LLM Agent Equity Curve for NVDA.

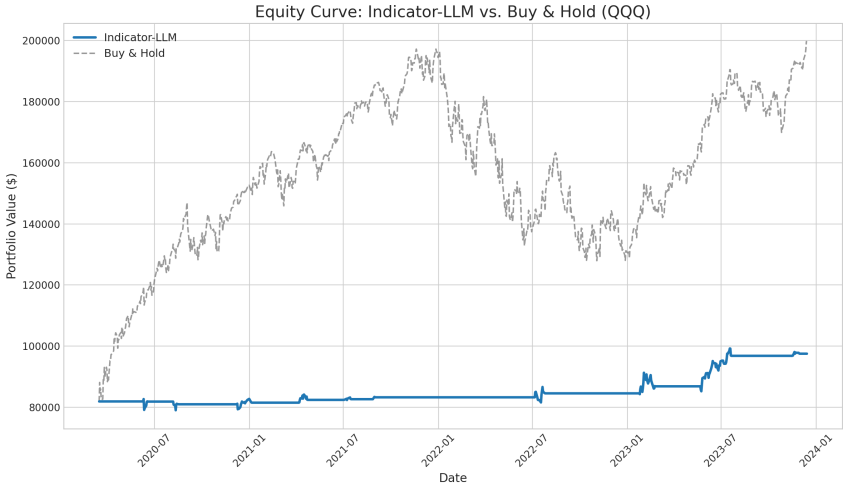


Figure 7.18: Indicator-LLM Agent Equity Curve for QQQ.

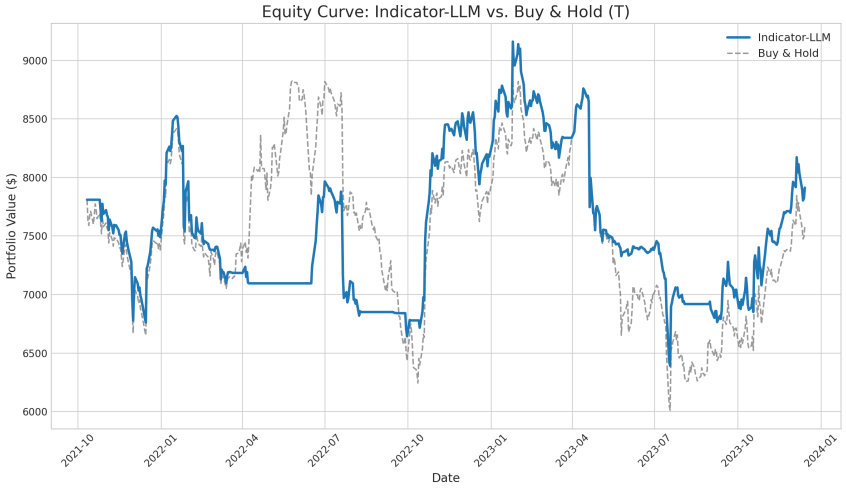


Figure 7.19: Indicator-LLM Agent Equity Curve for T.

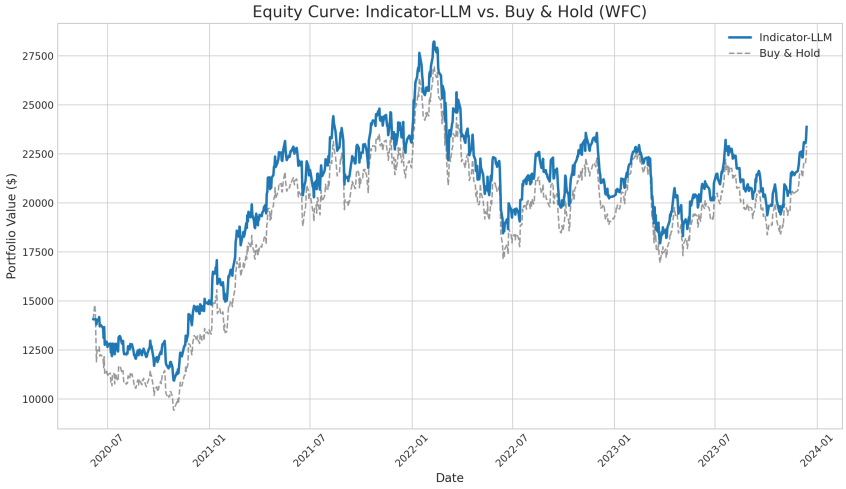


Figure 7.20: Indicator-LLM Agent Equity Curve for WFC.

7.2.3 Headline-LLM Fusion Performance (Experiment III)

The following figures display the performance of the Headline-LLM Fusion agent, which augmented the baseline Agent with sentiment features derived from daily news headlines.

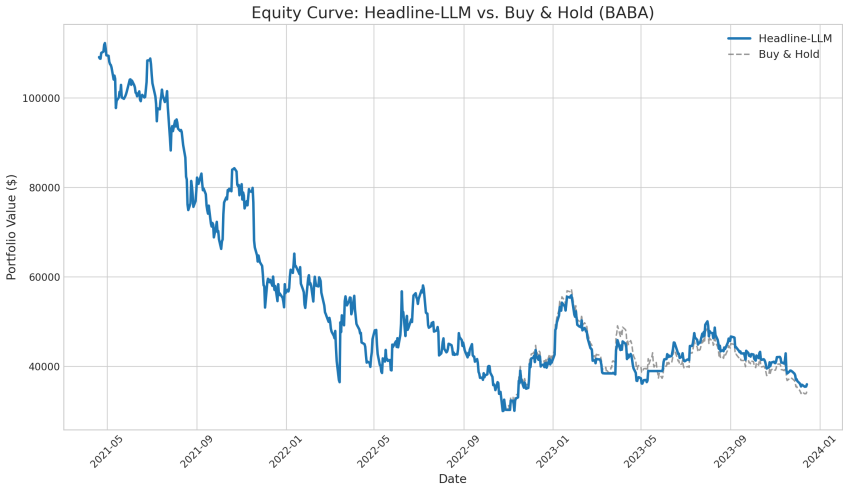


Figure 7.21: Headline-LLM Agent Equity Curve for BABA.

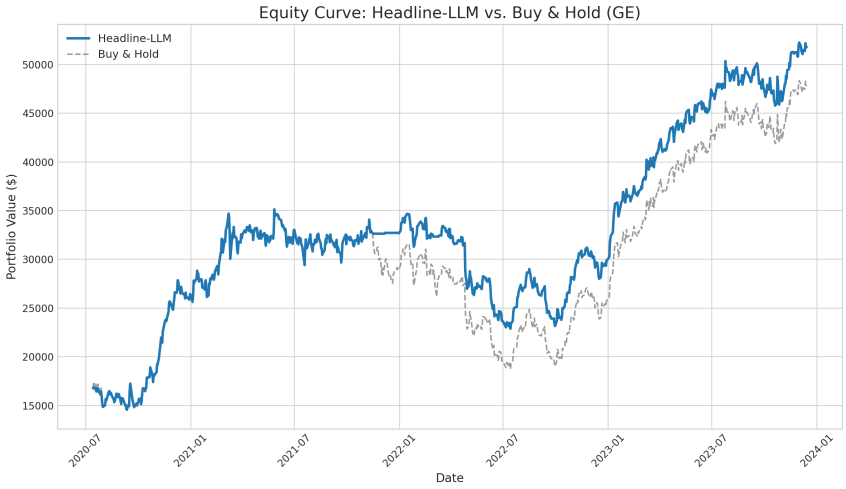


Figure 7.22: Headline-LLM Agent Equity Curve for GE.

