A Reinforcement Learning Approach to Portfolio Management

Author:

Jamie Bell-Thomas

Email: ws19177@bristol.ac.uk

Student Number: 1820499

University of Bristol
Computer Science MSc
Faculty of Engineering

Submitted on 3rd September 2024

Acknowledgements

I would like to express my sincere gratitude to my academic supervisor, James Cussens, for his continued support and guidance throughout this project. His advanced knowledge of Machine Learning has been invaluable in the development and success of this research.

Declaration

> *The accompanying research project report entitled: "A Reinforcement Learning Approach to Portfolio Management" is submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science at the University of Bristol. The report is based upon independent work by the candidate. All contributions from others have been acknowledged above. The views expressed within the report are those of the author and not of the University of Bristol.*

I hereby declare that the above statements are true.

Signed (author)

.....

Full Name

.....

Date

.....

Executive Summary

This report investigates the application of reinforcement learning (RL) techniques in financial portfolio management, aiming to develop strategies that can consistently outperform traditional heuristic-based approaches. The main focus is to use RL techniques, specifically the policy-based Proximal Policy Optimisation (PPO) and the actor-critic Deep Deterministic Policy Gradient (DDPG), to improve risk-adjusted returns while managing different types of financial risk. This research aims to show that RL can effectively handle complex market conditions and achieve strong returns.

The main objectives of this research are to assess the efficacy of RL models in consistently outperforming a series of heuristic-based strategies and to evaluate the ability of these models to autonomously discover and apply both conventional and innovative portfolio management strategies. Additionally, the study aims to test the resilience and adaptability of the RL models during periods of market crises and high volatility, ensuring they can maintain stability and effectively manage risk. Another key objective is to analyse how different reward structures influence the models' strategic decisions and their success in achieving specified financial goals. These objectives provide a comprehensive framework for assessing the performance and strategic flexibility of RL models in portfolio management.

This study implemented a novel data pre-processing technique, integrating the Capital Asset Pricing Model (CAPM), Amihud Illiquidity Ratio, linear regression, and asset volatility to describe asset's characteristics, providing a more comprehensive view of the market dynamics and enhancing the model's ability to make informed investment decisions. A range of fixed-heuristic baseline strategies was established to serve as benchmarks for comparison. These included the uniform buy-and-hold (UBAH), uniform constantly rebalanced portfolio (UCRP), follow-the-winner (FtW), and follow-the-loser (FtL) strategies, each representing different, more traditional approaches to portfolio management. A comprehensive sensitivity analysis was conducted to investigate the impact of different reward function configurations on model performance. This analysis aimed to understand how varying emphasis on different metrics of portfolio management present in the reward function, such as Sharpe Ratio, Treynor Ratio, Sortino Ratio, and portfolio entropy, affected the RL models' investment behaviors.

The results of the study provided several important insights. DDPG models consistently outperformed PPO models and all baseline strategies across the testing periods. It achieved strong absolute returns and superior risk-adjusted returns, indicating its ability to manage risk effectively while capitalising on profitable opportunities. The DDPG model demonstrated strong risk management capabilities by selecting assets with low volatility and high liquidity. This strategy enabled it to navigate periods of market turmoil, such as the 2008 financial crisis, with reduced losses compared to other strategies. Furthermore, the DDPG model maintained higher portfolio entropy, indicating a well-diversified portfolio. This diversification helped spread risk across a broader range of assets, reducing the impact of any single asset's poor performance. The sensitivity analysis revealed that different reward function configurations led to distinct changes in model behavior that aligned with the respective changes in reward structures. Furthermore, the novel pre-processing strategies proved highly effective, allowing the model to maintain high level of performance over a large asset universe, significantly improving on the current state of the art in this field.

In conclusion, this research successfully demonstrated the potential of reinforcement learning for portfolio management. The findings show that RL models, particularly DDPG, can outperform traditional heuristic strategies by effectively balancing risk and return. This research also identified areas for improvement. However the final model demonstrated limited adaptability to changing market conditions, consistently opting for a single, sensible strategy across all test periods.. Potential solutions are discussed to address these issues and expand the models' capabilities to handle larger markets in future work. Ultimately, this study provides a solid proof of concept for the application of RL in financial markets, highlighting the potential for developing sophisticated and effective investment strategies.

Contents

Figures

List of Tables

1. Background and Problem Outline

1.1 Background

1.1.1 Historical Context

The concept of financial markets has existed since the start of civilisation, with early evidence of commodity trading among the Sumerians and Babylonians around 2000 BC. However, the idea of a formal stock market was first conceived in the early 17th century, with the establishment of the Amsterdam Stock Exchange in 1602 [1]. This is considered the world's first official stock market and pioneered concepts such as the issuance and trading of shares. By doing so, it facilitated the operations of the Dutch East India Company.

The next significant step in the evolution of financial markets was the establishment of the London Stock Exchange (LSE) in 1801 [2]. This institution formalised trading of securities within the British Empire, but more importantly laid a blueprint that different countries all over the world would embody for years to come. These exchanges were characterised by physical trading floors, where traders would gather to shout their buy and sell orders in a method known as "open outcry".

The advancement of communication technologies and financial markets have gone hand-in-hand. This started in the 19th century where the introduction of the telegraph revolutionised global market operations by allowing for the rapid transmission of prices, as well as buy and sell orders over long distances. This technology was replaced by telephone-based communication which in turn was later replaced by computer technology. These progressions significantly enhanced the pace and efficiency of trading activities.

The next major paradigm shift in financial markets took place in 1971, with the creation of the National Association of Securities Dealers Automated Quotations (NASDAQ). This was the world's first electronic stock market, enabling trades to be conducted via a network of computers rather than in a physical location as had been the way since the foundation of the LSE. This was a landmark development that demonstrated the potential of computers to transform financial markets completely. The transition to electronic trading enhanced market liquidity and accessibility, therefore reducing cost and increasing the speeds of transactions. In more recent years, globalisation has had a profound impact on financial markets, with the advances in information technology and communication linking global markets together. This has created an interdependent global financial network. This connectivity has not only allowed for the rapid movement of capital across the world, but also the spread of financial crises, such as the 2008 financial crash.

The latest developments in the financial landscape are focused on the transition from human-driven decision making to automated models, through algorithmic trading and more recently artificial intelligence. The first examples of this are in the 1980's and 1990's where neural networks were explored for financial forecasting. One notable early application was the use of neural networks for predicting IBM daily stock returns [3], demonstrating that AI could potentially outperform traditional statistical methods. The progressions described in this section are visualised in Figure 1.1.

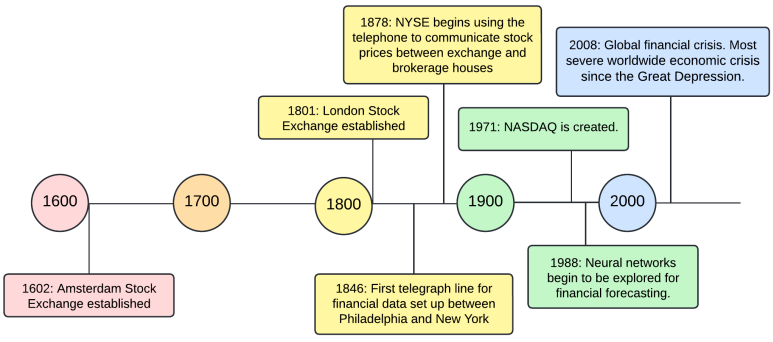


FIGURE 1.1: Evolution Timeline of Financial Markets

1.1.2 Importance of Algorithmic Trading

Algorithmic trading automates the trading process by using computer programs to execute trades based on pre-defined criteria, without human intervention. This approach revolutionised trading by leveraging mathematical models to create algorithms that can make trading decisions at speeds that would be impossible for human traders to replicate. The origins of algorithmic trading can be tracked back to the creation of the NASDAQ. This advent of electronic trading platforms set the stage where complex algorithms could be deployed to execute both rapid and high-volume trades.

Benefits over Traditional Methods

- **Increased Efficiency:** Algorithms can process significantly more data than any one person and can execute trades within fractions of a second, greatly increasing the efficiency of trading operations.
- **Reduced Costs:** By automating the trading process, algorithmic trading reduces the need for human traders, thus lowering the costs associated with trading.
- **Minimised Market Impact:** Algorithmic trading can execute large orders in smaller, discrete batches to minimise market impact and avoid significant price changes.
- **Enhanced Accuracy:** Reducing the human element in trading decreases the potential for errors that can occur in manual entries.

1.1.3 Role of Reinforcement Learning

Introduction to Reinforcement Learning

Reinforcement Learning (RL) is a branch of machine learning that looks at how an intelligent agent ought to take actions in a dynamic environment in order to maximise a cumulative reward. Unlike supervised learning methods that require labelled input-output pairs to learn, RL models learn optimal actions via a trial-and-error method through interacting with an environment. In this process, the agent explores different actions to see how they affect the environment, receiving rewards based on the quality of each action taken in a given state. These rewards serve as feedback, indicating how beneficial certain actions are in achieving the desired outcome. The RL model then adjusts its policy, which is the strategy used to choose actions, to maximise the accumulated reward. This involves a careful balance between exploration—trying out new actions to discover their potential benefits—and exploitation—using known actions that have yielded high rewards in the past. Achieving this balance is crucial for the model to effectively learn and improve its performance over time. The fundamental iterative process behind RL is shown in Figure 1.2.

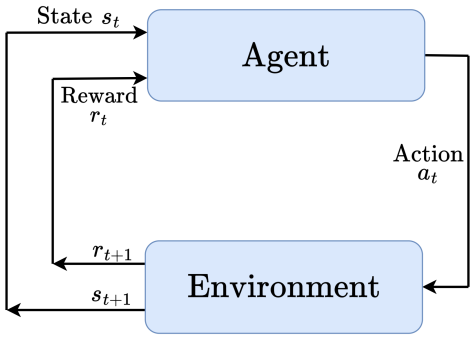


FIGURE 1.2: Fundamental Flow of Reinforcement Learning

Application in Financial Portfolio Management

Due to their ever-changing and unpredictable nature, financial markets are well-suited for reinforcement learning techniques. In portfolio management, RL can be used to continuously learn and adjust investment strategies based on real-time market conditions, allowing for more responsive and autonomous decision-making without needing human input. For example, in portfolio management, RL models may be able to determine the best time for buying or selling assets or holding them so that returns are maximised while risks minimised subject to a reward structure defined by the portfolio manager.

- **Risk Management:** RL models can learn to optimise the trade-off between risk and reward by analysing volatility and trends in market data [4]. This capability is particularly valuable when

managing diverse portfolios that contain a wide variety of assets. Portfolio managers can leverage this to strike an effective balance between positions in large-cap, stable assets such as Apple and Amazon which, generally speaking, return steady positive returns, and slightly more volatile assets which, while having more risk associated with them, can potentially yield greater rewards.

- **Adaptive Learning:** RL models can adjust their strategies based off incoming data, allowing them to effectively respond to changes in the market. This is called model generalisation, which means that an RL model does not require a predefined model of the market, but rather learns what the best investment strategies are for different market conditions. Furthermore, RL models are capable of making real-time decisions, giving them an ability to constantly learn and update their investment strategies.

Potential Challenges in Implementation

Despite the clear advantages of this method in the area of portfolio management, there are some large challenges that would need to be overcome for a successful model to be developed.

- **Computation:** RL is very computationally expensive. Extensive time and resources are required to train a model in a complex environment.
- **Reward alignment:** Defining an appropriate reward structure is critical. Without the right incentives, RL models might learn behaviours that do not align with the intended goals, fail to improve, or get stuck repeating the same actions, which prevents them from achieving the desired results.
- **Complexity of data:** The unpredictable nature of market environments can lead to instability in learned behaviors, making it hard for models to generalise across different market conditions. This complexity can result in a very unstable learning process.

1.2 Problem Statement and Objectives

1.2.1 Definition of Problem

In the world of financial portfolio management is the dynamic and unpredictable nature of markets. The price of a stock is essentially a tug-of-war between buyers and sellers: if more people want to buy a stock than sell it, the price goes up and more people want to sell it, the price goes down. Unfortunately, the number of buyers and sellers can be influenced by a limitless number of factors, many of which are impossible to predict. A very powerful recent example of this being the COVID-19 pandemic in 2020 which caused the S&P 500 to fall by 34% in the space of a month.

This problem was first formalised by Louis Bachelier in 1900 [6]. Bachelier realised that it was virtually impossible to predict all the factors that effect a stock's price accurately and the best you can do is to assume that at any point in time, the value of a stock is just as likely to go down as it is to go up and therefore over a long period of time, stock prices follow a "random walk". In theory, this should mean that it is near impossible to consistently make money by investing in stocks. This is known as the Efficient Market Hypothesis. Bachelier concluded that the expected future price of a stock at any given moment in time can be modelled by a normal distribution centred around the current price, with the distribution's variance increasing as the prediction extends further into the future.

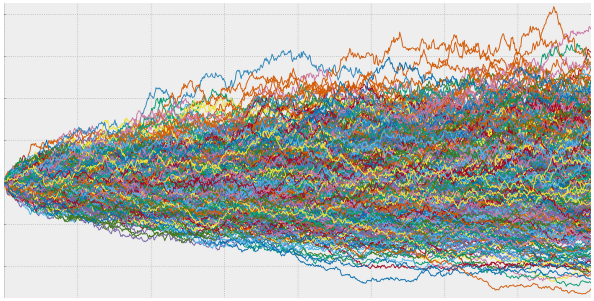


FIGURE 1.3: Monte Carlo Simulation of Asset Price Paths Illustrating the Increasing Variance Over Time [5].

This phenomenon can be shown via a Monte Carlo simulation, which generates multiple simulated price paths for an asset over a fixed period by applying Bachelier's principles . This idea is shown in Figure 1.3. The spread of these paths widening over time, indicates that the potential range of asset prices becomes more varied as the time progresses.

The above behaviour follows the assumptions laid out in the Efficient Market Hypothesis, where the prices reflect all available information. In reality however markets can exhibit various inefficiencies such as momentum, delayed information dissemination, and other irrational behavioural biases. These imperfections create opportunities, as well as risks. This means that an effective portfolio management strategy could be capable of capitalising on these imperfections to create consistent positive returns and manage the subsequent risk appropriately. This study aims to explore the efficacy of RL models in exploiting such inefficiencies and leveraging them to optimise portfolio performance. Furthermore, the chosen methodology can be assessed by its ability to implement strategies such as dynamic asset allocation, diversification of assets, and management of risk levels - strategies that are fundamental modern portfolio management and that have proven to be essential in navigating the complexities of financial environments. By simulating various market conditions through the use of historical financial data, the model's adaptability and strategy selection can also be assessed.

1.2.2 Research Questions

Primary Research Question

Can an RL trading model consistently generate higher returns than a fixed strategy? Section 1.2.1 looks at the high degree of complexity inherent to financial markets, making it challenging to deliver investment strategies consistently deliver significant positive returns. If an RL model can not only match but exceed the returns of a fixed-strategy heuristic, it would validate its effectiveness as a viable approach to portfolio management and optimisation.

Secondary Questions

In addition to the primary question, there are several secondary questions that aim to evaluate the more advanced behaviours that the model is expected to exhibit. These questions will help to assess the model's sophistication and adaptability.

- **Can the model implement effective strategic behaviour in portfolio management?** This question looks at whether an RL model can do more than just pick high-performing stocks; it aims to see if the model can show some strategic thinking. It doesn't matter if these are traditional strategies like diversification and dynamic asset allocation, or new strategies that come from the model's own learning and decision-making process. The key focus here is whether the model can develop a plan that adapts to the current market conditions.
- **How does the RL model respond to extreme market conditions?** Extreme market events, such as financial crises or sudden market crashes, test the resilience and adaptability of any investment strategy. For RL models, these conditions present a unique set of challenges and opportunities that are very difficult to prepare for. This research aims to explore how an RL model reacts under such extreme scenarios and assess the robustness of RL-driven strategies in turbulent market periods.
- **How will different reward structures impact the RL model's performance?** As mentioned in Section 1.1.3, the design of the reward structure is a critical component of creating an RL model. Here it will play a key role in influencing how the model handles the balance between risk and return. For instance, if the reward structure emphasises short-term rewards over long-term rewards, a riskier policy may be adopted to attempt to maximise short-term rewards. By experimenting with diverse reward criteria, this study aims to identify which incentives lead to the most robust portfolio management strategies.

1.2.3 Project Objectives

Table 1.1 outlines the objectives for this research on RL in portfolio management. There are two classes of objectives; the primary objective which is highlighted in blue and the supporting objectives which are highlighted in green. These objectives are designed to guide the direction of research and development of this of this model, ensuring a focused and systematic workflow.

These objectives have been crafted based off the questions laid out in Section 1.2.2 and are designed to assess not only the general ability of the final model but also its ability to handle the more nuanced aspects of portfolio management

Objective	Description	Success criteria
Assessing RL Model Efficacy	To evaluate whether the RL model can consistently outperform a series of heuristic-based strategies	Successful if the RL model achieves higher absolute returns and risk-adjusted returns than all the benchmark strategies
Strategic Behaviour Identification	To investigate whether the RL model can autonomously discover and apply both conventional and innovative strategies in portfolio management.	Successful if the model identifies and implements a range of strategic behaviours, including established techniques and/or novel strategies, as evidenced by improved performance metrics and adaptability to a range of simulated market scenarios.
Response to Extreme Market Events	To evaluate the model's resilience and adaptability during market crises and high-volatility events.	Successful if the model maintains stability and manages risk effectively. It must avoid significant losses, greater than the baseline strategies.
Impact of Reward Structures	To analyse how different reward structures influence the RL model's strategic decisions and its success in achieving specified financial goals.	Successful if variations in reward parameters lead to discernible changes in the model's investment behaviours. These changes should align with the specific financial goals behind that the given reward structure

TABLE 1.1: Primary and Secondary Objectives of This Research

Deliverables

1. **Market simulation framework:** A comprehensive software architecture that models the behaviour of financial markets will need to be produced. This framework will be designed to track asset values over time, enabling the measurement of portfolio performance across different market conditions. This framework will also need to be able to interface with the chosen RL method.

2. **Reinforcement leaning model:** A model capable of interacting with the previously mentioned architecture to manage a virtual portfolio of assets.

3. **Functionality to evaluate model performance relative to baseline strategies:** Scripts capable of evaluating a model's performance in a given environment are essential. These scripts will determine whether the success criteria have been met, providing a clear and objective assessment of the model's performance.

4. **Analysis of learned strategies:** Beyond analysing the results of a model, techniques need to be derived that can analyse an RL model's underlying asset selection strategies.

5. **Sensitivity analysis of a model's reward function:** A study should be carried out to investigate how a the reward function composition effects the learned behaviours of the model - and whether this aligns with the financial goals that are implied by the reward structure.

1.3 Scope and Significance

1.3.1 Scope of the Study

Limitations of the Research

There are a wide range of investment fields that can be considered for the scope of this project, each with their own distinct characteristics and market dynamics. Options range from equities in major global stock markets such as the LSE, New York Stock Exchange (NYSE), and the Tokyo Stock Exchange (TSE), to bonds, commodity markets which looks at the trading of physical goods such as oil and gold, as well as foreign exchange (Forex) markets which looks at the buying and selling of foreign currencies - a highly liquid and volatile environment. There are even emerging fields such as cryptocurrencies or more traditional areas such as real estate and private equity.

Despite the potential in these various markets, this research will focus on the NASDAQ stock market. This choice is motivated primarily by the abundance and ease-of-access of historical data associated with NASDAQ assets. As it was the first electronic stock market, NASDAQ provides reliable asset data, at all levels of granularity, all the way back to the 1970s. With appropriate data extraction and handling methods, this extensive dataset offers the opportunity to train the model effectively and evaluate its performance during periods of heightened volatility. Furthermore, the ease of access to this data will allow this research to remain firmly centred around the RL aspect of portfolio management, rather than being diverted by the nuances of advanced data collection and manipulation. In addition to the abundance of financial data, the NASDAQ exchange hosts a diverse array of companies, from stable market leaders such as Apple and Microsoft, to rapidly evolving biotechnology firms. This not only offers a broad spectrum of market behaviours but also volatile stock movements. This diversity ensures that RL models are exposed to a wide range of market scenarios. This will hopefully increase the model's adaptability. Considering the above, the NASDAQ should be an exceptionally suitable financial environment for this research. The abundance of high-quality financial market data and complex market dynamics should provide a terrific platform to determine the viability of an RL-based portfolio management tool for NASDAQ assets.

Data Limitations

Unfortunately, this scope imposes quite rigid geographical and sectoral limitations. By focusing on the NASDAQ, the model will be exposed to US-centric market behaviours. Such behaviours may not be entirely, or at all, applicable to markets in Europe, Asia, or emerging economies. This will limit any findings from this research and restrict its application to global contexts. Furthermore, the NASDAQ is centered predominantly around the technology and biotech sectors. This means that the findings may not be applicable to other sectors such as utilities or consumer goods that may have different market trends and risk profiles. This will affect the model's ability to explore certain portfolio strategies such as diversification.

Reinforcement Learning (RL) is, generally speaking, a computationally intensive process, and the market environment that an RL model must navigate is highly complex. As a result, the success of this research may heavily depend on the processing power available. If the computational resources available turn out to be insufficient, it may be necessary to analyse a reduced market sample instead.

Despite these limitations, this research remains relevant. It stands to reason that if an RL model can successfully manage a portfolio using a representative sample of NASDAQ assets, it should also be capable of managing a portfolio with access to all NASDAQ assets or even adapting to different market environments. This would demonstrate the model's robustness and versatility in portfolio management across various contexts.

2. Background

Literature Review Methodology This research has only examined scholarly literature to ensure the authenticity of the information being gathered and is divided into four sections. Section 2.1 provides an overview of portfolio theory, examining the fundamental aspects of portfolio management as well as the evolution of methods used to calculate the expected return of a portfolio. Section 2.2 evaluates portfolio performance metrics and how 'risk' can be quantified in different ways when calculating risk-adjusted returns. Following on from this, Section 2.3 provides a formal mathematical foundation for Reinforcement Learning (RL) and evaluates four prominent techniques, each representing one of the major RL paradigms, in the context of their applicability to this research. Finally, Section 2.4 explores some past examples where RL has been applied in the field of portfolio management, highlighting their chosen methodologies and outcomes.

2.1 Overview of Portfolio Management

2.1.1 Modern Portfolio Theory

One of the most fundamental questions in the field of portfolio management is - How do you allocate wealth among alternative assets? This is a problem faced by all financial institutions and the study of how such allocations should be decided is known as portfolio theory. Harry Markowitz is the father of Modern Portfolio Theory (MPT). Publishing his discoveries for the first time in 1952 in the Journal of Finance, he made numerous insights and suggestions that anticipated many of the subsequent developments in the field [7]. Markowitz approached the problem of building a portfolio by focusing on the balance between the average return (mean) and the risk (variance) of the investments. He proved a key idea: if you want to keep risk constant, you should aim to maximise the expected return, and if you want to keep the expected return constant, you should aim to minimise the risk. These ideas led to the concept of an "efficient frontier", which is a range of optimal portfolios that investors can choose from based on how much risk they're willing to take for a certain level of return. The most important theory that this work delivered was that assets could not be selected purely on characteristics that were unique to the security. Rather, an investor also had to consider how each security co-moved with all other securities. Taking these co-movements into account resulted in an ability to create a portfolio that had the same expected return but less risk than a portfolio that ignored interactions between securities [8]. Formally, Markowitz defined MPT as follows:

Maximise the expected return of the portfolio:

$$\mathbb{E}(R_p) = \sum_{i=1}^{n} \omega_i \mathbb{E}(R_i) \tag{2.1}$$

Minimise the variance of the expected returns (risk)

$$V = \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i \omega_j \sigma_{ij} \tag{2.2}$$

Where:

$$\begin{aligned} R_i &= \text{Expected Return of asset } i \\ R_p &= \text{Expected Return of the portfolio} \\ \omega_i &= \text{Portfolio weight of asset } i \\ \sigma_{ij} &= \text{Covariance between assets } i \text{ and } j \\ \sum_{i=1}^{n} \omega_i &= 1 \end{aligned}$$

Modern Portfolio Theory (MPT), often referred to as Mean-Variance Theory due to equations above, is highly regarded because it provides a systematic way to balance risk and return, helping

investors create diversified portfolios that can optimise returns for a given level of risk. However, it is very theoretical framework that is limited by a few key assumptions. Firstly, it has no regard for user sentiment. Secondly, it assumes that all investors have the same expectations about market performance and that these expectations will remain constant [9]. Thirdly, risk is measured purely by the volatility of an asset's price, ignoring any other factors that may influence investment decisions. Finally, this model relies on the assumption that asset returns follow a normal distribution. As discussed in Section 1.2.1, this is a property of an efficient market, which markets today are not[8]. In addition to these assumptions, it provides no mechanism to evaluate the expected return of individual relative to the market (R_i). This represents a critical gap that must be addressed to fully understand asset-specific risks and returns.

2.1.2 Capital Asset Pricing Model

The Capital Asset Pricing Model (CAPM) was created by Sharpe in 1964 [10], with parallel contributions from Lintner 1965 [11] and later refinements by Black in 1972 [12]. For this reason, it is also referred to as the SLB model. It builds on the framework created by Markowitz and defines a simple linear relationship between the mean return on an individual security and the systematic risk of this security (Beta/β). This systematic risk is the only risk factor considered by CAPM. The remarkable simplicity of and ease-of-use of this model made this theory widely popular among both academics and practitioners alike. Beta represents how much the asset's returns move in relation to the overall market. Sharpe and Lintner prove that under a set of assumptions, the following relationship holds:

$$\mathbb{E}(R_i) = R_f + \beta(\mathbb{E}(R_m) - R_f) \tag{2.3}$$

Where:

$$\begin{aligned}
\mathbb{E}(R_i) &= \text{Expected Return of asset } i \\
\mathbb{E}(R_m) &= \text{Expected Return of the market} \\
R_f &= \text{Risk-free rate} \\
\beta &= \frac{\text{Cov}(R_i, R_m)}{\sigma_m^2} = \text{Systematic risk} \\
(R_m - R_f) &= \text{Market risk premium}
\end{aligned}$$

The risk free-rate (R_f) serves as the baseline for what investors should expect to earn by taking no risk. This is usually represented by the yield of government bonds like U.S. Treasury bills. The market risk premium ($R_m - R_f$) represents the additional return over the risk-free rate that investors demand for choosing to invest in the market rather than risk-free securities.

CAPM was one of the first formalised financial models to estimate the expected return of an asset in relation to its risk relative to the overall market as well as the relationship between expected return and risk, establishing that higher risk (higher β) should be compensated with higher expected returns. Generally, the market is assigned a β of 1.0. An asset with a beta greater than 1.0 is considered more volatile than the market. It tends to experience larger price swings compared to the market and vice versa. An asset with a negative beta means it moves in the opposite direction of the market. This is rare but can occur with certain types of investments like gold [13].

This model has been influential in how both academics and financial professionals think about the relationship between risk and returns. The core idea of this model is that the market portfolio, which includes all invested wealth, is efficient in terms of risk and return. This efficiency not only means that the expected return on any asset is directly and positively related beta, which in turn measures how an asset's return moves with the whole market but also that beta is the sole measure of risk for an asset. However over the following decades several real world observations would emerge that would challenge this model [14, 15].

The first such challenge was in 1980 when Stattman found that stocks with a higher book to market ratios (BMRs)[1] tend to have higher average returns [16]. This relationship was also found in Japanese stocks by Chan, Hamao, and Lakonishok in 1991 [17]. Then in 1981, Banz discovered that the size of a company, measured by its market equity (the stock price times the number of shares), also affects returns [18]. Specifically, smaller companies (with lower market equity) tend to have higher average returns than the model would predict based on their beta, and larger companies tend to have lower returns. Finally, in 1988, Bhandari showed that companies with higher leverage (more debt relative to equity) also have higher average returns [19]. While it makes sense that more leverage could mean more risk and higher returns, the CAPM model suggests that this risk should be reflected in the company's beta. However, Bhandari found that leverage provides additional explanatory power beyond beta. These limitations were addressed in 1992 by Eugene Fama and Kenneth French, when they decided to gap the limitations posed by CAPM model.

2.1.3 Fama-French Three-Factor Model

Fama and French's groundbreaking study in 1992, "The Cross-Section of Expected Stock Returns," challenged the then-prevailing CAPM by demonstrating that it failed to explain the variation in stock returns adequately [14]. The Fama-French model proposed that along with market risk, two additional factors — size and value — are significant determinants of stock returns.

Risk factors incorporated into the Fama-French Model:

- **Market Risk:** Like CAPM, the Fama-French uses systematic market risk as a primary factor of risk. This factor acknowledges the influence of overall market movements on individual asset returns.
- **Size Factor (SMB – Small Minus Big):** The size factor captures the excess returns of small-cap stocks over large-cap stocks. This factor is calculated as the difference in returns between a portfolio of small-cap stocks and a portfolio of large-cap stocks, reflecting the historical tendency of small-cap stocks to outperform their counterparts.
- **Value Factor (HML – High Minus Low):** The value factor measures the difference in returns between stocks with high book-to-market ratios (value stocks) and those with low book-to-market ratios (growth stocks). This factor reflects the risk and return dynamics associated with the investment style of buying undervalued (high BMR) stocks versus growth-oriented stocks (low BMR)

Using this more compelete representation of risk, Fama and French derived the following model:

$$\mathbb{E}(R_i) = R_f + \beta_{iM}(R_m - R_f) + \beta_{iSMB} \cdot SMB + \beta_{iHML} \cdot HML + \epsilon_i \tag{2.4}$$

Where:

SMB = Portfolio return of small stocks minus the portfolio return of large stocks. This represents the size premium.

β_{iSMB} = Sensitivity of the stock's returns relative to the SMB factor

HML = Portfolio return of stocks with high BMRs minus the portfolio return of stocks with low BMRs.

β_{iHML} =Sensitivity of the stock's returns relative to the high minus low (HML) factor.

ϵ_i =Error term for stock i representing residual returns not explained by the model.

In 2017 Sattar did an investigation comparing the CAPM and Fama-French models in explaining the returns of the cement industry in Bangladesh over ten years. The findings support the conclusion that the Fama-French has more power to explain stock returns than the CAPM, as beta alone (used in CAPM) doesn't capture much of the variation in returns. However, he also states that the Fama-French model is significantly more complex both in terms of the calculations that need to be performed and the data that is required, and that collecting this might not be cost-effective for practitioners. In the context of using RL to calculate expected returns for assets, the increase in complexity between CAPM and Fama-French may lead to significant increase in

[1]Book to market ratio is given by the an asset's Book Value (the value of a company's assets as recorded on its balance sheet, minus its liabilities.) divided by its Market Value (the total market capitalisation of the company, which is calculated by multiplying the current stock price by the total number of outstanding shares)

computational demand. Furthermore, Sattar goes onto conclude that in the Dhaka Stock Exchange, most individual investors prefer simpler methods due to a lack of deep financial knowledge and warns that institutional investors who consider switching from CAPM to the Fama-French model should carefully evaluate the additional time and effort required before making the change [20].

2.1.4 Carhart Four-Factor Model

While the Fama-French model advanced investor's abilities to calculate expected returns, there turned out to be a critical factor missing: the impact of human behaviour on market dynamics. In 1985, prominent behavioural economists De Bondt and Thaler used the common belief that people tend to overreact when they get new information to infer that stock prices might also overreact [21]. For example, if a company reports bad news, investors might react by selling off the stock too aggressively, causing the price to drop more than it perhaps should. This overreaction can result in the stock becoming undervalued creating an opportunity for above-average returns. This led De Bondt and Thaler to conclude that contrarian strategies (buying past losers and selling past winners) achieve abnormal returns. While the results of this contrarian strategy are still being debated[2], it did reveal a key inefficiency in the market: new information can trigger an overreaction in investors which in turn can have serious implications for the overall market efficiency by creating a momentum in price change. This momentum was first exploited by Jegadeesh and Titman in 1993, where they show that strategies which involve buying stocks that have performed well in the past and selling those that have performed poorly can lead to significant positive returns over short-term holding periods of 3 to 12 months. This study also concluded that the profitable returns were not due to the systematic risk of the stocks or delayed reactions to common market factors. It was also observed that a portion of the abnormal returns earned in the first year tend to fade over the next two years [22].

These studies empirically prove that there is an additional risk factor that can significantly contribute to the expected returns of an asset – momentum. This was formalised in 1997, when Carhart created the Carhart Four-Factor Model [23]. This model is a direct extension of the Fama-French Three-Factor Model by simply adding a momentum factor to the calculation.

Additional Risk Factor:

- **Momentum Factor (MOM - Up Minus Down):** This measures the excess returns of assets that have performed well in the past (winners) compared to those that have performed poorly (losers)

Following on from (2.4), Carhart created the following relationship:

$$\mathbb{E}(R_i) = R_f + \beta_{iM}(R_m - R_f) + \beta_{iSMB} \cdot SMB + \beta_{iHML} \cdot HML + \beta_{iMOM} \cdot MOM + \epsilon_i \tag{2.5}$$

Where:

$$MOM = \text{Momentum factor, capturing the excess returns of winning assets compared to losing assets}$$
$$\beta_{iMOM} = \text{Sensitivity of the stock's returns relative to the Momentum (MOM) factor}$$

Despite appearing to be the most complete model, it is the model that has the most critique surrounding it. It has been heavily scrutinised in the past for its inconsistencies across different market conditions and climates. A major critique is that momentum-based strategies are prone to significant losses during market reversals, as identified by Daniel and Moskowitz in 2016 [24]. They argue that momentum strategies, while ultimately profitable, can suffer sudden and severe downturns during periods of market stress, which raises serious concerns about the robustness and reliability of using momentum as a factor in asset pricing models. Furthermore, Cooper, Gutierrez, and Hameed highlight that the effectiveness of the momentum factor varies with market conditions, performing poorly during market downturns [25]. These conclusions support the idea that momentum factors, while effective, are variable and not consistent across all economic climates.