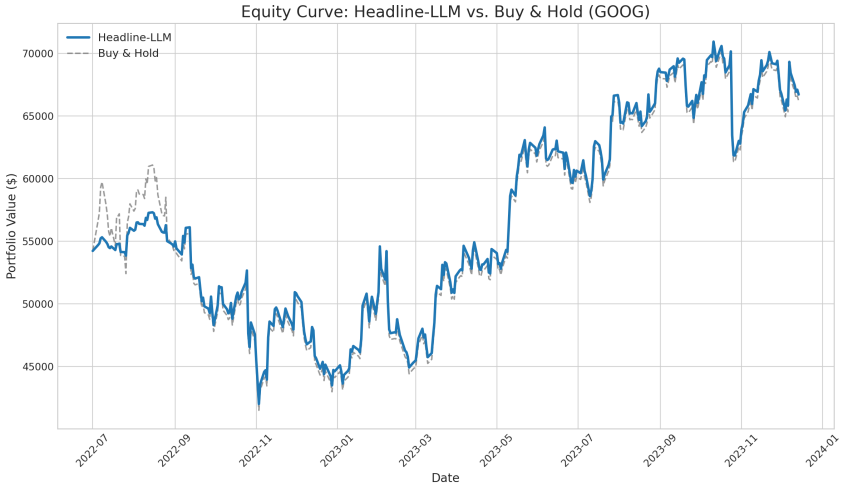


Figure 7.23: Headline-LLM Agent Equity Curve for GOOG.

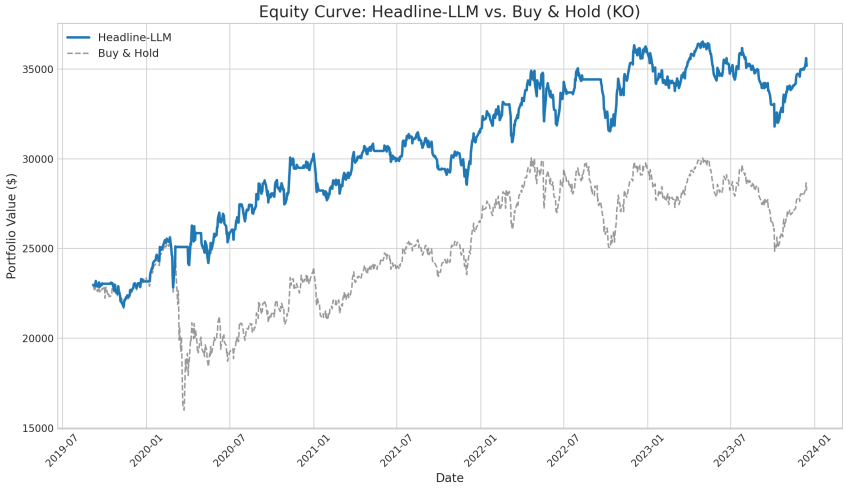


Figure 7.24: Headline-LLM Agent Equity Curve for KO.

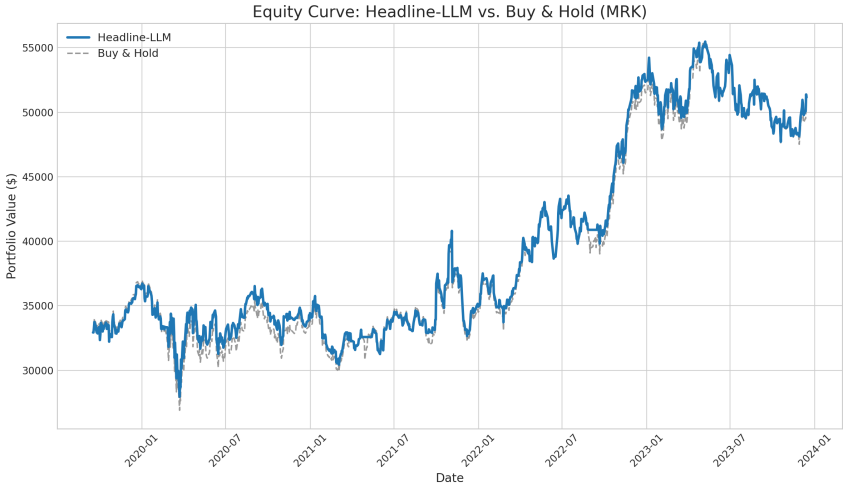


Figure 7.25: Headline-LLM Agent Equity Curve for MRK.

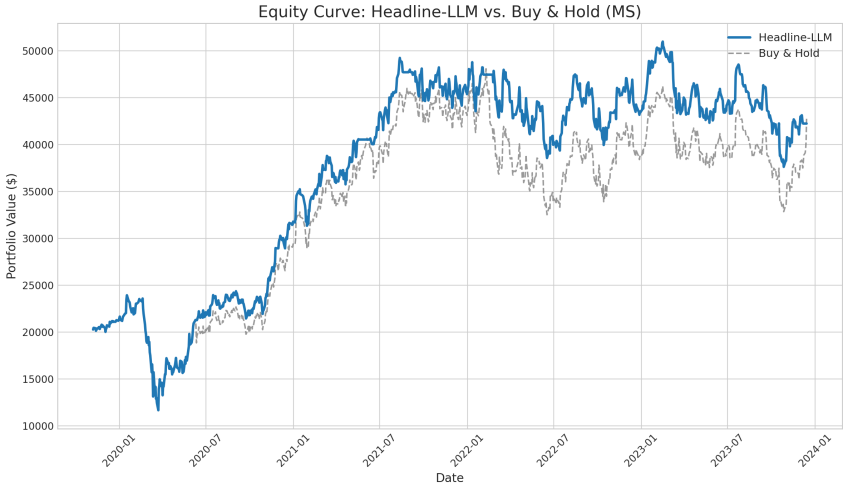


Figure 7.26: Headline-LLM Agent Equity Curve for MS.