[2]Some critics argue that the better performance of the "losers" could be explained by other factors, such as those stocks being riskier or smaller in size (which can also lead to higher returns). Additionally, since the outperformance mainly happens in January, it's unclear whether this pattern is due to overreaction or just a seasonal effect.

In addition to this, Lesmond, Schill, and Zhou challenge the value of momentum-based strategies all together [26]. They state that as a result of the short-term nature of momentum-based strategies, transaction costs can significantly erode the profits, indicating that the net gains may not be as substantial as theoretical models suggest. These critiques outline the potential risks associated with utilising the momentum factor within the Carhart model and suggest that investors should exercise caution.

2.2 Portfolio Evaluation

As introduced in Section 1.1.3, RL models rely on a well-defined reward function to indicate to the model how useful the previous action was. In this context, that reward will be some form of portfolio evaluation metric. While it would be very easy to make this reward function a basic return on investment (ROI), this would make it very difficult for the model to determine whether a change in the reward was as a result of a good action, or an unpredictable change in asset prices. This confusion may lead to the model developing a policy that is neither robust nor adaptable. With this in mind, it's clear that a reward function should focus not just on achieving a good outcome, but on developing a sound, consistent policy. This reward function will need to build on the axioms of MPT laid out in Section 2.1.1 by aiming to maximise the expected returns of the portfolio while minimising the implied risk.

2.2.1 Sharpe Ratio

The Sharpe Ratio was introduced by William Sharpe in 1966, it is a one of the most fundamental financial metrics used to assess the performance of an investment compared to a risk-free asset, after adjusting for its risk [27].

$$\text{Sharpe Ratio (ShR)} = \frac{R_p - R_f}{\sigma_p} \tag{2.6}$$

Where:

$$\begin{aligned} R_p &= \text{Portfolio return} \\ R_f &= \text{Risk-free rate} \\ \sigma_p &= \text{Standard deviation of portfolio returns (portfolio risk)} \end{aligned}$$

However, like MPT, this metric assumes that returns are normally distributed. It also cannot differentiate between upward and downward volatility, which can lead to the misrepresentation of the risk in portfolios [28].

2.2.2 Treynor Ratio

The Treynor Ratio was proposed by Jack Treynor in 1965 and it also measures the returns earned in excess of the risk-free rate per unit of market risk but uses the beta of the portfolio as the denominator instead of volatility. This ratio provides insight into how well the portfolio is performing relative to systematic risk [29].

$$\text{Treynor Ratio (TR)} = \frac{R_p - R_f}{\beta_p} \tag{2.7}$$

Where:

$$\beta_p = \text{Portfolio beta relative to the market}$$

While the Treynor Ratio is valuable for evaluating investments that are well diversified, it is less applicable to portfolios that are not diversified as it does not account for unsystematic risk[3]. Additionally, its reliance on beta assumes a linear relationship between portfolio returns and market movements which, as seen in Section 2.1.2, is not always true [30].

[3]Unsystematic risk, also known as idiosyncratic or specific risk, is the type of risk that is specific to a particular company or industry, unlike systematic risk which is risk that affects the entire market

2.2.3 Sortino Ratio

Finally, the Sortino Ratio was developed by Frank Sortino in the late 1980s and early 1990s. This ratio modifies the Sharpe Ratio by focusing only on downside risk, therefore distinguishing harmful volatility from total overall volatility. This metric divides the excess return of the portfolio over the risk-free rate by the standard deviation of negative asset returns, making it more relevant for investors who are primarily concerned with downside risk [31].

$$\text{Sortino Ratio (SoR)} = \frac{R_p - R_f}{\sigma_d} \tag{2.8}$$

Where:

$$\sigma_d = \text{Standard deviation of negative asset returns (downside risk)}$$

However, when calculating σ_d, the investor needs to set a minimum acceptable return. The Sortino Ratio measures how much the actual returns fall below this target and only considers this downside deviation in its calculation. This additional requirement adds complexity and variance to the model as it has a significant effect on the output. If the target is set too high, almost all returns might be considered downside risk, making the Sortino Ratio overly sensitive to poor performance relative to this unrealistic benchmark [31].

2.2.4 Portfolio Entropy

Portfolio entropy is a measure of diversification and it was originally introduced as a part of information theory. In this original application, entropy is used to quantify the unpredictability or randomness of information in a system. This idea was introduced by Claude Shannon in 1948 [32]. This metric was adapted for the world of finance measure portfolio diversification specifically by measuring the distribution of weights among the assets in a portfolio. High entropy indicates a more evenly spread allocation across different assets, suggesting a well-diversified portfolio, while low entropy suggests concentration in fewer assets.

One may ask, why is an specific measurement of portfolio diversification required? The Treynor, Sharpe, and Sortino ratios are valuable for evaluating risk-adjusted returns, but they do not account for asset-specific (idiosyncratic) risk, only the overall risk associated with a portfolio's returns. Consequently, a portfolio could achieve high scores on these metrics by concentrating investments in a few high-performing assets, ignoring the potential risks associated with individual asset performance. This concentration increases vulnerability to idiosyncratic risk, which is mitigated by investment diversification, potentially leading to significant losses if those specific assets perform poorly. The formula for portfolio entropy H is given by:

$$H_t = -\sum_{i=1}^{n} \omega_{i,t} \ln(\omega_{i,t}) \tag{2.9}$$

2.2.5 Summary of Evaluation Metrics

Each of the Sharpe, Treynor, and Sortino ratios provide a method to calculate either the expected or actual risk-adjusted return, making them applicable for both designing reward functions in modeling environments and validating performance in practical investment scenarios. However, these ratios primarily focus on systematic risk or overall volatility, and do not explicitly account for asset-specific (idiosyncratic) risk, which can lead to concentrated portfolios that are vulnerable to specific asset performance. To address this limitation, portfolio entropy can be used as an additional metric, providing insight into the diversification of a portfolio by measuring how evenly investments are distributed across different assets. Given the variations in risk perspective and the potential limitations of each approach, including the risk of concentration, determining the most effective metric or combination of metrics for a reward function within a financial model can only be found through practical experimentation. This hands-on approach will ensure that the choice of reward function is comprehensive and informed, promoting both high returns and effective risk management.

2.3 Reinforcement Learning Techniques

2.3.1 Formalisation of the Reinforcement Learning Paradigm

Reinforcement Learning is a broad term that encompasses a variety of techniques designed to solve problems where an agent learns to make decisions by interacting with an environment. There are four major components to a RL architecture [33].

Agent and Environment: The goal of a reinforcement learning (RL) algorithm is to train an agent to interact with the environment (or system) in a way that maximises a given reward function over time.

Action: $\omega_{t+1} \in \mathbb{A}$ is the signal that the agent sends to the system at time t. It is the only the way the agent can influence the environment state, and subsequently generate new reward signals. The action space $\mathbb{A}$ refers to the set of actions that the agent is allowed to take in a given state. It can either be:

- Discrete: $\mathbb{A} = \{a_1, a_2, ...a_M\}$
- Continuous: $\mathbb{A} \subseteq [x, y]^M$

Reward: $r_t \subseteq \mathbb{R}$ is a feedback signal indicates how well the agent is performing at time step t. The agent's goal is to maximise the cumulative reward signal over a sequence of time steps. Reinforcement learning tackles sequential decision-making tasks by training agents to optimise for delayed rewards.

State and Observation: The overall state, $s_t \in \mathbb{S}$ refers collectively to the both the environment and agent state. The environment state, s_t^e, is a representation of the system in order to determine the next observation, o_{t+1}, and the reward, r_{t+1}. The environment state is usually invisible to the agent, as if it is not, it can contain irrelevant information that would only add noise [34]. The history, $\boldsymbol{h_t}$, at time t is the sequence of observations, actions and rewards up to time step t, such that:

$$\boldsymbol{h}_t = (o_1, a_1, r_1, o_2, a_2, r_2, \cdots, o_t, a_t, r_t) \tag{2.10}$$

The agent state, s_t^a, is the internal representation of the environment that is presented to the agent, used to select the next action, a_{t+1}. This state can be any representation of the history.

$$s_t^a = f(\boldsymbol{h}_t) \tag{2.11}$$

The term 'state space', $\mathbb{S}$, is used to refer to the states that agents can observe or construct. Just like the action space it can be:

- Discrete: $\mathbb{S} = \{s_1, s_2, ...s_N\}$
- Continuous: $\mathbb{S} \subseteq \mathbb{R}^N$

Environments can either be fully observable or partially observable. In a fully observable environment, agents can directly observe the environment state. Therefore:

$$o_t = s_t^e = s_t^a \tag{2.12}$$

In partially observable environments, agents have indirect access to the the environment state, therefore the agent has to construct its own state representation, s_t^a.

RL models are predominantly based on the framework of Markov Decision Processes (MDPs), which is a mathematical approach used to model environments where decisions are made sequentially under conditions of uncertainty [35]. An MDP is defined by a set of states, a set of actions and a transition function, $P(s_{t+1}|s_t, a_t)$, which determines the probability of transitioning from state s_t to s_{t+1} after taking action a_t, and a reward function $R(s_{t+1}, s_t, a_t)$ that assigns rewards for transitions between states due to actions. The objective in an MDP is to find a policy, $\pi(a|s)$,

that maximises the expected cumulative discounted reward, where the discount factor, γ, weighs the importance of future rewards against immediate rewards.

$$\textbf{Expected Reward} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})\right] \tag{2.13}$$

There are four families of RL techniques [36] :

- **Policy-based Methods:** These methods directly optimise a policy that maps states to actions, which offers advantages in environments with high-dimensional or continuous action spaces.
- **Value-based Methods:** Value-based methods focus on deriving a value function that estimates the future rewards of actions, providing a form of decision making where the policy isn't explicitly defined.
- **Actor-critic Methods:** These models combine the strengths of the value-based and policy-based approaches, using two models to balance the learning of the policy and value estimation.
- **Model-based Methods:** These take a different approach altogether by developing an internal model of the environment, which the agent uses to simulate and plan future actions

In this review, the most prominent methodology from each family will be evalated.

2.3.2 Policy-based: Proximal Policy Optimisation (PPO)

Proximal Policy Optimisation (PPO) is a very powerful deep RL tool that was designed by researchers at OpenAI in 2017. It is the latest generation policy-based RL method, surpassing Trust Region Policy Optimisation (TRPO). Within PPO, there are two deep learning models.

Policy Neural Network This network is responsible for directly mapping states to actions. It outputs a probability distribution for action values for continuous action spaces, representing the policy, $\pi(a|s;\theta)$, where θ are the parameters of the policy neural network. The objective of training the policy network is to maximise the expected return from the environment, calculated over the probability of taking actions in given states under the current policy.

Value Neural Network This network estimates the value function, $V(s;\phi)$, where ϕ are the parameters of the value neural network. This network predicts the expected return from a given state to provide a baseline against which the policy's performance can be judged and the policy neural network be adjusted.

PPO optimises both the policy and value networks simultaneously through an actor-critic framework[4]. The policy network, serving as the actor, generates a policy that maps states to a probability distribution over available actions. Based on this distribution, an action is selected, propelling the model into a new state. The subsequent state's outcome, combined with the received reward, is evaluated by the value network, known as the critic.

The critic estimates the expected value of the new state and compares it to the actual outcome to determine the advantage, often denoted as A_t. This advantage indicates whether the action taken was better or worse than the policy's average expected outcome. The loss function for the policy network is then calculated by multiplying this advantage by the policy ratio, $(r_t(\theta))$.

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \tag{2.14}$$

$$L_{\text{policy}}(\theta) = \mathbb{E}_t(r_t(\theta)A_t) \tag{2.15}$$

[4]Despite using an actor-critic framework, PPO is classified primarily as a policy gradient method because its central mechanism and primary objective revolve around optimising the policy directly.

The policy ratio measures how much more or less likely the current policy π_θ values the action compared to the old policy $\pi_{\theta_{\text{old}}}$. It is recalculated whenever the policy parameters θ are updated. This ratio acts as a regulatory mechanism for policy updates: if the policy ratio is high, indicating a significant deviation from the old policy, the loss function may scale up, leading to more substantial changes. This momentum factor creates the opportunity for significant instability in the case that the policy ratio is very large. For this reason, PPO has a built-in clipping mechanism (denoted by the clipping factor ϵ) that ensures that the loss function remains within a certain range. This ensures stability during the learning process. The final surrogate loss function for the policy network is shown in (2.16) and a basic visualisations of the deep learning networks can be seen in Figure 2.1.

$$L^{\text{CLIP}}_{\text{policy}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta) A_t, \text{clip} \left(r_t(\theta), 1 - \epsilon, 1 + \epsilon \right) A_t \right) \right] \tag{2.16}$$

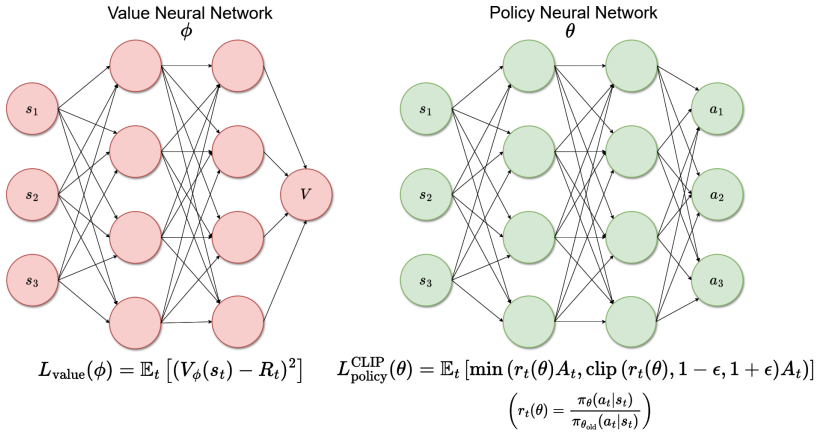


FIGURE 2.1: Visualisation of the Deep Learning Models within PPO

PPO is recognised for its effectiveness and efficiency in reinforcement learning tasks across various domains. The benefits of PPO include its stability in policy updates, achieved through a clipping mechanism in its objective function, which prevents drastic changes in policy behaviour and ensures smooth convergence [37]. This makes it particularly robust in environments with high-dimensional or continuous action spaces. However the disadvantages of PPO include its sensitivity to the choice of hyperparameters, such as the clipping threshold, which requires careful tuning to balance exploration and exploitation adequately [38]. Moreover, while PPO generally performs well in a broad range of tasks, it may underperform in highly stochastic or unpredictable environments where the advantage estimation becomes less reliable due to noise or sparse reward distributions.

2.3.3 Value-based: Deep Q-networks (DQN)

Deep Q-Networks (DQNs) represent a significant advancement in the field of RL. By combining the classic Q-learning algorithm with the power of deep neural networks to address complex problems with high-dimensional state spaces.

Q-learning DQNs are based on Q-learning, which is a is a model-free, off-policy algorithm introduced by Watkins in 1989 [39]. Q-learning aims to learn the optimal action-value function ($Q(s, a)$), which measures the expected reward that can be obtained by taking a particular action in a particular state and following the optimal policy thereafter. This traditional method updates the Q-values using the Bellman Equation:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right] \tag{2.17}$$

Where:

s and a are the current state and action.

r is the reward received for this action.

γ is the discount factor.

s' is the subsequent state.

α is the learning rate.

This approach provides a recursive decomposition of the action values. Once the graph has been thoroughly explored, the action value functions are stored in a look-up table and the model's policy is to take the maximum action-value path. While highly effective, Q-learning requires a discrete and manageable action space, becoming impractical at in higher dimensional spaces.

DQN Extension In 2015, Q-learning was extended in a ground breaking paper by Minh *et al.* [40], where they demonstrated that a neural network could effectively learn to represent the Q-values for actions. The DQN uses a convolutional neural network (CNN) to approximate the Q-function:

$$Q(s, a; \theta) \approx Q^*(s, a)$$

where θ represents the weights of the neural network, and $Q^*(s, a)$ represents the optimal action-value function. This approach allows DQNs to handle complex inputs that would be infeasible to manage with traditional tabular Q-learning. To demonstrate the effectiveness of DQNs, DeepMind used DQNs to achieve super-human performance on Atari 2600 video games using the raw pixel data from the game as the model inputs [40]. However, as DQNs share the same fundamental architecture as Q-learning it can still only handle discrete action spaces. Unfortunately, this limitation likely makes a value-based method impractical for this application as it is not feasible to rebalance a portfolio using a discrete action space.

2.3.4 Actor-critic: Deep Deterministic Policy Gradient (DDPG)

There were 2 methods considered to be actor-critic candidate for this project. The first was Asynchronous Advantage Actor-Critic (A2C) and the second was Deep Deterministic Policy Gradient (DDPG). Initial research indicated that DDPG is better suited for environments with continuous state and action spaces [41]. This characteristic aligns more closely with the environment requirements, making DDPG the more appropriate choice for this research.

DDPG is a RL algorithm designed for environments with continuous action spaces. Developed by Timothy Lillicrap *et al* and published in their 2015 paper, DDPG integrates concepts from DPG (Deterministic Policy Gradient) and DQN (Deep Q-Network) [42]. This algorithm addresses the gap within the DQN method in efficiently handling continuous action spaces, which were problematic for earlier methods that typically dealt with discrete actions. DDPG has been widely recognised for its efficacy in various applications, ranging from robotics to game playing. DDPG is an actor-critic method with four main components: the actor, critic and target networks, and the experience replay buffer. DDPG is an actor-critic method with four main components: the actor, critic and target networks, and the experience replay buffer.