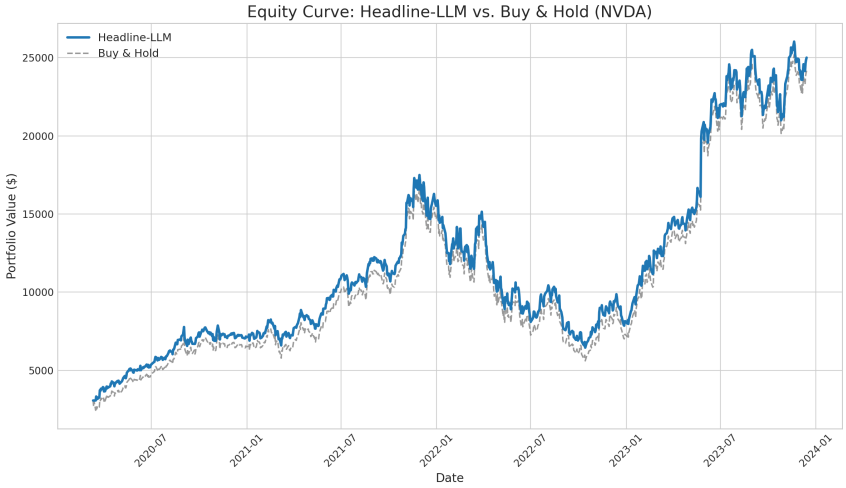


Figure 7.27: Headline-LLM Agent Equity Curve for NVDA.

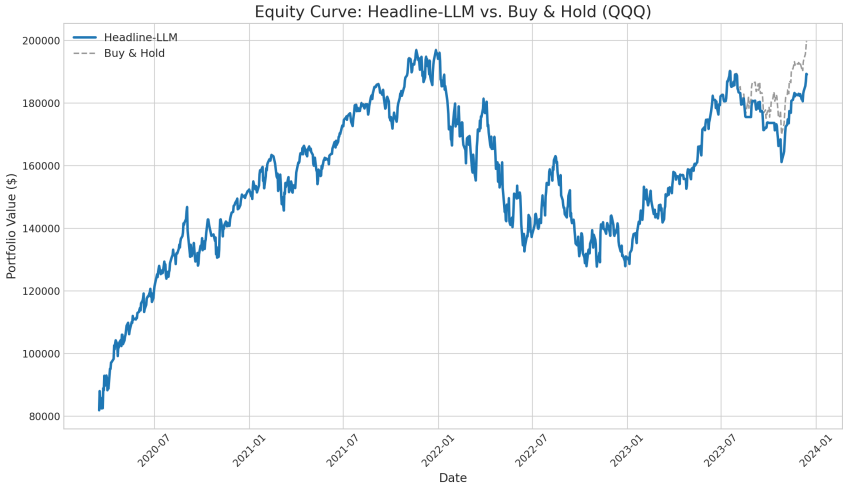


Figure 7.28: Headline-LLM Agent Equity Curve for QQQ.

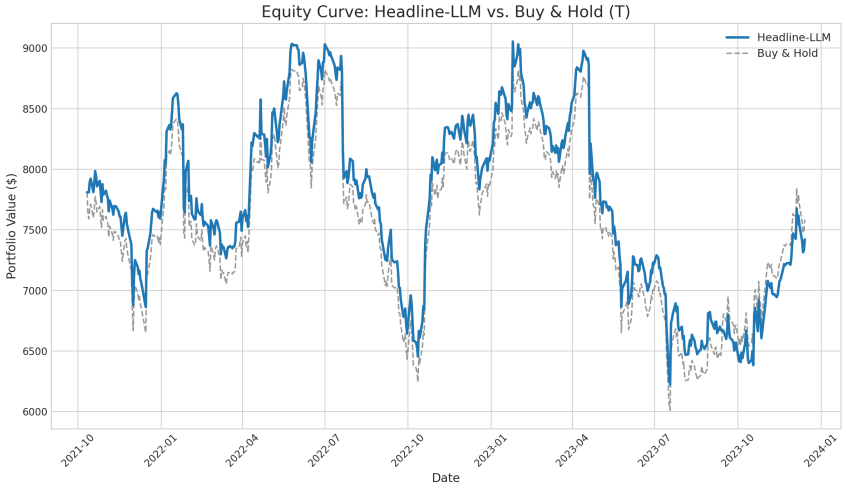


Figure 7.29: Headline-LLM Agent Equity Curve for T.

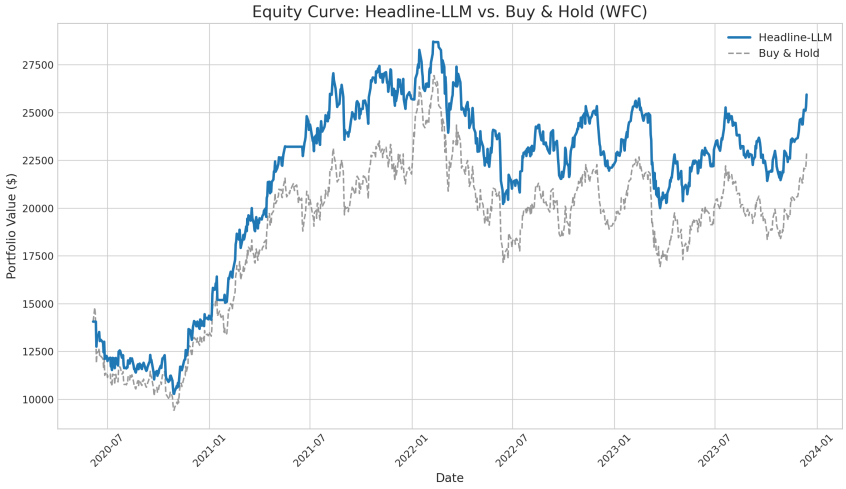


Figure 7.30: Headline-LLM Agent Equity Curve for WFC.

7.2.4 Combined-Fusion Agent Performance (Experiment IV)

The following figures display the performance of the Combined-Fusion agent, which was trained on all available features: quantitative, indicator sentiment, and headline sentiment.

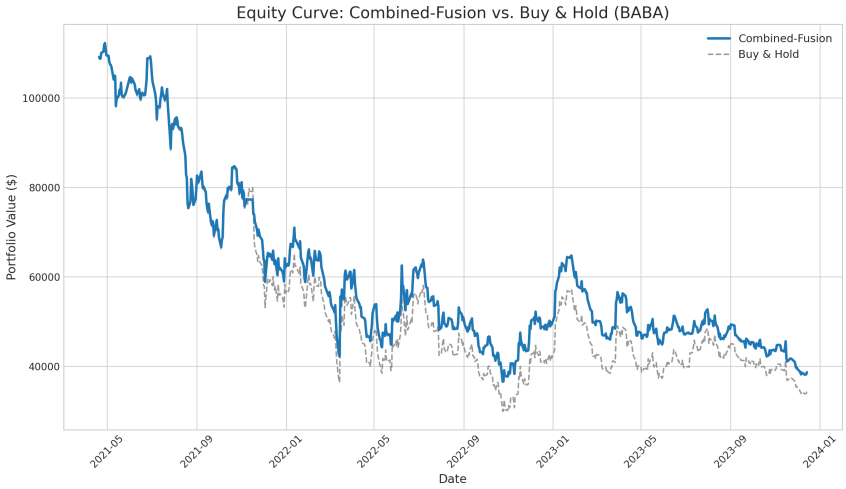


Figure 7.31: Combined-Fusion Agent Equity Curve for BABA.

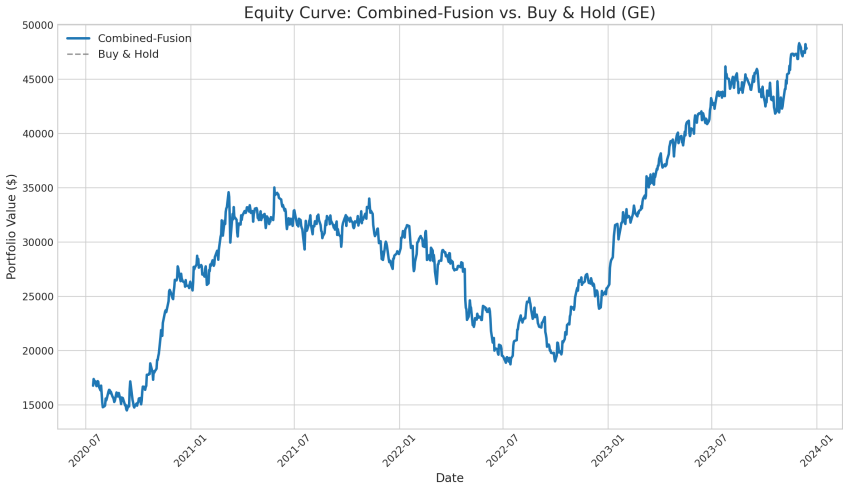


Figure 7.32: Combined-Fusion Agent Equity Curve for GE.

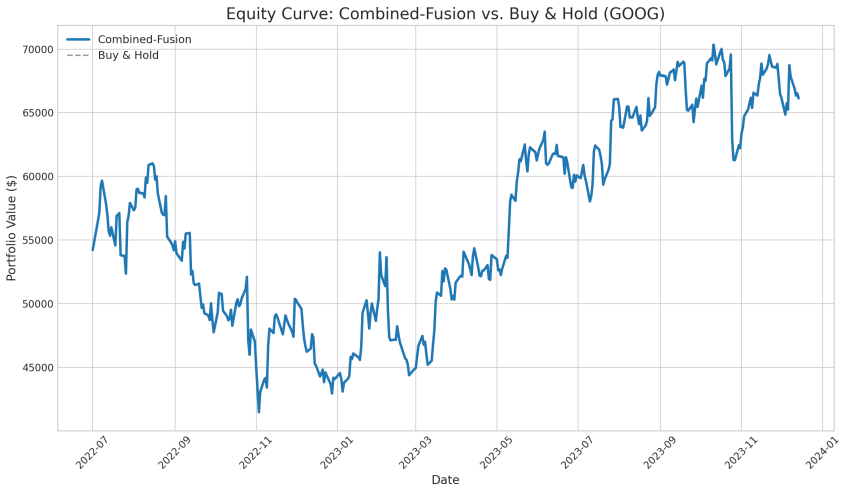


Figure 7.33: Combined-Fusion Agent Equity Curve for GOOG.

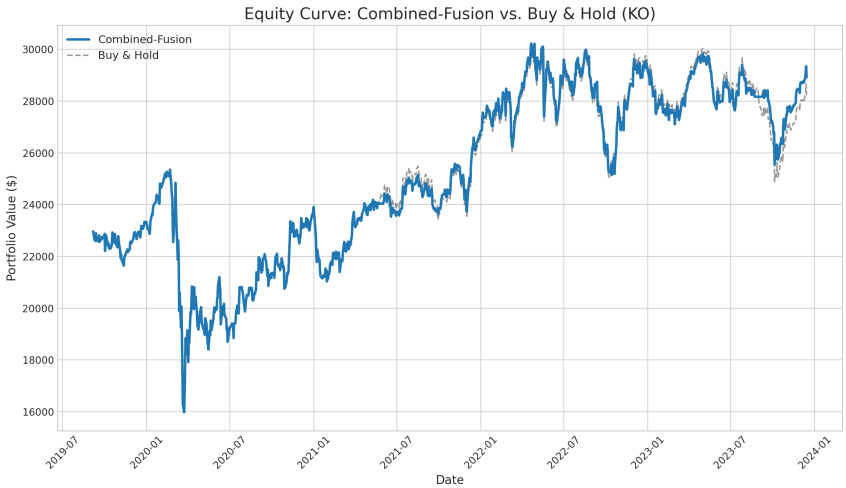


Figure 7.34: Combined-Fusion Agent Equity Curve for KO.

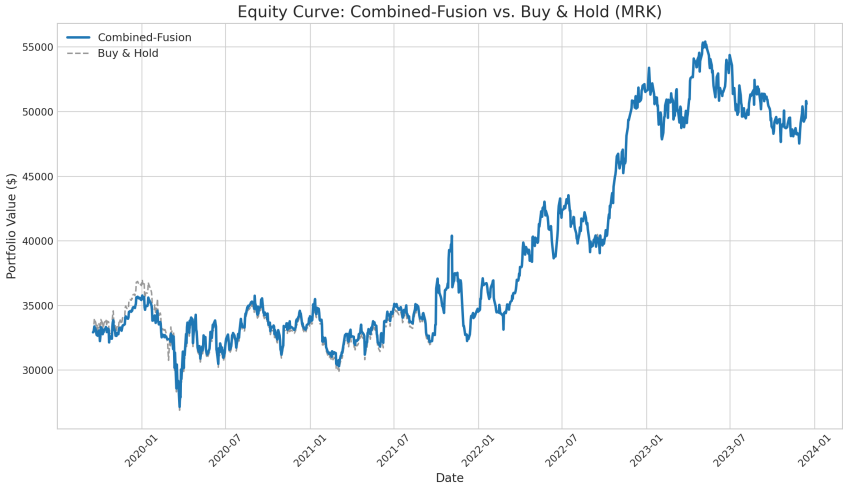


Figure 7.35: Combined-Fusion Agent Equity Curve for MRK.

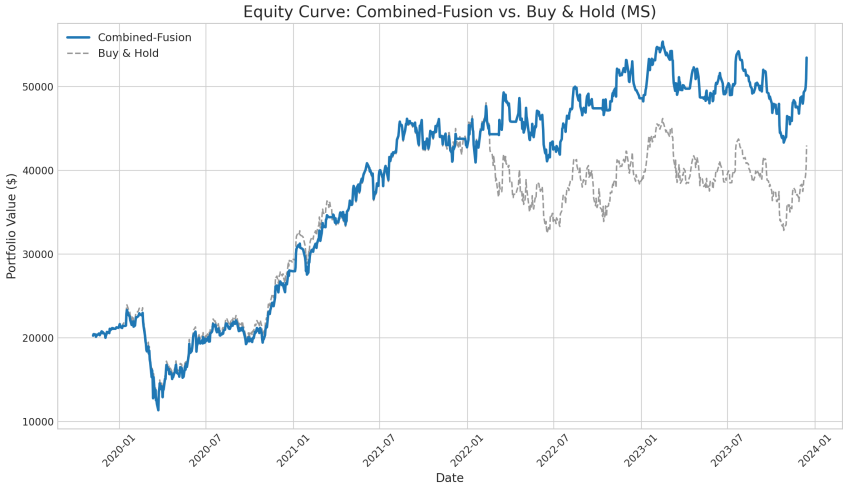


Figure 7.36: Combined-Fusion Agent Equity Curve for MS.