- **Actor (Policy) Network:** This network directly maps states to actions. It determines the optimal action given a state.
- **Critic (Value) Network:** This network estimates the Q-value of the current state-action pair. Recall Q-value is a measure of the "quality" or value of an action taken in a given state.
- **Target Networks:** DDPG uses two target networks, one for the actor and one for the critic. These networks are copies of the actor and critic networks, but their weights are updated more slowly than the main networks to provide stability to the learning updates.
- **Experience Replay Buffer:** This is where the transitions (state, action, reward, next state) are stored. DDPG samples from this buffer to update the networks, which decorrelates the transitions and leads to more stable learning.

DDPG Training Process

At the start of the DDPG training process, the networks are initialised. The actor network with parameters θ^{μ} ($\mu(s|\theta^{\mu})$), the critic network with parameters θ^{Q} ($Q(s,a|\theta^{Q})$), the target networks – both the target actor (μ') and the target critic (Q'). The target networks have equal weights to their corresponding main networks, i.e., $\theta^{\mu'} \leftarrow \theta^{\mu}$ and $\theta^{Q'} \leftarrow \theta^{Q}$. In addition to this the replay buffer is initialised. To reiterate, this stores transition tuples (s_t, a_t, r_t, s_{t+1}), where s_t is the current state, a_t is the action taken, r_t is the reward received, and s_{t+1} is the next state.

The first step in a training cycle is to interact with the environment and sample an action: $a_t = (\mu(s|\theta^{\mu})$. Due to the deterministic nature of the action selection process, noise is added to the action to ensure continued exploration of the action space. a_t is then executed, yielding a reward, r_t and a new state, s_{t+1}. This information is bundled into a state tuple and stored in the replay buffer. Then a randomised mini-batch of sample transitions are drawn from the replay buffer. To train the critic network, the target networks are used to estimate the total expected return for taking action a_t in state s_t. This is the target value y_t as calculated in (2.18).

$$y_t = r_t + \gamma Q' \left(s_{t+1}, \mu'(s_{t+1}|\theta^{\mu'})|\theta^{Q'} \right) \tag{2.18}$$

This target value is then used to estimate the loss function (2.19) on which the critic network will be trained:

$$L = \frac{1}{N} \sum_i \left(y_i - Q(s_i, a_i|\theta^{Q}) \right)^2 \tag{2.19}$$

The next step is the actor network is updated using a policy gradient method that aims to maximise the expected return. The gradient is calculated by first determining the derivative of the Q-value function, as evaluated by the critic, with respect to the action outputs of the actor. This gradient is then used to adjust the actor's parameters via gradient ascent, enhancing the policy towards more rewarding actions. The update is formalised by the equation:

$$\theta^{\mu} \leftarrow \theta^{\mu} + \alpha \nabla_{\theta^{\mu}} J \tag{2.20}$$

where α is the learning rate, a small positive scalar determining the size of the update step and $\nabla_{\theta^{\mu}} J$ is defined by (2.21).

$$\nabla_{\theta^{\mu}} J \approx \mathbb{E} \left[\nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s=s_t} \cdot \nabla_a Q(s, a|\theta^{Q})|_{s=s_t, a=\mu(s_t)} \right] \tag{2.21}$$

A full derivation can be found in Lillicrap's original paper [42].

DDPG excels in environments where precise, continuous control is essential. This characteristic is particularly advantageous for applications such as algorithmic trading, where the model must finely adjust the size of trades and portfolio allocations in response to dynamic market conditions. The ability of DDPG to generate a broad range of actions allows for nuanced adjustments, crucial for optimising financial returns. In addition to this, DDPG has a high sample efficiency. In financial markets, obtaining and processing extensive data can be prohibitively expensive. DDPG addresses this challenge by employing replay buffers, which enhance the algorithm's ability to learn effective policies from a limited number of samples. This feature is particularly beneficial in scenarios where data is costly or interactions with the environment are restricted [42].

However, similar to PPO, DDPG's performance can significantly vary based on hyperparameter settings. This sensitivity creates the need careful tuning, which can be challenging without extensive trials and domain expertise, especially in complex environments like financial markets [43]. Furthermore, DDPG can suffer from overestimation bias due to its Q-learning foundation. This bias may lead to sub-optimal decision-making, particularly problematic in financial trading where risk assessment is critical [44].

2.3.5 Model-based: Probabilistic Inference for Learning Control (PILCO)

Two methods were considered as candidates for model-based methods: AlphaZero and PILCO. After initial research it was determined that PILCO would be more suitable for in this application. Unlike AlphaZero, which was initially designed for environments with discrete action spaces, PILCO is inherently equipped to handle continuous state and action spaces [45, 46].

PILCO, introduced by Marc Peter Deisenroth and Carl Edward Rasmussen in 2011, is a model-based reinforcement learning algorithm renowned for its efficiency and effectiveness in continuous state and action spaces [46]. It utilises Gaussian Processes (GPs) to model the dynamics of the environment, enabling it to optimally learn control policies with minimal data interaction. This feature makes PILCO exceptionally well-suited for scenarios where data is scarce or collecting data is costly.

PILCO (and indeed most model-based RL methods) have two distinct sections in the training process: the model training and policy development. In the model training section a large sample of data is collected from the environment via random exploration. This data is saved in the form of tuples, consisting of the current state (s_t), the chosen action (a_t) and the subsequent state (s_{t+1}) This is similar to the transition tuples in the experience replay buffer utilised by DDPG as seen in Section 2.3.4 although note how the reward r_t is not stored in this method. This is because all PILCO is trying to do at this point is model the environment dynamics - not evaluate effective policies. The collected data is then used to train a Gaussian Process model. The GP is trained to minimise the prediction error, typically using maximum likelihood estimation or similar methods. The GP model is characterised by a mean function, $m(x)$, and a covariance function, $k(x, x')$ as seen in (2.22).

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{2.22}$$

In the context of PILCO, the GP predicts the next state, s_{t+1}, based on the current state, s_t, and action, a_t. The prediction incorporates uncertainty and is given by the predictive distribution:

$$p(s_{t+1}|s_t, a_t) = \mathcal{N}\left(\mu_{t+1}, \sigma^2_{t+1}\right) \tag{2.23}$$

Once a (reliable) environment model has been established; an effective policy can be derived. The first step is to initialise a policy π that maps states to actions. PILCO then calculates the expected return of a policy over a trajectory of states, using the GP model to integrate over uncertainties in state transitions. The expected return J from a policy π starting in state s_0 is given by:

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r(s_t, \pi(s_t))\right] \tag{2.24}$$

where $r(s_t, a_t)$ is the reward function and γ is the discount factor.

This policy can then be optimised using gradient-based optimisation methods to adjust the policy parameters. The gradients are computed using the chain rule, considering both the direct impact of policy changes on the actions and the subsequent effects on the state transitions and rewards.

One of the key strengths of PILCO is its data efficiency. The algorithm's ability to learn effective policies from a minimal number of data points is particularly advantageous in settings where data acquisition is expensive or challenging. This efficiency is primarily attributed to its use of Gaussian processes, which are adept at capturing the dynamics of complex environments with fewer samples. Deisenroth and Rasmussen (2011) highlight this strength, demonstrating how PILCO achieves significant performance improvements over other methods with much fewer interactions with the environment [46]. Furthermore, PILCO is capable of handling uncertainty through the probabilistic modelling provided by Gaussian processes. This capability allows PILCO to make informed decisions even under significant environmental uncertainty, which is crucial for applications such as financial trading or robotic navigation where conditions can change unpredictably [47].

PILCO does have two decisive weaknesses, the first is computational complexity. As the number of data points grows, the computational time complexity of expanding the model is $O(n^3)$, which can significantly limit scalability, particularly in environments with large datasets such as financial markets [48]. More importantly, the accuracy and reliability of the GP within PILCO can be seriously diminished in highly dynamic and non-stationary environments like financial markets. In such instances, maintaining an accurate and reliable model can be challenging, potentially leading to sub-optimal decisions [49].

2.4 Reinforcement Learning in Portfolio Management

In 2018, Liang *et al.* employed both DDPG and PPO models for portfolio management in the Chinese Stock Market [50]. They used historical processed pricing data as input features for training the models. The effectiveness of these models was assessed by comparing their performance against heuristic-based strategies, such as follow-the-winner and Constant Rebalanced Portfolio (CRP). To facilitate this comparison, they used the Sharpe Ratio and Maximum Drawdown[5] (MDD) metrics. Generally they drew positive conclusions, stating that RL was capable of capturing patterns in market movements, and improve its own performance. However they did say that the results were not as promising as those achieved by reinforcement learning in other domains such as gaming and robotic control.

In 2021, Wu, Syu and Lin, compared two deep reinforcement learning frameworks, each with a different underlying neural network, convolutional (CNN) and recurrent (RNN) [51]. Their novelty was to use the Sharpe ratio as a reward function for the model. They utilise four normalised time series: the open; close; high; and low prices over the last twenty days as observations for each asset. They concluded that the "Sharpe ratio reward function outperformed the conventional return-based reward function, resulting in 39.0% higher profits and 13.7% less drawdown." In addition to this, both methods were validated across four data sets, producing steady returns and outperforming nearly every benchmark.

In 2020 Hieu, applied PPO, DDPG and Generalized Deterministic Policy Gradient (GDPG) to stock-based portfolio optimisation [52]. These policy gradient algorithms were identified for their ability to handle continual actions spaces. He states that a simple reward scheme such as percentage change of portfolio value isn't sufficient as it does not consider asset volatility, or general risk. For this reason, he incorporates Sharpe Ratio into the into the reward function. However, unlike Wu, Syu, and Lin, he combines it with the change in portfolio value, creating the following compound reward function:

$$r(t) = \log(\text{wealth change} - \text{transaction cost}) + (\text{ShR})$$

Like the previous works discussed in this section, this study uses historical prices of each stock in the portfolio over a window of time as the state space. It doesn't mention which pricing data it uses, nor which normalisation procedure (if any) it employs. This set up yielded much less convincing results than the similar study be Liang, with the PPO model even suffering from negative returns. He does conclude however, that from the evaluated models, DDPG yielded the highest absolute returns.

In reviewing existing literature on reinforcement learning applications in portfolio management, it is evident that most examples rely solely on raw financial data for asset observations. This approach not only introduces a high degree of noise, potentially obscuring useful patterns, but also significantly limits the amount of historical data that can be considered in a single observation. Such limitations can affect the model's ability to capture long-term trends and make informed decisions. Additionally, these forms of observations introduce a high degree of complexity due to their size, limiting the number of assets that can be considered. This research proposes a novel approach by integrating aspects of financial analysis such as those in Section 2.1 into asset observations. The aim is to determine whether these more concise features can effectively summarise the pricing data such that an RL model can still effectively capture patterns in market movements, and devise suitable strategies.

[5]Max Drawdown (MDD) is a risk measure used to assess the largest single drop from peak to trough in the value of a portfolio, before a new peak is achieved.

3. Research Methodology

3.1 Modelling Financial Markets as a Discrete-Time System

3.1.1 Model Assumptions

In this research, trades are conducted where the agent 'pretends' to be at a certain point in time in the history of the NASDAQ market. The agent will have no knowledge of future (relative to the agent's current position in time) market information. As a requirement for all experiments, three assumptions must apply, all of which are realistic if the assets being considered for trades have a high enough volume in the market [53].

Zero Market Impact

As discussed in Section 1.2.1, asset prices are determined by supply and demand, therefore in a real world scenario, and trade will impact the the value of an asset (buying increases demand, therefore price and selling increases the supply, therefore decreasing price)

Assumption #1: The capital invested by the trading agent has a negligible effect on the market.

Sufficient Liquidity

A liquid asset is an asset that can easily be converted into cash in a short amount of time with little to no loss in value [54].

Assumption #2: All market assets are liquid, and each transaction can be executed under identical conditions.

Zero Slippage

Slippage is the difference between the expected price of a trade and the actual price at which the trade is carried out. [55].

Assumption #3: The liquidity of all market assets is sufficient to allow each trade to be executed immediately at the most recent price when an order is placed.

3.1.2 Action Space

To address the problem of allocating portfolio funds at each time step, the trading agent should be able to determine the portfolio weighting vector $\boldsymbol{\omega}_t$ at each time step t. Therefore, action $\boldsymbol{a}_t$ at time t is the normalised portfolio weighting vector $\boldsymbol{\omega}_{t+1}$ at time $t+1$ multiplied by some constant k.

$$\boldsymbol{a}_t \equiv k\boldsymbol{\omega}_{t+1} = k\begin{bmatrix}\omega_{t+1,1} & \omega_{t+1,2} & \cdots & \omega_{t+1,M}\end{bmatrix} \tag{3.1}$$

From this action space $\mathbb{A}$, new portfolio weightings for each of the M assets can be derived and translated into buy and sell orders by comparing $\boldsymbol{\omega}_t$ to $\boldsymbol{\omega}_{t+1}$. The action space $\mathbb{A}$ is a subset of the continuous M-dimensional real space $\mathbb{R}^M$ where:

$$\boldsymbol{a}_t \in \mathbb{A} \subseteq \mathbb{R}^M, \qquad \forall t \geq 0 \qquad \text{Subject to:} \qquad \sum_{i=1}^{M} \frac{\omega_{t,i}}{k} = 1 \tag{3.2}$$

3.1.3 Observation Space

At time step t the agent can observe each of the N features of the M assets. This creates a $1 \times N$ observation space for each asset. These N features are covered in more detail in Section 3.5

$$\boldsymbol{o}_t \equiv \begin{bmatrix}f_{t,1} & f_{t,2} & \cdots & f_{t,N}\end{bmatrix} \tag{3.3}$$

Where the observation space $\mathbb{O}$ is s a subset of the continuous real space $\mathbb{R}^{M\times N}$

$$\boldsymbol{o}_t \in \mathbb{O} \subseteq \mathbb{R}^{M\times N}, \qquad \forall t \geq 0 \tag{3.4}$$

3.1.4 Reward Function

The design of the reward function is one of the most critical steps in the design of an RL architecture. As described in Section 2.3.1, the reward is a value that summarises the objective of the agents according to the predefined objective, where the maximisation of the cumulative reward will lead to the optimal solution of the task. Section 2.2 looked at the different metrics used to measure risk-adjusted returns and portfolio diversification, but as concluded in Section 2.2.5 the optimal reward function can only be found through experimentation. With this in mind, a modular reward function framework needs to be designed to enable the comparison of different reward functions. This framework will incorporate the three risk-adjusted return ratios, portfolio entropy, and the Return on Investment (ROI). Such a structure allows for the adjustment of model hyperparameters to identify a near-optimal configuration, optimising the balance between returns, risk management, and diversification. It is important to note that the three risk adjusted ratios will be centered around the **expected returns** of the portfolio as opposed to the **actual returns**. The expected returns are calculated using (2.3). This is to ensure the model's reward function incentivises policies that are consistent with the primary axiom of MPT laid out in Section 2.1.1, which is to maximise the **expected return** of the portfolio. The general reward function equation will be:

$$r_t = c_1 \cdot \mathbb{E}(\mathrm{ShR}_t) + c_2 \cdot \mathbb{E}(\mathrm{TR}_t) + c_3 \cdot \mathbb{E}(\mathrm{SoR}_t) + c_4 \cdot H_t + c_5 \cdot \mathrm{ROI}_t \qquad \text{Subject to: } \sum_{i=1}^{5} c_i = 1 \tag{3.5}$$

Where $\boldsymbol{c}$ is a vector of hyperparameter coefficients that will be varied with the aim of finding the optimal configuration. The minimal acceptable return benchmark for the SoR will be set to zero to measure returns purely against losses, emphasising any negative performance.

3.2 Research Scope

3.2.1 Data Collection

There are over 5,000 assets listed on the NASDAQ. Due to the limited computational resources available, the scope of assets under consideration has been scaled down to 10% of this. The sample of 500 assets were carefully chosen to be representative of the broader market, by selecting a range of high-cap and low-cap stocks as well as stable and volatile stocks. This will ensure that the findings from this study remain relevant and insightful. By focusing on a smaller, yet diversified sample, this project serves as a proof of concept for the proposed reinforcement learning-based portfolio management strategy. The results obtained from this reduced dataset will demonstrate the model's effectiveness in making informed investment decisions. If successful, the methodology can be extended and applied to larger markets with a more extensive range of assets when greater computational power is available. With this in mind, it is important that scalability is a key consideration when designing the model architecture.

3.2.2 Reinforcement Learning Methodology Selection

For this research, **PPO and DDPG will be carried forward to the investigation phase.** PPO has been selected due to the stability it provides in the learning process, making it suitable in high-dimensional environments with continuous action spaces. As outlined in sections 3.1.2 and 3.1.3, the ability to effectively handle such spaces is a crucial requirement for the model in this study. DDPG has been selected not only for its ability to operate in high-dimensional environments and over continuous action spaces but also for its ability to handle precise and continuous control, and create nuanced adjustments. While the learning process may not be as stable as PPO, the deterministic policy approach it employs makes it particularly efficient at sampling and learning, potentially leading to better decision making and therefore overall performance. These conclusions are further justified by the historical use of these methods in Portfolio Management as seen in Section 2.4.