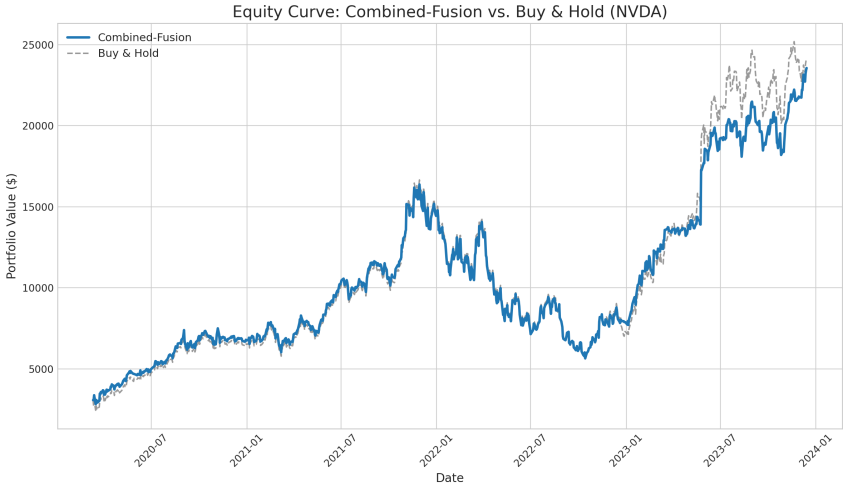


Figure 7.37: Combined-Fusion Agent Equity Curve for NVDA.

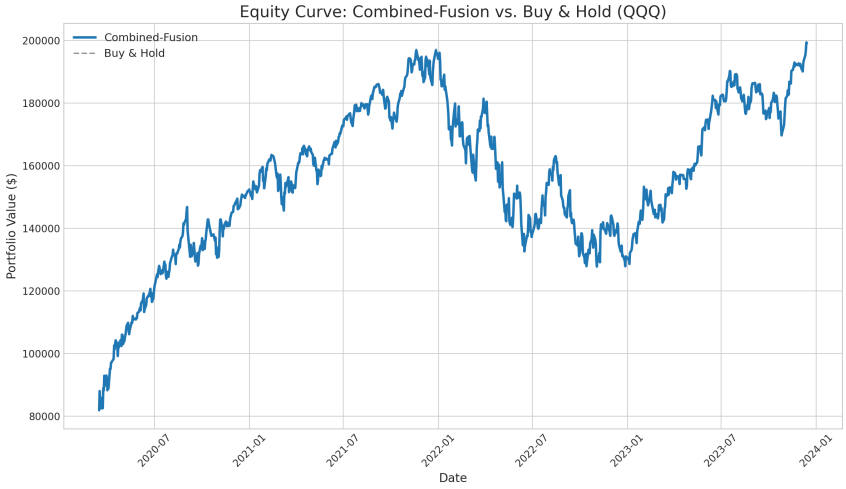


Figure 7.38: Combined-Fusion Agent Equity Curve for QQQ.

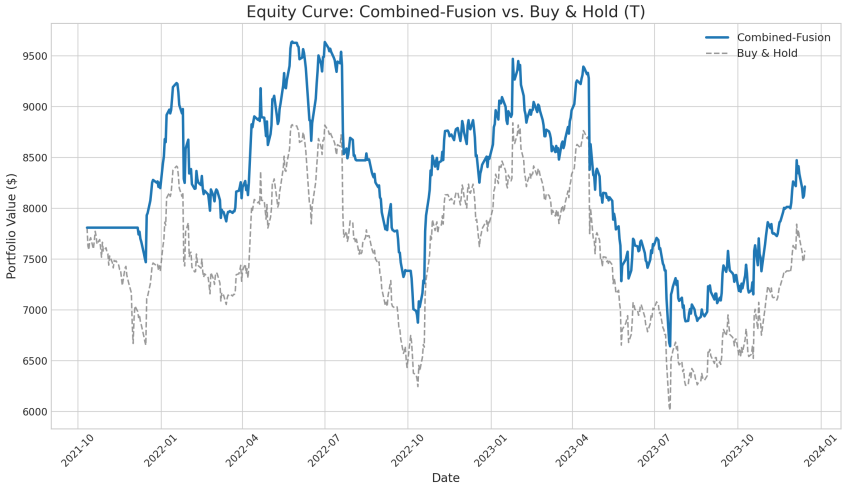


Figure 7.39: Combined-Fusion Agent Equity Curve for T.

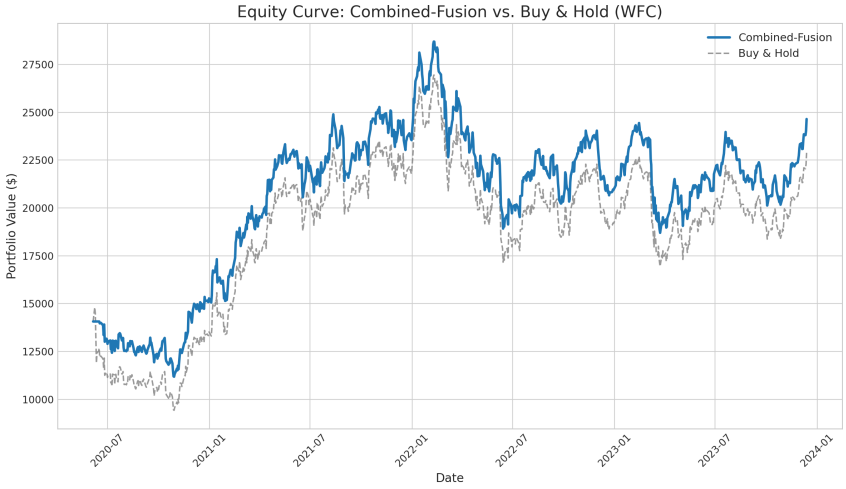


Figure 7.40: Combined-Fusion Agent Equity Curve for WFC.

7.2.5 Combined Equity Curves by Stock

The following figures display the performance of all four DRL agent configurations simultaneously on a single plot for each stock, allowing for direct visual comparison of their behavior over the test period.