Despite DQN proving to be a robust method in many scenarios, it simply cannot be used in this environment due to it's inability to handle continual action spaces without significant modification. Such a complex environment demands a continual action space, especially when looking at trading volumes. Adapting DQN to continuous spaces requires modifications like discretising the action space or extending it with additional techniques, which can complicate the model and introduce inefficiencies or inaccuracies in action selection. For this reason, DQN will not be carried forwards.

PILCO is a more interesting candidate. On the face it it looks like a strong alternative to the mainstream approaches in PPO and DDPG - having a very high data efficiency, and having an innate ability to handle uncertainty as well as continual action spaces. However, there is a critical weakness which makes this technique infeasible in this application. PILCO is a model-based method, meaning in the initial stages Gaussian Processes are used to create a model of the environment, specifically looking to establish probability distributions that predict the next state, s_{t+1}, based on the current state, s_t, and action, a_t. The issue with this approach can be found in Assumption #1 in Section 3.1.1, which states that any capital invested by the trading agent is so insignificant that is has no influence on the market. Therefore in the context of this project:

$$p(s_{t+1}|s_t, a_t) \equiv p(s_{t+1}|s_t) \tag{3.6}$$

This means PILCO would effectively be trying to predict market movements based purely off the current state of the market[1]. Financial markets are highly dynamic and stochastic, meaning any model it devised would be very inaccurate - leading to poor decision making and low performance.

3.3 Benchmarks

To fulfill the primary objective of determining whether the model can outperform heuristic-based strategies (see Table 1.1), it is essential to compare its performance against a series of fixed heuristic benchmarks. This approach will test the model's adaptability and provide a clear comparison point to measure its effectiveness relative to simpler strategies. In 2012, Li and Hoi did a survey of various portfolio selection methods, some of these methods will be used as benchmarks for this research [56]. The following section explains how weightings are calculated for each of these methods and visualises these calculations by applying them to the asset sample described in Section 3.2.1 over the 20 month period from January 2023 to August 2024. These visualisations can be seen in Figure 3.1.

3.3.1 Uniform Buy and Hold

The Buy and Hold (BAH) strategy is the most common and simplest baseline strategy. In BAH, the investor defines an initial portfolio position in the first time period, from there they hold these positions until the end with no rebalancing. The Uniform Buy and Hold Strategy (UBAH) is a form of the BAH strategy where the investor divides their capital equally among the pool of assets at the initial time step. This strategy reflects the overall performance of the market it is in. This policy will indicate whether the active RL-derived management approach is adding value above and beyond a passive investment strategy.

3.3.2 Uniform Constant Rebalanced Portfolio

The Uniform Constant Rebalanced Portfolio (UCRP) strategy is an extension of UBAH, where instead of initialising a portfolio of equal weightings and taking a passive strategy, the portfolio is rebalanced to maintain equal weightings after each time step. This is a very popular strategy as it does not rely on predictive signals and it is easy to implement. Furthermore, it is an effective benchmark as it assesses the value added to the portfolio beyond a heuristic that entirely relies on overall market performance to generate returns.

$$\omega_{t+1,i} = \frac{1}{M} \qquad \text{For all } i \tag{3.7}$$

[1] Although the current state would include the portfolio composition, the Zero Market Impact assumption defined in Section 3.1.1 dictates that this will have no influence on the market.

3.3.3 Follow-the-Winner Approach

Follow-the-Winner (FtW) approaches revolve around rebalancing portfolio weightings at each interval so that the relative weights of assets that have performed well are better than those that have not. Li and Hoi detail numerous FtW approaches in their survey [56], but the one that will be used in this research will involve setting portfolio weightings at time $t+1$ to the vector of assets' short-term ROI that has been normalised via a softmax function:

$$\omega_{t+1,i} = \frac{e^{\text{ROI}_{t,i}}}{\sum_{j=1}^{M} e^{\text{ROI}_{t,j}}} \tag{3.8}$$

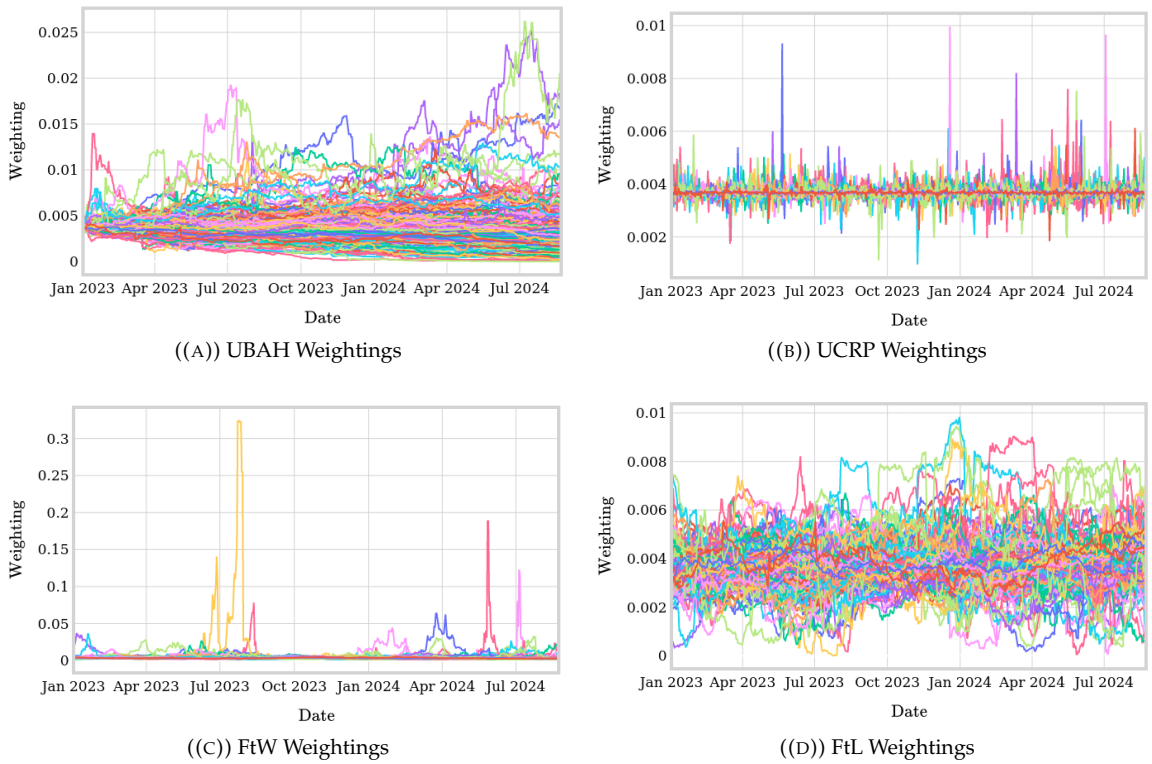
3.3.4 Follow-the-Loser Approach

Follow-the-Loser is the opposite of FtW. This approach is characterised by diverting portfolio wealth towards the assets that are performing poorly. This approach builds on on de Bondt and Thaler's work on mean reversion, as discussed in Section 2.1.4 [21]. The expectation here is that assets that have performed poorly are inherently undervalued due to market overreaction, and will therefore increase in value when the asset price stabilises. This approach will be similar to the FtW approach.

$$\omega_{t+1,i} = \frac{e^{\text{adj-ROI}_{i,t}}}{\sum_{j=1}^{M} e^{\text{adj-ROI}_{j,t}}} \tag{3.9}$$

Where:

$$\text{adj-ROI}_{i,t} = |\text{ROI}_{i,t} - \max_{k=1,\cdots,n} \text{ROI}_{k,t}| \tag{3.10}$$



((A)) UBAH Weightings

((B)) UCRP Weightings

((C)) FtW Weightings

((D)) FtL Weightings

FIGURE 3.1: Weighting progression of the 4 baseline strategies

3.4 Transaction costs

Transaction costs are an important part of modelling a financial trading environment, and portfolio management because they directly impact the net returns of a portfolio. Whenever an asset is bought or sold, transaction costs, such as broker fees, bid-ask spreads, and taxes, are incurred. These costs can erode the gains from an investment, especially in strategies that involve frequent trading. As transaction costs will have a direct impact on the policy developed by an RL agent, it is important they are implemented accurately. Ignoring these costs can lead to overestimating the profitability of a strategy and might result in suboptimal decision-making in real-world applications. It is difficult to pin down a specific value for the average transaction cost, as there are many types of brokerage fees. However a study concluded found that the average transaction fee for an actively managed portfolio in 2023 was 0.42% [57]. This value assumes that the transaction cost of buying γ_b is equivalent to the cost of selling γ_s. This means there is a single universal transaction cost γ_t such that:

$$\gamma_b \equiv \gamma_s \equiv \gamma_t = 0.0042 \tag{3.11}$$

Let's now look at how the transactional fee γ_t is applied to trades. At each time step it can be assumed that γ_t is only being applied to the portion of wealth involved in a trade, δ_ω. For example, if an investor has an investment in an asset worth £10 and they decide to sell 20% of this position (£2), the share of the position that the investor sells will be taxed at the rate of γ_t. This means that the trader's new net worth after this trade (V_{t+1}) is:

$$\begin{aligned} V_t &= £10 \\ V_{t+1} &= £8 + £2 \cdot (1 - \gamma_t) \end{aligned}$$

Defining this more formally for an entire portfolio:

$$\begin{aligned} \delta_\omega &= \sum_{n=1}^{M} |\omega_{t+1,n} - \omega_{t,n}| \\ V_{t+1} &= V_t(1 - \delta_\omega) + V_t\delta_\omega(1 - \gamma_t) \\ V_{t+1} &= V_t\left[(1 - \delta_\omega) + \delta_\omega(1 - \gamma_t)\right] \\ V_{t+1} &= V_t\left(1 - \delta_\omega\gamma_t\right) \end{aligned} \tag{3.12}$$

3.5 Feature Engineering

The novelty of this research lies in its approach to the processing and presentation of input financial data. Previous works (outlined in Section 2.4) use the raw historical pricing data as observations. Such data can potentially contain large amounts of noise which can lead to either overfitting or a failure to identify useful patterns in the data [58]. In an attempt to overcome this, this research will apply financial and data analysis techniques in an attempt to summarise the incoming financial data in a concise manner. Since only a sample of assets is used in this research, with the goal of scaling up to encompass all NASDAQ assets in the future, a primary consideration in choosing analysis techniques is their computational complexity. A design objective is to ensure that the model maintains good scalability, allowing it to handle a larger asset universe effectively as the scope of the analysis expands in future.

3.5.1 CAPM

CAPM was the primary method of calculating an asset's expected returns in Section 2.1 and will be the chosen method for determining an asset's expected return. Despite the Fama-French and Carhart extensions reviewed in Sections 2.1.3 and 2.1.4 respectively, potentially offering a more accurate representation of the asset's true expected return, CAPM offers a much more concise and scalable alternative. The additional factors in the extension models (SMB, HML & MOM) not only require more input data which is difficult to source, but also incurs a significant increase in data handling complexity, which can delay computational processes. Thus, CAPM's simplicity makes it

more efficient for large-scale applications by balancing accuracy with computational efficiency. These additional risk factors will be summarised more concisely by the remaining features.

There are two main components to evaluating an asset's expected return using CAPM, the evaluation of the expected market return $E(R_m)$, and the evaluation of the asset's β. To evaluate these, a market baseline is required. Since this research focuses around NASDAQ-traded securities, the NASDAQ Composite Index will be used [59]. The expected annual market return will be computed by annualising[2] the average daily percentage change over a specified period. This period, represented by N days, will serve as an adjustable hyperparameter in the model.

$$\begin{aligned}
\delta_{m,t} &= \frac{V(m_t) - V(m_{t-1})}{V(m_{t-1})} \\
R_{m,t} &= \begin{bmatrix} \delta_{m,t-N} & \delta_{m,t-(N-1)} & \delta_{m,t-(N-2)} & \cdots & \delta_{m,t} \end{bmatrix} \\
\mathbb{E}_{\text{daily}}(R_{m,t}) &= \overline{R_{m,t}} \\
\mathbb{E}_{\text{daily}}(R_{m,t}) &= \frac{\sum_{i=0}^{N} \delta_{t-i}}{N} \\
\mathbb{E}_{\text{annual}}(R_{m,t}) &= (1 + E_{\text{daily}}(R_{m,t}))^{252} - 1
\end{aligned}$$

The asset's daily returns $R_{i,t}$ are calculated in an equivalent fashion to $R_{m,t}$, except instead of using market price m, the asset price p is used. The final bit of data required is the value of the 3-Month US Treasury Bill Rate at time t, this will serve as the risk free rate $R_{f,t}$. From here, it is a simple process to calculate expected annual returns by expanding on Equation 2.3:

$$\mathbb{E}_{\text{annual}}(R_i) = R_{f,t} + \beta_i(\mathbb{E}_{\text{annual}}(R_{m,t}) - R_{f,t}) \tag{3.13}$$

Where:

$$\beta_i = \frac{\text{Cov}(R_{i,t}, R_{m,t})}{\sigma^2_{R_{m,t}}} \tag{3.14}$$

Once this model had been developed, it was tested by applying it to the historical financial data of *Apple* as seen in Figure 3.2. This figure shows that there is strong correlation between the value of *Apple's* stocks and the value of the NASDAQ Composite Index. As a result, the calculated expected returns reflect the state of the overall market, forecasting negative returns in periods of recession such as 2003 and 2008, and positive returns in periods of market growth.

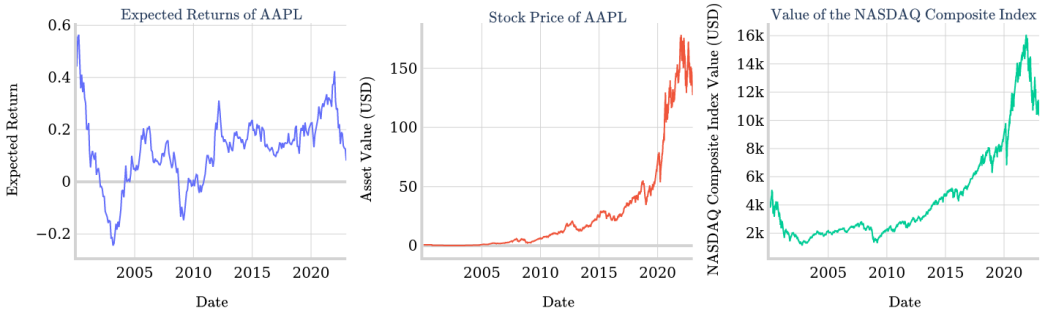


FIGURE 3.2: Predicted expected returns of AAPL using CAPM alongside AAPL and NASDAQ Composite Index Value from 2000 - 2021

3.5.2 Regression Model

In an effort to approximate the momentum factor *MOM* of the Carhart model, a linear regression analysis will be applied to each of assets. This regression, when applied to an asset at time t, will consider the last P days of financial data (where P will be an adjustable hyperparameter), calculate the cumulative ROI for each day within that period, and derive a relationship between the date and

[2]There are 252 trading days in a calendar year.

the ROI. While the specific relationship between the date and the asset value isn't especially helpful, the gradient of the line of best fit will provide a clear and concise indication as to whether the asset is on a upward or downward trajectory. From there the RL model can make it's own deductions about whether to invest in assets that are either rising or falling in value. In general, the time complexity of linear regression is $O(nm^2 + m^3)$ where m is the number of descriptive features, and n is the number of data points [60]. However, in this instance there is only one feature (the date), therefore this process is $O(n)$ per asset. Linear approximations (blue) for the cumulative returns (red) for *Apple*, *Cineverse*, and *ACI Worldwide* over the course of 2015 are shown in Figure 3.3.

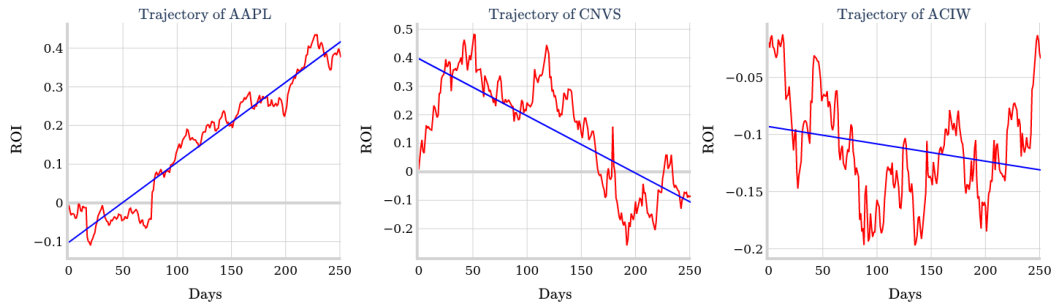


FIGURE 3.3: Linear regression models applied to the cumulative returns of AAPL, CNVS, and ACIW in 2015

3.5.3 Volatility

Asset volatility is a critical measurement for the observation space. Although volatility itself does not directly contribute to an asset's expected return, it serves as a crucial indicator of the risk associated with investing in that asset. By quantifying the extent of price fluctuations, volatility provides the RL model with essential information about the uncertainty and potential variability in returns. Integrating volatility into the observation space allows the RL model to balance risk and reward, tailor investment strategies to match risk tolerance, and make more informed decisions about portfolio allocation. Consequently, asset volatility helps the model to navigate market dynamics and manage potential risks, thereby enhancing overall portfolio performance and stability.

Volatility is simply calculated by taking the standard deviation of the daily asset returns over the past Q days (where Q will be an adjustable hyperparameter). This is a constant time process being repeated for each asset, making this process $O(n)$. The technique for calculating this value has already been seen in (3.14), where the variance of the daily market returns $\sigma^2_{R_{m,t}}$ was calculated. By applying the same technique, the volatility of asset i, $\sigma_{R_{i,t}}$ can be calculated.

Once the volatility model has been developed, it was tested by computing the volatility in the price of Apple shares from July 2023 to July 2024 with $Q = 1$ year. The results were considered accurate as the model appropriately assigned higher volatility to periods with significant price fluctuations over the preceding year and lower volatility to periods characterised by consistent price changes (steady growth). These findings can be seen in Figure 3.4 - the start of the volatility calculation period is shown by the red vertical line.

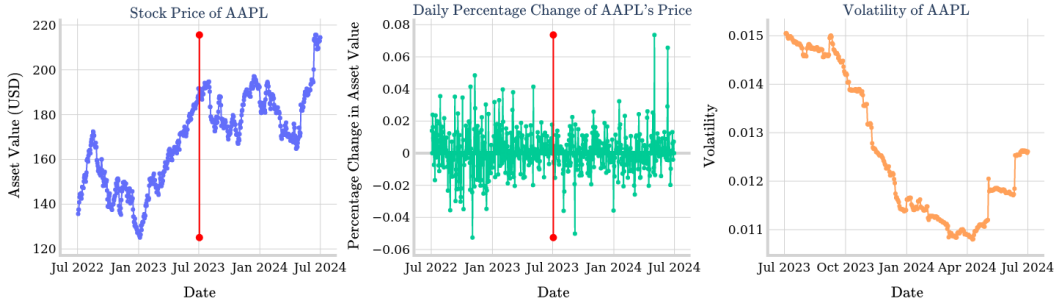


FIGURE 3.4: Volatility calculation procedure demonstrated on AAPL between from July 2023 - July 2024

3.5.4 Amihud Illiquidity Ratio

Liquidity refers to the ease with which an asset can be bought or sold in the market without significantly affecting its price. Highly liquid assets, like large-cap stocks, can be traded in large quantities with minimal price impact, while illiquid assets may experience substantial price changes even with small trades. The Amihud Illiquidity Ratio, developed by Yakov Amihud in 2002 [61], is a widely used measure that quantifies this price impact by examining the relationship between absolute price changes and trading volume. Specifically, the Amihud ratio captures the average daily price response associated with one unit of trading volume, providing a clear indicator of how illiquid a stock is. Higher values of the Amihud ratio indicate greater illiquidity, suggesting that even small trades can lead to significant price movements. This measure is particularly useful for assessing the liquidity risk associated with investing in smaller or less frequently traded stocks.

The Amihud illiquidity ratio, which measures the price impact of trading volume, can be closely linked to the Small Minus Big SMB factor in the Fama-French model, which captures the size premium between small-cap and large-cap stocks. Smaller firms, typically characterized by lower trading volumes and higher illiquidity, tend to exhibit higher Amihud ratios. This suggests that small-cap stocks, which contribute to the SMB factor, generally face greater liquidity risk, as even minor trades can cause significant price fluctuations.

The assumption that large-cap assets have a high mean trading volume and therefore low illiquidity, and small-cap have a low mean trading volume and therefore high illiquidity would create an inverse relationship between the Amihud ratio and the market capitalisation of an asset. While this assumption isn't ideal, the lack of available data for the market capitalisation of NASDAQ-listed securities makes it necessary. Similar to the expected returns, linear regression and volatility, the Ahimud Ratio at time t is calculated based off the financial data from the previous S days (where S will be an adjustable hyperparameter). The formula for calculating the Amihud illiquidity ratio can be seen in Equation 3.15.

$$AR_t = \frac{1}{S}\sum_{i=1}^{S}\frac{|R_{t-i}|}{V_{t-i}} \tag{3.15}$$

Where:

$$|R_t| = \text{The absolute daily return at time } t$$
$$V_t = \text{The volume traded at time } t$$

Once the model had been developed, the Amihud ratio was calculated for for all assets in the asset universe, centred around the January 2024 with a look back period of $S = 5$ years. Figure 3.5 shows a clear inverse relationship between the mean traded volume and the Amihud ratio, validating the results of the model. The μ symbols on the y-axis represent $\times 10^{-6}$

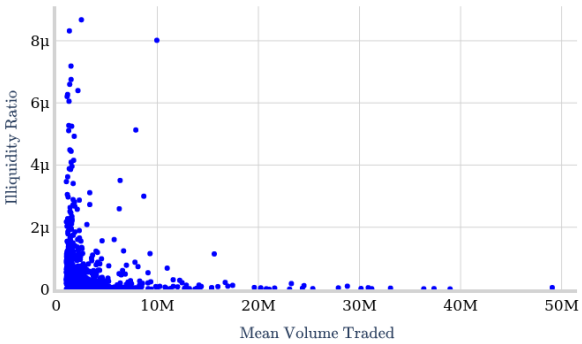


FIGURE 3.5: Illiquidity ratio against Mean Traded Volume

3.6 Evaluation Metrics

To evaluate the performance of the RL model, several metrics will be employed. For the most part, these metrics are drawn from the reward function (3.5) as these are the values that are trying to be maximised. The key difference for the evaluation however is that actual returns are considered rather than expected returns when calculating the risk adjusted ratios.

1. **ROI:** This will measure the absolute return of the portfolio over a specific period, providing the most straightforward view of its performance. This metric shows the measured the effectiveness of the RL model in generating profits.

2. **Risk-adjusted Returns:** While the reward function described in (3.5) looks at the risk-adjusted **expected returns**, a critical evaluation point is the risk-adjusted **actual returns**. Applying the Sharpe, Treynor, and Sortino Ratios to the evaluation will quantify the the rate of actual returns relative to the overall, systematic and downside risks respectively. Furthermore additional volatility analysis can be carried out, to determine how the portfolio managed asset volatility throughout the testing period. Identifying any correlations between the portfolio's volatility and risk-adjusted returns can not only give insights into potential strategies being adopted by the RL model, but also indications as to how successfully it implemented these strategies.
3. **Portfolio Entropy:** This is a critical point of evaluation as it measures one of the most fundamental aspects of portfolio management - diversification. Portfolio's with a higher entropy levels have a more diverse range of investments, whereas those with lower entropies have more concentrated investments in fewer assets.
4. **Weighted Mean Asset Percentile (WMAP):** This new evaluation metric has been devised specifically for this research evaluation to assess the portfolio's asset selection abilities. At each time step t, each asset i will be ranked based on their cumulative ROI over the last 30 time steps, from there each asset can be assigned a percentile $\%\text{ile}_{i,t}$, indicating it's overall relative short term performance. By looking at the weighted average of assets the portfolio has invested in, the portfolio's asset selection abilities can be assessed. Consistently achieving a weighted average above the 50% line would signify that the portfolio's asset selection is above average, reflecting strong performance in identifying and investing in outperforming assets.

$$\text{WMAP}_t = \sum_{i=1}^{M} \omega_{i,t} \cdot \%\text{ile}_{i,t} \tag{3.16}$$

3.7 Dividing Historical Data in Training and Testing Data

For this study, the historical data is divided into training and testing periods to evaluate the performance of the RL models under various market conditions. The models will be trained on data from 01/01/2012 to 31/12/2021, which provides a substantial period of recent market history. This period includes a variety of market environments, characterised by different economic conditions, such as the recovery from the global financial crisis and the market fluctuations during the COVID-19 pandemic. Training on such a diverse range of market behaviours should help ensure that the models are well-prepared to handle various scenarios and not overfit to any specific market phase. During the training process, the model will loop through the training period repeatedly. The episode will reset if one of two conditions are hit:

1. The current date in the market environment exceeds the end of the training period
2. The portfolio ROI becomes less than or equal to -0.9. This means it has lost the vast majority of it's value, triggering a premature termination of the episode.

It is crucial that the testing data does not overlap with the training data to ensure that the evaluation of the models reflects their ability to generalise to unseen market conditions, rather than simply recalling patterns from the data they were trained on. To comprehensively test the models' adaptability, the testing phase will be conducted in two distinct periods. The first test period will use data from 2006 up to 2012 to evaluate the models' ability to adapt to adverse market conditions, such as those experienced during the 2008 global financial crisis. While it may seem unorthodox to test a model on data that chronologically precedes the training period, this period is selected to assess how well the models can handle extreme downturns and maintain performance in highly volatile environments, a critical capability for robust portfolio management.

The second test period will use data from 2021 to mid-2024, providing an opportunity to evaluate the models against the most current version of the market. Testing on this recent data ensures that the models are not only robust in historical contexts but also effective in navigating contemporary market dynamics, capturing any recent trends, changes, or anomalies in the financial markets. By using these distinct test periods, the study aims to rigorously assess the models' performance across a variety of challenging and evolving market conditions.

4. Implementation

4.1 Reinforcememnt Learining Model Implementation

This research utilises the PPO and DDPG models from the Stable-Baselines3 (SB3) library [62, 63]. This software was chosen for its overall reliability and because it is implemented in PyTorch [64], which enables graphics processing unit (GPU) acceleration during the training process. PyTorch's integration with CUDA, a parallel computing platform and application programming interface (API) model created by NVIDIA [65], allows for efficient computation on NVIDIA GPUs. Using CUDA can significantly speed up reinforcement learning training by enabling multi-threading, which reduces training time and allows more complex models to be processed efficiently.

The primary argument for an SB3 model is an environment object that adheres to the OpenAI Gym interface (`gym.Env`) [66]. This interface provides a a standardised way for RL models to interact with the environments they are trying to solve. The key methods in this interface are `step(action)` which describes how the environment will respond to a given action, and `reset()` which describes what state the model should revert to when a training episode is terminated.

4.2 Object Orientated Architecture

This project was implemented using an object orientated paradigm. The most fundamental unit is the `Asset` class, this stores the financial of a given asset and has a series methods to evaluate the features described in Section 3.5. The `Asset` class is versatile, capable of not only storing information about individual assets, such as Apple, but also maintaining histories of macroeconomic series, such as the value of the NASDAQ Composite Index and the 3-Month US Treasury Bill. Above this, the `Collection` class is defined, which takes in a dictionary of ticker[1] - `Asset` pairs. Using the asset ticker as the key enables rapid lookup compared to a standard list comprehension. This structure allows for efficient management and retrieval of multiple `Asset` instances. The `Collection` class has three extensions. The `AssetCollection` class, the `PortfolioCollection` class, and the `MacroEconomicCollection` class. Each one has a different collection of `Asset` instances, and different responsibilities.

- `AssetCollection` stores all tradable `Asset` instances that the model can buy or sell. It's primary task is collection the observations from all `Asset` instances.
- `MacroEconomicCollection` stores the histories of macro economic series and returns the "current" interest, unemployment and risk-free rate to the observation.
- `PortfolioCollection` stores the `Asset` instances the model is currently invested in. It's primary purpose is to track portfolio value and the various risk-adjusted return metrics and evaluate the model's current investment strategy.

The highest point in this class hierarchy is the `TradingEnv` class. This is takes in all the relevant `Collection` instances and establishes a trading environment. This class extends the `gym.Env` interface, translating actions into buy and sell orders in the `step()` method.

This object orientated hierarchy helps significantly with data organisation and overall code maintainability. The modular structure allowed for new features to be integrated and tested in a quick and effective manner. This approach will also help greatly with regards to scalability, with no modifications being required to facilitate a larger asset universe. The designed object architecture can be seen in Figure 4.1.

[1] In finance, a ticker is a unique symbol or series of letters assigned to a publicly traded company or security for identification purposes on a stock exchange. For instance Apple is 'AAPL'

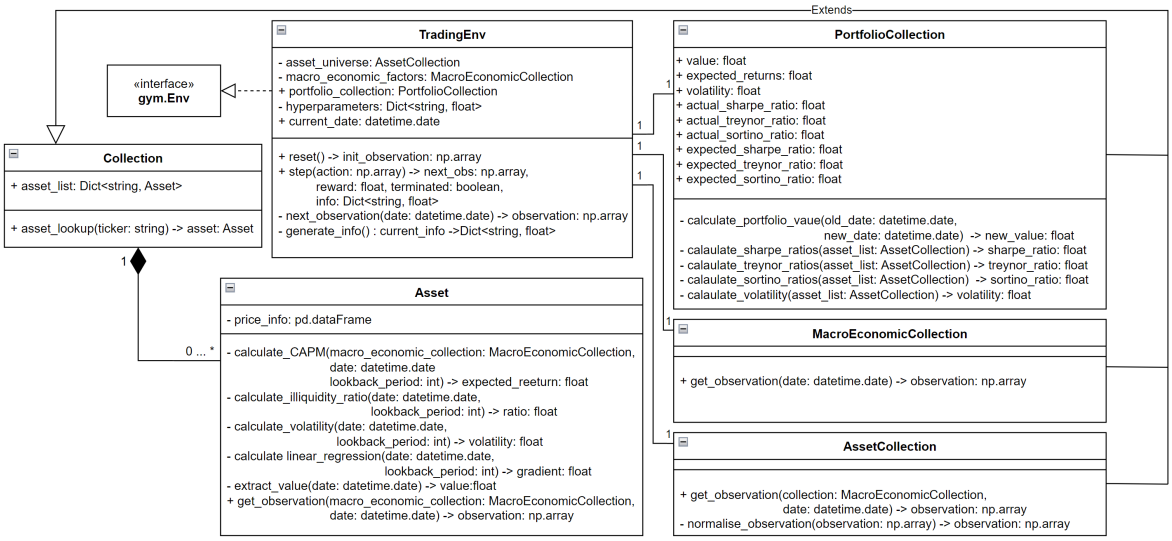


FIGURE 4.1: UML Class Diagram Showing the Relationships Between **'TradingEnv'** and Asset Management Classes.

4.2.1 Model Training Procedure

In order to take full advantage of the hardware optimisations described in Section 4.1 that come with SB3, model training was done in a Google Colab environment [67]. This environment was chosen as it provides easy and free access to a NVIDIA T4 GPU - a powerful option for deep learning tasks due to its tensor cores, which accelerate matrix operations common in neural network training. It also provides paid access to the more powerful NVIDIA A100 GPU however, the A100's capabilities will likely exceed the computational requirements for the sample dataset used in this project. To allow the Google Colab environment to interact with the local class files, a Google Drive intermediary and local Python synchronisation script were created. This script, when run, performs two key functions. Firstly, it uploads the latest versions of all environment management classes and hyperparameters to Google Drive. Secondly, it downloads new models and log files to the local instance. Once the class files and hyperparameters document are in Drive, they can be seamlessly imported into the Google Colab environment, allowing model training to commence. Figure 4.2 illustrates the workflow for file synchronisation and model training.

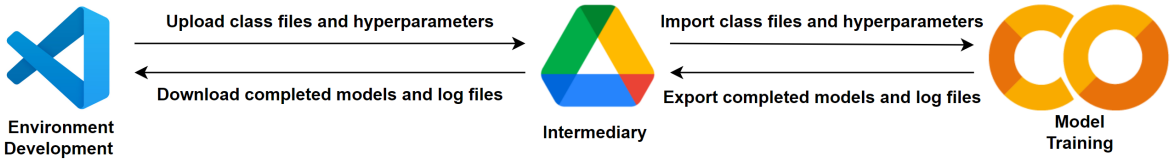


FIGURE 4.2: Transmission of information between a local instance, Google Drive and Google Colab

Keeping the environment development process in a local setting makes it significantly more convenient to iteratively develop, refine, test, and debug the class files where it is easier and quicker to implement changes - ensuring the robustness and accuracy of the environment management code before scaling up to full model training. The use of the Google Colab GPU resources is reserved specifically for the very computationally expensive task of model training. This approach optimises the workflow by allowing local environment development and remote model training to happen in parallel.

5. Results and Discussion

The primary objective of this section is to systematically present the results of trained PPO and DDPG models through a comprehensive analysis of the key performance metrics outlined in Section 3.6. Multiple trials were conducted with different reward functions and parameter configurations for both models. The results of the highest performing configurations with regards to the most general form of risk-adjusted returns - the Sharpe Ratio - for PPO and DDPG are presented in Section 5.2. This approach allows for a detailed evaluation of each model's effectiveness, enabling comparisons based on the returns they generated and the strategies they employed. The conclusion of this section identifies which RL technique performs better in this application. Following on from this, a sensitivity analysis will be performed on the selected RL model, to determine how modifying the reward function affects the chosen policy and the overall results. Ultimately the final model can then be scrutinised according to the the project objectives outlined in Table 1.1 to determine whether this model has achieved their respective success criteria.