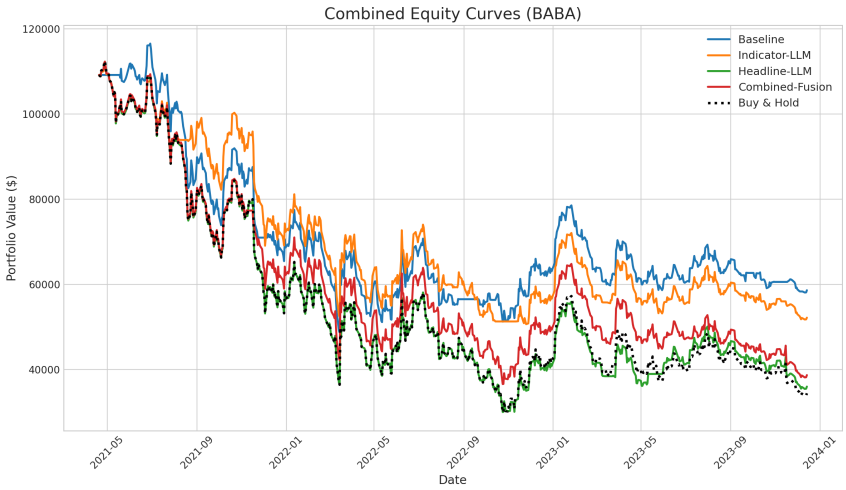


Figure 7.41: Combined Agent Equity Curves for BABA.

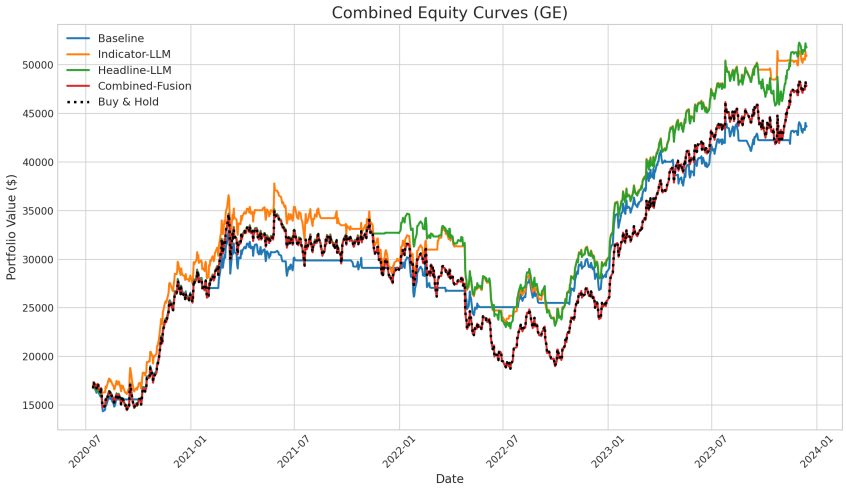


Figure 7.42: Combined Agent Equity Curves for GE.

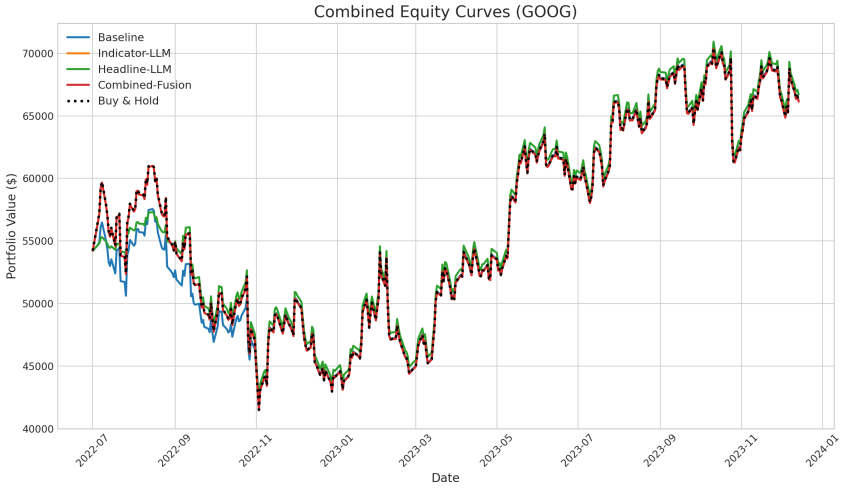


Figure 7.43: Combined Agent Equity Curves for GOOG.

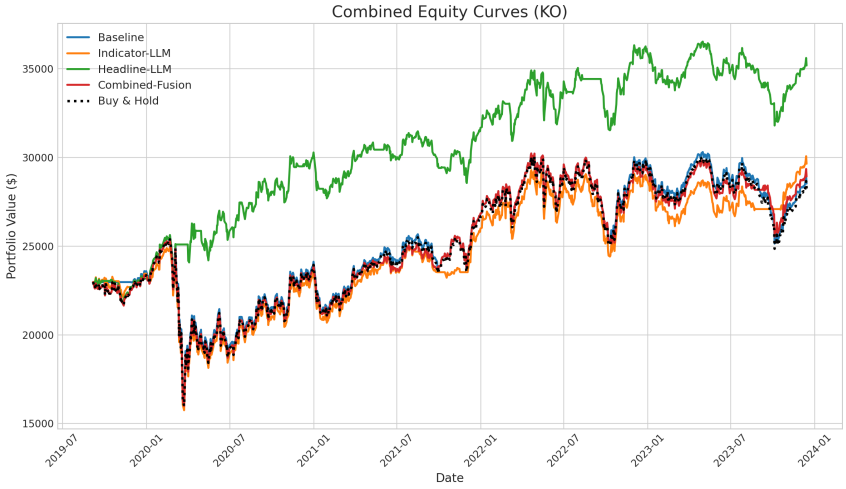


Figure 7.44: Combined Agent Equity Curves for KO.

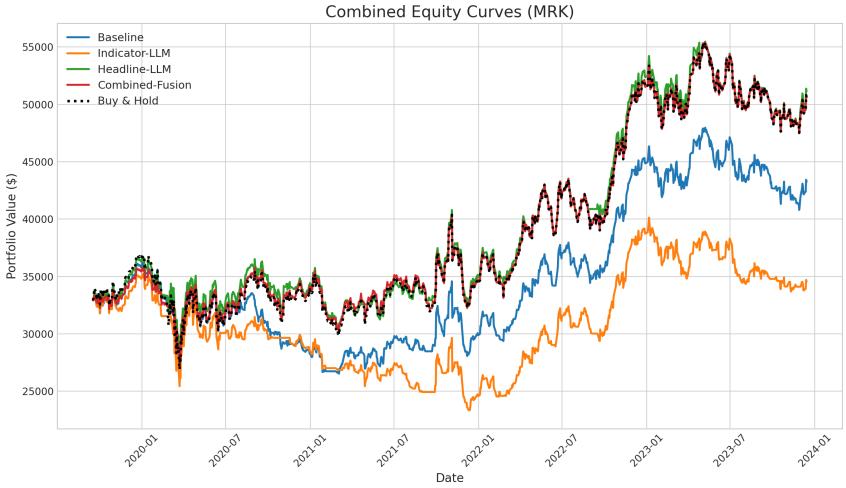


Figure 7.45: Combined Agent Equity Curves for MRK.

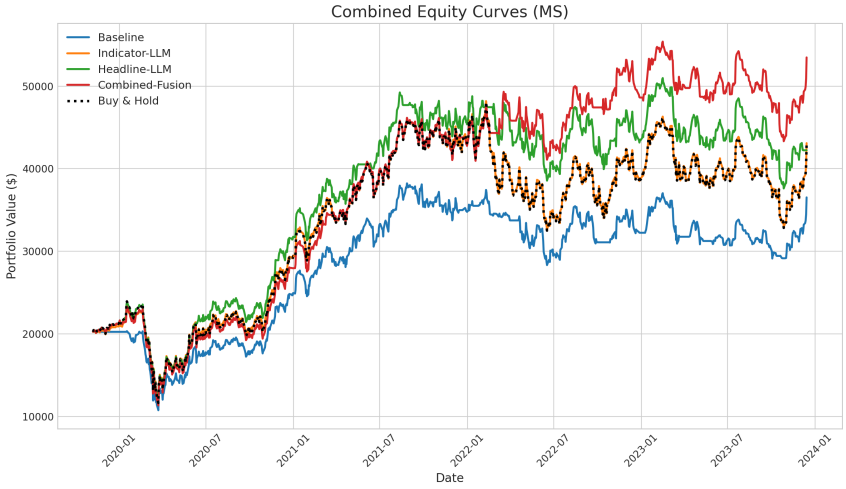


Figure 7.46: Combined Agent Equity Curves for MS.

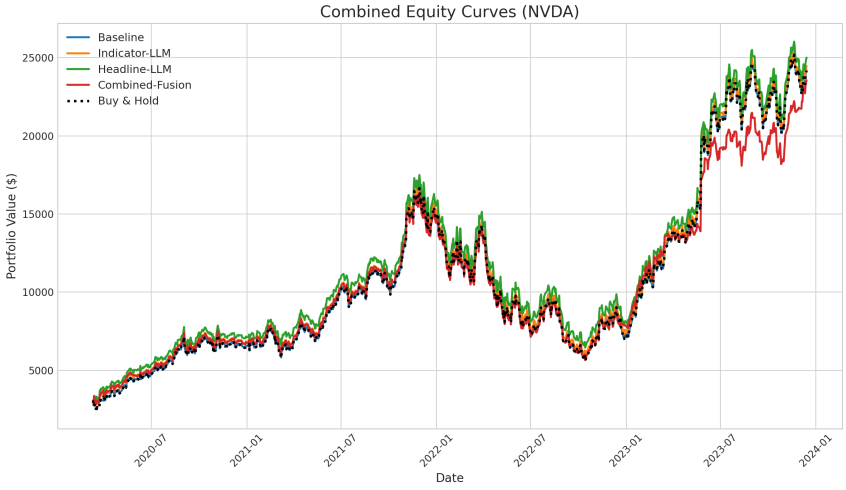


Figure 7.47: Combined Agent Equity Curves for NVDA.

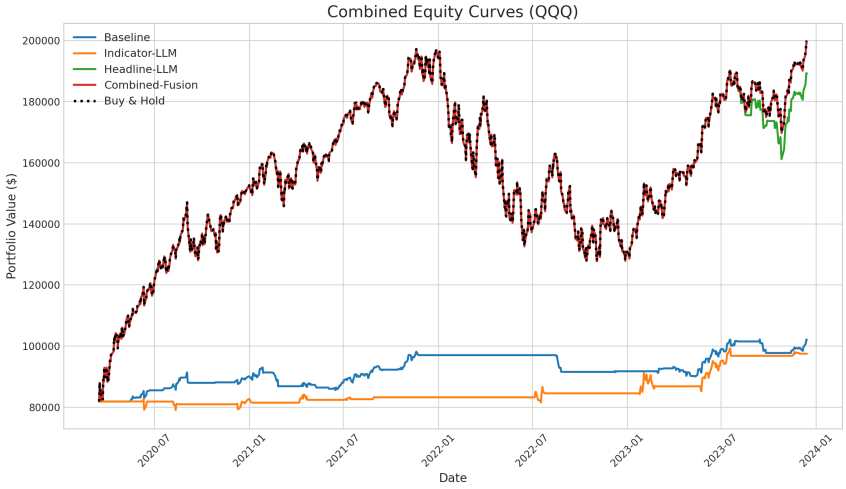


Figure 7.48: Combined Agent Equity Curves for QQQ.

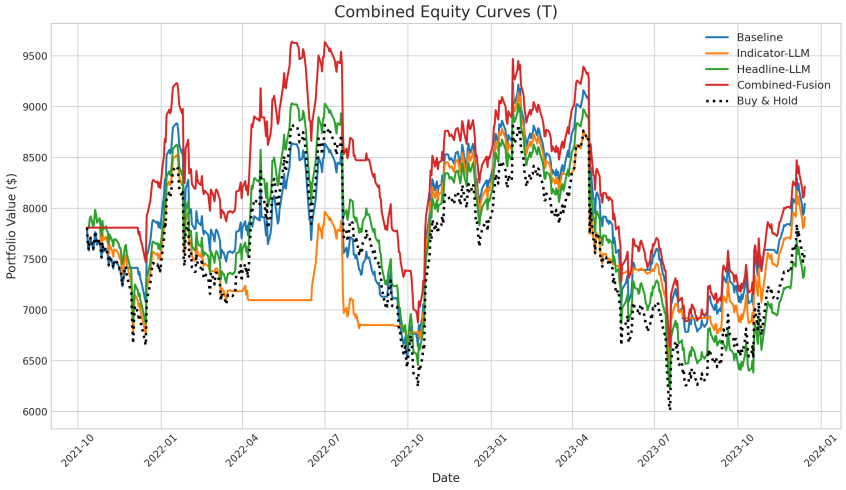


Figure 7.49: Combined Agent Equity Curves for T.

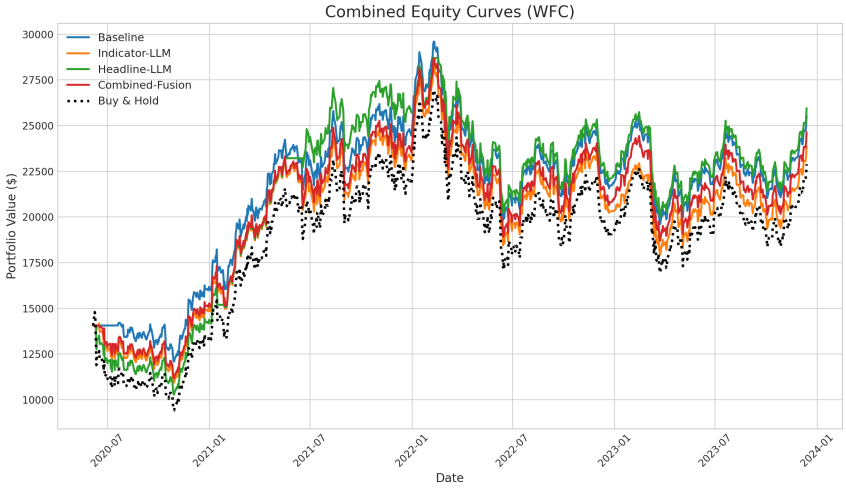


Figure 7.50: Combined Agent Equity Curves for WFC.