5.1 Market Analysis Over the Test Domain

5.1.1 Test Period #1: 2006 - 2012

This first period, as seen in Figure 5.1, can be split into three distinct sections, the pre-crisis period, where the US economy experienced robust and steady growth and therefore the NASDAQ, like other major stock indices, benefited from this positive economic sentiment. In fact in 2007, the NASDAQ Composite Index hit an all time high. Then in mid-2008, the Global Financial Crisis hit, sending the economy into a downward spiral. The NASDAQ experienced a dramatic downturn, loosing nearly 55% of its value Between October 2007 and March 2009, highlighting the scale of the downturn. This period was characterised by high volatility, with significant daily swings in the market as uncertainty gripped investors. The final section in this period is the post-crisis recovery period. Here, the US Federal Reserve implemented unprecedented monetary and fiscal stimulus measures to stabilise the economy such as setting interest rates to near zero. This helped to restore investor confidence and the NASDAQ began to recover as early as mid-2009, with the recovery accelerating through 2010 and beyond. By 2012, the NASDAQ had regained its lost ground. This six-year period includes phases of steady economic growth, rapid decline, and swift expansion, providing a diverse range of scenarios for any reading algorithm to navigate.

FIGURE 5.1: Value of the NASDAQ Composite Index from 2006 to 2012

5.1.2 Test Period #2: 2021 - 2024

The second test period, as seen in Figure 5.2, is characterised by significant volatility and shifts in market dynamics. Initially boosted by post-pandemic recovery efforts, including government stimulus and low interest rates, the NASDAQ saw substantial growth. However, by late 2021, rising inflation due to supply chain disruptions, labour shortages, and increased demand raised concerns among investors. In response, the Federal Reserve raised interest rates to combat inflation, leading to increased market volatility. This caused the NASDAQ's value to decline in early 2022, which was further impacted by geopolitical tensions, such as the Russian invasion of Ukraine in February 2022, prompting investors to shift towards safer asset holdings. Since 2023, economic uncertainty has persisted, with fears of a recession due to ongoing monetary tightening. The NASDAQ has shown mixed performance, with recovery phases interrupted by sell-offs. Ongoing geopolitical tensions, particularly between the U.S. and China, continue to create a volatile environment. In such an unpredictable scenario, models may struggle to predict price movements accurately and identify profitable opportunities, leading to increased uncertainty and potentially sub-optimal decision-making.

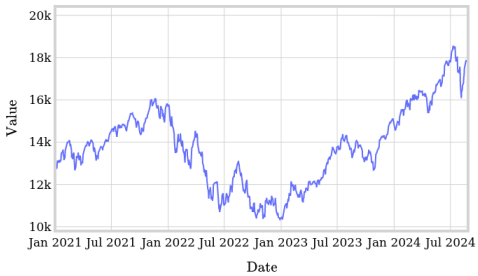


FIGURE 5.2: Value of the NASDAQ Composite Index from 2021 to 2024

5.2 Performance Evaluation of PPO and DDPG

5.2.1 Return on Investment

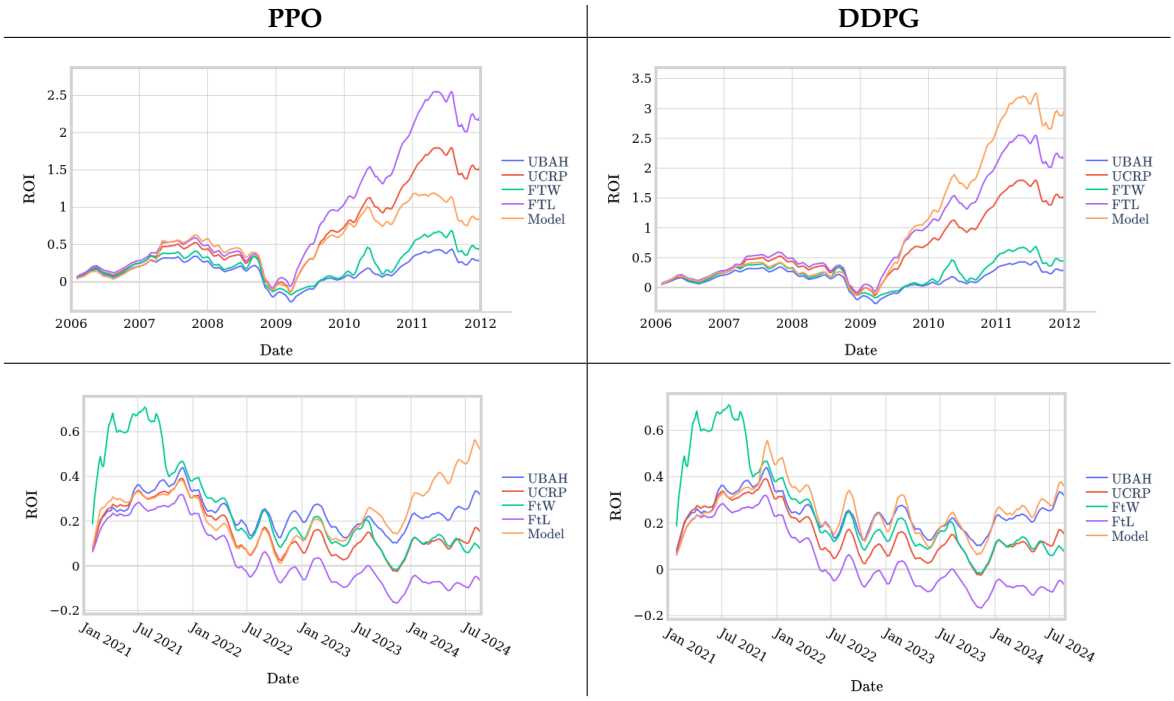


TABLE 5.1: ROI achieved by PPO and DDPG across both testing periods compared to the baseline strategies

Analysis

Test Period #1 With regards to ROI, the strongest baseline strategy during the 2006 to 2012 testing period was the 'follow-the-loser' strategy with the majority of its growth coming after the 2008 crash in the recovery period. This is likely due to the fact that this strategy would have cultivated large positions in the biggest losers after the crash. As these heavily discounted assets rebounded in the recovery, they experienced substantial gains, contributing to the overall success of the strategy. In this period, PPO and DDPG show near-equivalent returns up until the start of the recovery period in early 2009. At this point DDPG displayed phenomenal performance, outperforming all baseline strategies comfortably, reaching a peak ROI of 3.35 in mid-2011. PPO however did not perform to the same level. While it did mange to produce positive returns, beating the buy-and-hold and the follow-the-winner strategies, it was unable to outperform even the uniform URCP indicating that it's asset selection strategy was below average.

Test Period #2 The strongest overall baseline strategy with regard to ROI over the 2021 - 2024 testing period was the uniform buy-and-hold strategy. In a period of investor uncertainty and volatility, the strategy that passively follows the market trend delivered the most steady returns, outperforming the other active-strategy baselines. Over this period, both PPO and DDPG managed to outperform all the baseline strategies - but did so in different manners. DDPG showed consistently high returns either being the highest performing method or close to it at each time-step, whereas PPO shows relatively underwhelming returns up until mid-2023, where it achieved a sudden surge relative to the market trend. As a result it ended up comfortably being the highest performing method over this period.

5.2.2 Risk Adjusted Returns Analysis

Cumulative Sharpe Ratio

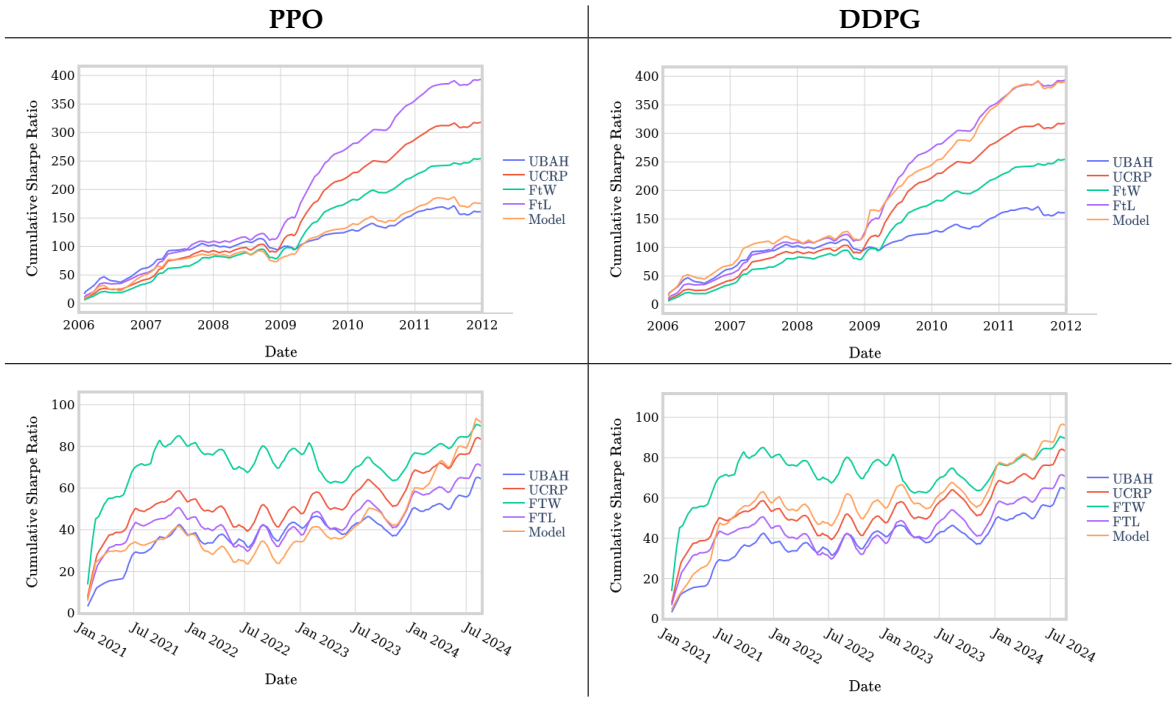


TABLE 5.2: Cumulative Sharpe Ratio achieved by PPO and DDPG across both testing periods compared to the baseline strategies

Recap: Sharpe Ratio measures the portfolio's excess returns (returns relative to the risk-free rate) relative to the overall volatility of the portfolio. A consistently high Sharpe Ratio implies that the portfolio is effectively managing its total risk and delivering strong returns.

Analysis

Test Period #1 While the general trends of the Cumulative Sharpe Ratio (CShR) are similar to that of the ROI for both PPO and DDPG, there are some slight differences. DDPG, displayed similar performance in CShR relative to the strongest baseline candidate - FtL. This fact coupled with the fact that it produced a higher ROI than FtL implies that these excess returns were created while exposing the portfolio to a higher degree of risk. While this may seem concerning at first, the fact that DDPG managed to achieve a higher overall ROI while maintaining similar levels of risk-adjust returns shows that it is effectively pushing the risk levels to leverage higher returns. Unfortunately, PPO's performance significantly declined in this metric relative to its performance in ROI. This shows that the returns generated by this model were done so through a high degree of portfolio risk, and significant exposure to negative returns. This further supports the suggestion that it's asset selection is suboptimal.

Test Period #2 The increased relative performance of the FtW strategy can be explained by the very volatile and profitable (see Tables 5.5 and 5.6) position it held early in this period, shown by the spike in ROI at the start of the period as seen in Table 5.1. The trends shown by PPO and DDPG here are very similar to the ROI metric. On the one hand, PPO displays relatively poor risk-adjusted returns throughout the period with the overall performance being saved by a 'spike' in returns. On the other hand DDPG displays a consistently high risk-adjusted returns, ultimately ending the period with the highest CShR due to the poor risk management capabilities of FtW.

Cumulative Treynor Ratio

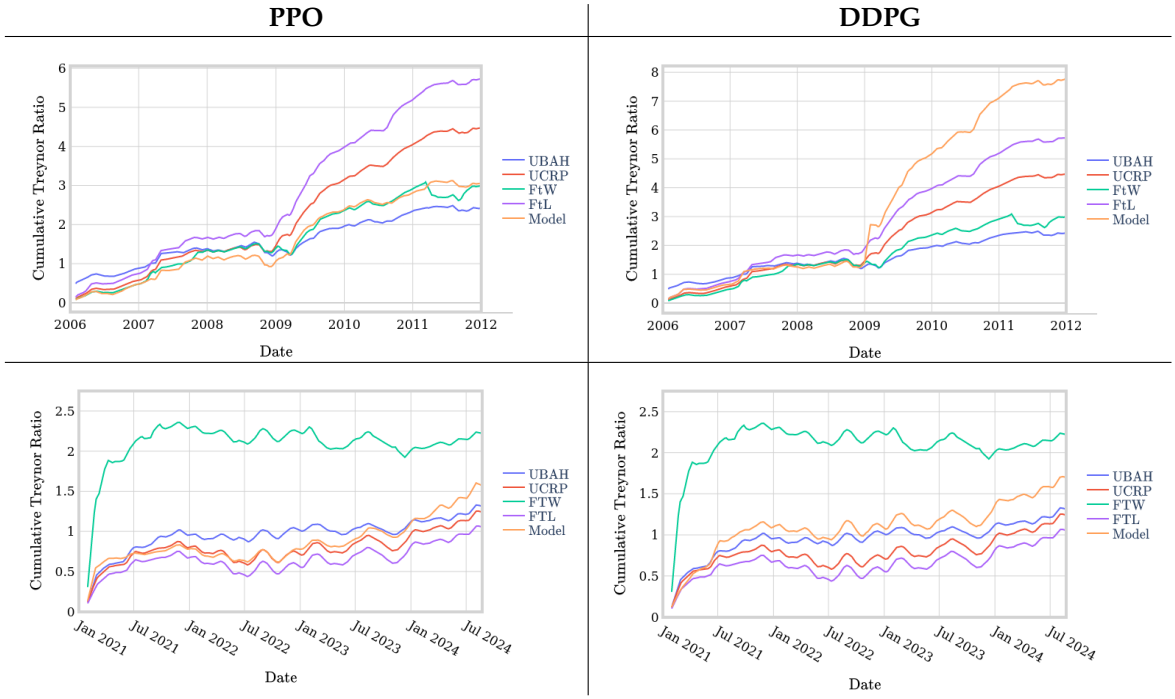


TABLE 5.3: Cumulative Treynor Ratio achieved by PPO and DDPG across both testing periods compared to the baseline strategies

Recap: Treynor Ratio measures the the portfolio's excess returns relative to the beta of the portfolio which is the sensitivity of the portfolio's returns relative to market movements (systematic risk). A high Treynor Ratio means the portfolio is effectively generating returns relative to its exposure to market risk.

Analysis

Test Period #1 This period reveals a similar trend to the CShR, with the Treynor Ratio further emphasising the impressive performance of the DDPG model in terms of risk-adjusted returns. Specifically, the DDPG model demonstrates outstanding returns when accounting for systematic risk. This suggests that the DDPG model is effectively managing a portfolio that not only generates strong returns but does so with minimal exposure to systematic risk relative to the market. Again, PPO under-performed relative to its counterpart, as well as the FtL and UCRP strategies. This shows that it is not efficiently managing systematic risk, which is particularly concerning during periods of market decline, for instance 2008. During such downturns, high sensitivity to market risk (beta) can lead to substantial losses as the entire market falls. It also shows that such exposure is not being met with adequate rewards to justify it.

Test Period #2 Similar to the CShR for this period, the spike in the Cumulative Treynor Ratio (CTR) for the FtW strategy slightly skews the results allowing it to comfortably finish the period with the highest CTR. However during this period, the DDPG model demonstrated the most consistent growth in CTR, indicating a steady ability to generate returns that compensated for market risk. This consistency is very impressive considering the volatile nature of recent markets managing to account for systematic risk while delivering reliable returns. In contrast, the PPO model showed a slower and more gradual increase in CTR throughout the period. The eventual spike in portfolio value towards the end of the testing phase significantly boosted its risk-adjusted returns. However, since this late surge aligns with the increase in ROI in this period, it likely reflects a fortunate gain in a volatile position's value rather than an improved risk management strategy.