7.3 Performance Variance

This appendix section includes the table of the run-to-run performance variance across all the experiments.

Table 7.2: Run-to-Run Performance Variance for CAGR & Sharpe Ratio Across Three Seeds

Ticker	Experiment	Metric	Best Run	Worst Run	Range (Δ)	Std. Dev.
BABA	Baseline	CAGR (%)	-20.91	-30.87	9.96	4.20
		Sharpe Ratio	-0.36	-0.59	0.23	0.10
	Headline	CAGR (%)	-34.19	-38.93	4.74	2.01
		Sharpe Ratio	-0.47	-0.66	0.19	0.08
	Techsent	CAGR (%)	-24.30	-36.33	12.03	5.22
		Sharpe Ratio	-0.49	-0.57	0.08	0.04
	All	CAGR (%)	-32.38	-35.27	2.89	1.25
		Sharpe Ratio	-0.49	-0.54	0.05	0.02
GE	Baseline	CAGR (%)	32.27	22.26	10.01	4.19
		Sharpe Ratio	1.31	0.81	0.50	0.21
	Headline	CAGR (%)	39.04	29.18	9.86	4.22
		Sharpe Ratio	1.26	0.96	0.30	0.13
	Techsent	CAGR (%)	38.34	34.48	3.86	1.63
		Sharpe Ratio	1.31	1.07	0.24	0.10
	All	CAGR (%)	35.84	33.65	2.19	0.98
		Sharpe Ratio	1.10	1.09	0.01	0.01
GOOG	Baseline	CAGR (%)	14.74	2.49	12.24	5.48
		Sharpe Ratio	0.59	0.24	0.35	0.16
	Headline	CAGR (%)	15.29	9.14	6.15	2.65
		Sharpe Ratio	0.61	0.43	0.18	0.08
	Techsent	CAGR (%)	14.61	11.57	3.04	1.33
		Sharpe Ratio	0.57	0.49	0.08	0.04

Continued on next page

Table 7.2 – continued from previous page

Ticker	Experiment	Metric	Best Run	Worst Run	Range (Δ)	Std. Dev.
	All	CAGR (%)	14.61	10.59	4.02	1.77
		Sharpe Ratio	0.57	0.47	0.10	0.04
KO	Baseline	CAGR (%)	5.19	-4.51	9.70	4.22
		Sharpe Ratio	0.34	-0.27	0.61	0.26
	Headline	CAGR (%)	10.49	2.82	7.66	3.32
		Sharpe Ratio	0.77	0.24	0.53	0.23
	Techsent	CAGR (%)	6.15	0.50	5.65	2.39
		Sharpe Ratio	0.50	0.12	0.38	0.17
	All	CAGR (%)	5.54	4.92	0.62	0.27
		Sharpe Ratio	0.36	0.33	0.03	0.01
MRK	Baseline	CAGR (%)	6.63	-10.34	16.97	7.55
		Sharpe Ratio	0.40	-0.70	1.11	0.49
	Headline	CAGR (%)	10.92	8.82	2.10	0.90
		Sharpe Ratio	0.57	0.51	0.06	0.02
	Techsent	CAGR (%)	1.16	-10.41	11.57	5.03
		Sharpe Ratio	0.16	-0.70	0.86	0.37
	All	CAGR (%)	10.64	9.54	1.10	0.47
		Sharpe Ratio	0.56	0.51	0.05	0.02
MS	Baseline	CAGR (%)	15.46	3.71	11.75	5.37
		Sharpe Ratio	0.61	0.28	0.33	0.15
	Headline	CAGR (%)	19.61	16.14	3.47	1.49
		Sharpe Ratio	0.69	0.60	0.09	0.04
	Techsent	CAGR (%)	20.21	0.41	19.80	8.52
		Sharpe Ratio	0.68	0.12	0.57	0.25
	All	CAGR (%)	26.70	15.09	11.61	5.01
		Sharpe Ratio	0.86	0.57	0.29	0.12

Continued on next page

Table 7.2 – continued from previous page

Ticker	Experiment	Metric	Best Run	Worst Run	Range (Δ)	Std. Dev.
NVDA	Baseline	CAGR (%)	73.21	0.00	73.21	34.50
		Sharpe Ratio	1.33	0.00	1.33	0.62
	Headline	CAGR (%)	74.72	67.68	7.04	3.01
		Sharpe Ratio	1.40	1.26	0.13	0.06
	Techsent	CAGR (%)	73.70	0.00	73.70	34.73
		Sharpe Ratio	1.33	0.00	1.33	0.63
	All	CAGR (%)	71.97	1.63	70.35	30.56
		Sharpe Ratio	1.31	0.18	1.13	0.49
QQQ	Baseline	CAGR (%)	6.05	0.00	6.05	2.58
		Sharpe Ratio	0.92	0.00	0.92	0.39
	Headline	CAGR (%)	25.01	21.74	3.27	1.44
		Sharpe Ratio	1.01	0.95	0.07	0.03
	Techsent	CAGR (%)	4.77	1.22	3.55	1.54
		Sharpe Ratio	0.68	0.16	0.53	0.23
	All	CAGR (%)	26.74	11.39	15.35	6.57
		Sharpe Ratio	1.06	0.60	0.46	0.20
T	Baseline	CAGR (%)	1.37	-3.21	4.57	2.05
		Sharpe Ratio	0.18	-0.00	0.18	0.08
	Headline	CAGR (%)	-2.31	-5.30	2.98	1.28
		Sharpe Ratio	0.04	-0.08	0.12	0.05
	Techsent	CAGR (%)	0.60	-10.25	10.85	4.79
		Sharpe Ratio	0.14	-0.40	0.54	0.23
	All	CAGR (%)	2.34	-2.85	5.19	2.23
		Sharpe Ratio	0.22	0.03	0.19	0.08
WFC	Baseline	CAGR (%)	18.47	4.94	13.52	5.86
		Sharpe Ratio	0.75	0.32	0.42	0.18

Continued on next page

Table 7.2 – continued from previous page

Ticker	Experiment	Metric	Best Run	Worst Run	Range (Δ)	Std. Dev.
	Headline	CAGR (%)	18.98	9.98	9.01	3.90
		Sharpe Ratio	0.74	0.45	0.28	0.12
	Techsent	CAGR (%)	16.22	-9.95	26.17	11.95
		Sharpe Ratio	0.64	-0.35	0.99	0.45
	All	CAGR (%)	17.25	7.59	9.66	4.14
		Sharpe Ratio	0.68	0.39	0.29	0.13