Cumulative Sortino Ratio

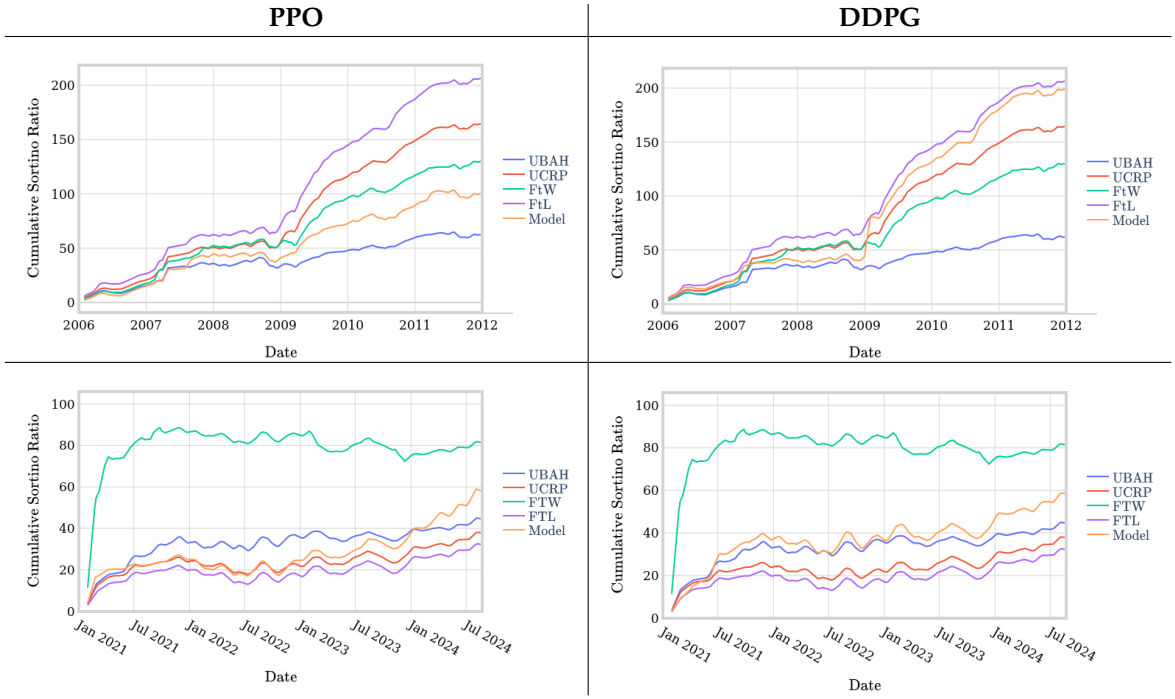


TABLE 5.4: Cumulative Sortino Ratio achieved by PPO and DDPG across both testing periods compared to the baseline strategies

Recap: Sortino Ratio is similar to the Sharpe Ratio except it only measures the excess return relative to the downside risk (the standard deviation of negative returns). A portfolio that consistently achieves a high Sortino Ratio is effectively managing downside risk by minimising losses and avoiding significant drawdowns.

Analysis

Test Period #1 This period shows the same trend between DDPG and PPO as with the other risk-adjusted ratios in this period. DDPG displays a more prominent progression of Cumulative Sortino Ratio (CSoR) in the market recovery period indicating a stronger ability to generate returns while effectively minimising downside risk. However, in this instance, the 'Follow-the-Winner' (FtW) strategy outperformed DDPG, unlike with the CShR and CTR metrics. This implies that DDPG is less effective at managing downside risk compared to its handling of overall risk and systematic risk. The fact that FtW achieved better results in terms of CSoR suggests that DDPG may struggle more with protecting against losses during market downturns, highlighting a potential weakness in its ability to limit exposure to negative returns.

Test Period #2 Again, similar to the CShR and CTR, the sharp spike at the start of the period makes FtW the highest performer overall. However DDPG shows a more steady increase in CSoR compared to PPO and FtW, indicating a more consistent and comprehensive approach to risk management. This suggests that DDPG effectively manages downside risk throughout the period, steadily enhancing its risk-adjusted returns. In contrast, PPO and FtW rely on sudden spikes in the value of specific positions to achieve higher levels of risk-adjusted return, pointing to a less stable and reactive risk management strategy that may not consistently protect against downside risk.

Mean Portfolio Asset Volatility

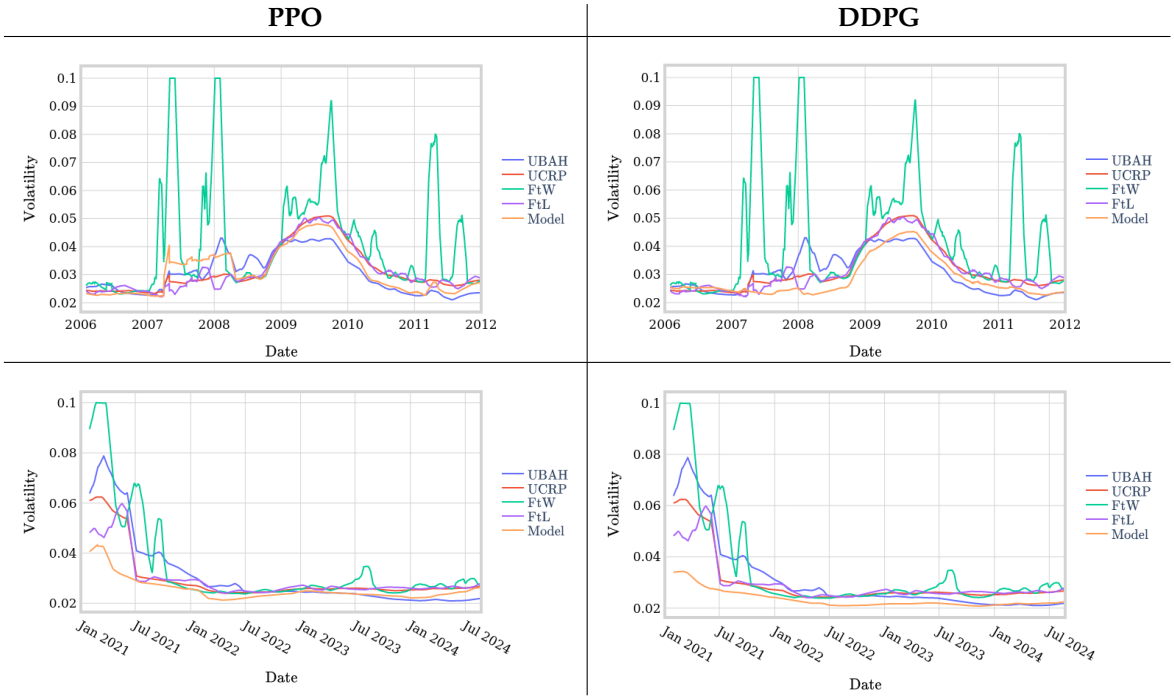


TABLE 5.5: Mean Portfolio Volatility achieved by PPO and DDPG across both testing periods compared to the baseline strategies

This is the weighted average across the portfolio of the volatility metric described in Section 3.5.3. Table 5.5 shows that PPO tended to select assets with higher volatilities, whereas DDPG opted for assets with lower volatilities. While this strategy cannot be criticised on the surface, the discrepancy between returns and risk-adjusted returns for PPO and DDPG can be explained by this data.

Analysis

With the selection of higher volatility assets, PPO is likely aiming to capture larger price movements, which can generate higher returns if the asset selection and position timing are correct. However, this approach comes with an increased degree of risk, making the portfolio more susceptible to sudden downturns. This explains how PPO was able to achieve a higher overall ROI in Test Period #2 despite having lower risk-adjusted returns across all three metrics in both testing periods. The price 'spike' discussed in the risk-adjusted return analysis resulted from a volatile investment experiencing a steep increase in price, highlighting the trade-off between potential gains and increased volatility.

An interesting additional observation is that the only period where DDPG consistently exhibited higher average portfolio volatility was during the later stages of the first test period, from around early 2010 to early 2011. This period coincided with DDPG's ROI significantly outpacing that of FtL, further demonstrating DDPG's ability to effectively leverage increased risk to achieve greater returns.

While the appropriate asset volatility level is a subjective measurement depending on the preferences of the investor, for this study, having a lower average asset volatility is considered positive, as it indicates a more effective portfolio management strategy. Lower volatility suggests that the portfolio is better at minimising risk and avoiding significant drawdowns, which are crucial for achieving stable returns over time. Furthermore, this ethos aligns with one of the fundamental principals of Modern Portfolio Theory - minimising the variance of the expected returns (risk) as described in Section 2.1.1.

5.2.3 Weighted Mean Asset Percentile

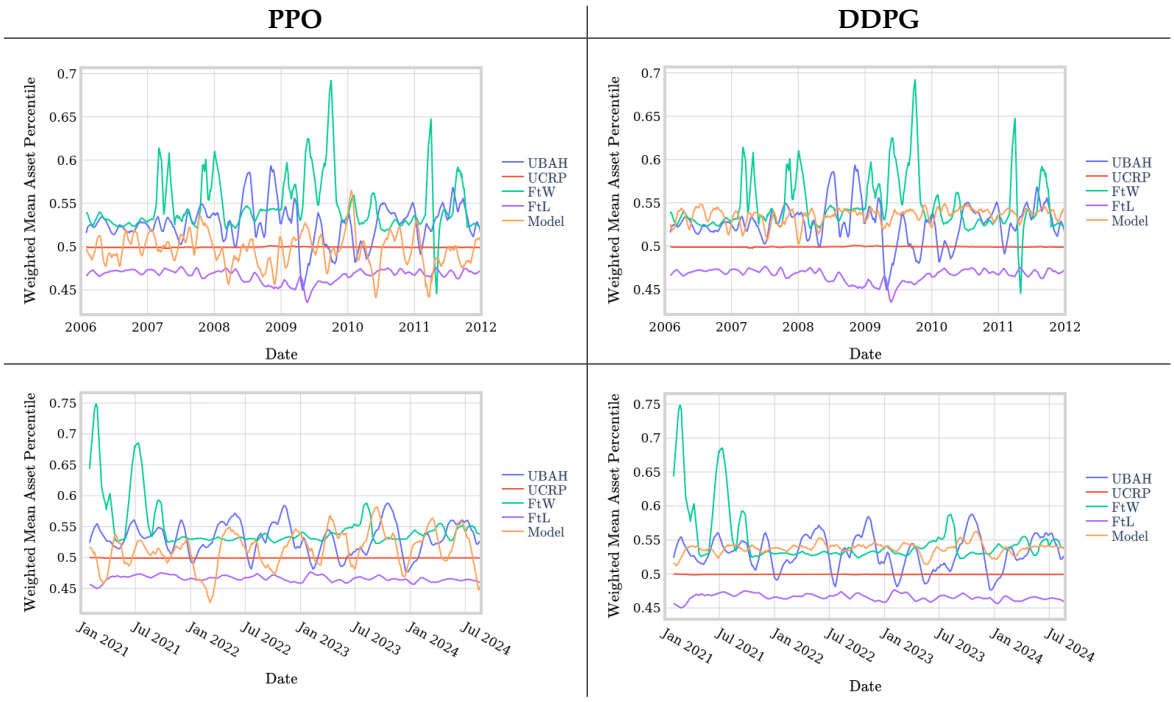


TABLE 5.6: Weighted Mean Asset Percentile achieved by PPO and DDPG across both testing periods compared to the baseline strategies

The results of the Weighted Mean Asset Percentile study reveal that DDPG consistently achieves a higher average percentile across both test periods compared to PPO, along with a lower standard deviation. This outcome suggests that DDPG demonstrates superior asset selection abilities, effectively identifying and investing in assets that perform above the market average. By consistently choosing higher-performing assets, DDPG is able to enhance its return potential while maintaining a more stable portfolio.

The lower standard deviation in the Weighted Mean Asset Percentile metric coupled with an average greater than 0.5 indicates that DDPG's asset selection strategy is not only successful but also reliable. This consistency in asset selection reduces the portfolio's exposure to unpredictable fluctuations, which is crucial for managing overall portfolio volatility. By maintaining a focus on assets that offer stable, above-average returns, DDPG minimises the risk of significant drawdowns, thereby providing a smoother return profile.

This effective asset selection and volatility management directly contribute to DDPG's ability to achieve better and more consistent risk-adjusted returns. By focusing on high-quality assets that are less prone to volatility, DDPG ensures that its returns are not only competitive but also more predictable and less susceptible to market swings. This aligns with the key principles of portfolio management, where the goal is to maximise returns while controlling for risk.

In contrast, the higher variance seen in PPO's Weighted Mean Asset Percentile suggests a less consistent approach to asset selection, leading to greater variability in performance. This variability can introduce a higher degree of risk into the portfolio, which may negatively impact risk-adjusted returns, as observed in the lower Sharpe, Treynor, and Sortino ratios.

5.2.4 Portfolio Entropy

TABLE 5.7: Portfolio Entropy achieved by PPO and DDPG across both testing periods compared to the baseline strategies

Recap: Entropy is a measure of the diversification in a portfolio, the higher the portfolio entropy, the more spread out it's investments are. UCRP is the gold-standard for this metric as it maintains a even split amongst all assets, meaning it will have the highest feasible entropy for any asset universe it resides in.

Across both periods, DDPG maintained a higher portfolio entropy, having consistent investments in approximately 61% of the assets in the asset universe across both periods, meaning it placed a higher value on portfolio diversification than PPO which had consistent investments in 41.5% of assets in Test Period #1 and only 17.6% in Test Period #2. This indicates that DDPG avoided over-concentration in a few assets and instead spread its capital more evenly. This is advantageous as it reduces exposure to idiosyncratic (asset-specific) risk. The notably lower level of diversification displayed by PPO in Test Period #2 not only explains the larger fluctuations in the weighted mean asset percentile over this period but also the how the this portfolio was able to see a sharp increase if portfolio value in the later stages of this period. This lack of diversification gives the portfolio a greater exposure to idiosyncratic risk, which if handled correctly can lead to greater returns.

However, the linear nature and the similarity of the entropy levels on both test periods displayed by the DDPG portfolio is concerning. While the steadiness of the respective lines can be explained by the fact that the results presented in this section so far are the moving average plots of the raw data to increase with readability - the fact that the average remains so consistent across both periods shows that the DDPG portfolio isn't adapting it's strategy to market conditions. While this isn't necessarily a negative point, as maintaining a consistent and high level of diversification can protect against specific asset risks, it suggests a lack of responsiveness to market dynamics. This rigidity could limit the model's ability to capitalise on emerging opportunities or adjust to adverse conditions

5.2.5 PPO - DDPG Comparison

The results presented so far in this section have generally been in favour DDPG - displaying that it has more reliable asset selection capabilities - as shown by its ability to consistently identify and allocate capital to assets that perform above the market average. Furthermore, it's superior risk management strategy of primarily focusing on assets that exhibit less variability in performance protects the portfolio from significant drawdowns during periods of market turbulence, creating more consistent returns and therefore higher risk-adjusted returns. Table 5.8 shows a summarised comparison between PPO and DDPG across the key metrics that have been discussed in this section. From this it is clear to see that DDPG has formed the more effective investment strategy, outperforming PPO in nearly every category. Furthermore, there was no single baseline strategy that consistently outperformed DDPG in both test periods. While the FtL strategy delivered comparable results in the first test period, this can be attributed to the specific market conditions during that time — a sharp downturn followed by a strong recovery. These conditions are ideal for the FtL strategy to generate strong returns for the reasons discussed at the start of Section 5.2.1. Its poor performance in the second test period highlights the strategy's limitations when market conditions are less favorable. This displays the robustness of the DDPG method - being able to minimise loses during a market crash, capitalise on the opportunities by taking on increased risk during the market recovery phase to leverage more significant returns in the first test period and generate steady profits during the volatile second test period.

	Validation Period #1 01/01/2006 - 31/12/2011		**Validation Period #2 01/01/2021 - 2024**	
Category	**PPO**	**DDPG**	**PPO**	**DDPG**
Overall ROI	0.88436	**3.051018**	**0.552116**	0.397407
Cumulative Sharpe Ratio	177.99	**394.63**	93.64741	**99.71256**
Cumulative Treynor Ratio	3.075891	**7.825835**	1.613442	**1.766213**
Cumulative Sortino Ratio	101.1034	**201.6412**	59.31999	**60.44954**
Mean Portfolio Asset Volatility	0.03107	**0.028437**	0.025632	**0.02343**
Portfolio Asset Volatility SD	0.00915	**0.006855**	0.004802	**0.003431**
Mean WMAP	0.496974	**0.535592**	0.5127	**0.537499**
WMAP SD	0.023691	**0.013268**	0.035945	**0.012434**
Mean Portfolio Entropy	4.6219	**4.9123**	3.8128	**4.9344**

TABLE 5.8: Comparison of PPO and DDPG across the key evaluation metrics.

The next stage is to conduct hypothesis tests on the results to determine whether the discrepancies in performance can be attributed to chance or if there is a statistically significant difference between the outcomes achieved by the models. This analysis will help clarify whether the observed differences in performance are due to the models strategies or merely the result of fortunate investments.

5.2.6 Statistical Analysis

DDPG has shown strong results, outperforming PPO in nearly every evaluation metric across both time periods. To determine the extent of DDPG's outperformance, a series of hypothesis tests will be conducted. These tests are divided into two sections: The first will examine the daily changes in Sharpe, Treynor, and Sortino ratios between the two methods over both periods. Since these metrics are normally distributed and contain independent observations, a t-test at a 5% significance level will be used. The second set of tests will analyse differences in portfolio volatility, WMAP, and entropy across both periods. As these metrics are not normally distributed and contain consecutively dependent observations, a Wilcoxon Signed-Rank Test will be employed, again at a 5% significance level [68].

T-Test		
Hypothesis	p-Value	Reject Hypothesis?
Hypothesis #1: Between 2006 and 2012, the average daily change in Sharpe Ratio was the same for PPO and DDPG.	1.55 x10^-6	**Yes**
Hypothesis #2: Between 2021 and 2024, the average daily change in Sharpe Ratio was the same for PPO and DDPG.	0.84537	**No**
Hypothesis #3: Between 2006 and 2012, the average daily change in Sortino Ratio was the same for PPO and DDPG.	0.0006141	**Yes**
Hypothesis #4: Between 2021 and 2024, the average daily change in Sortino Ratio was the same for PPO and DDPG.	0.96255	**No**
Hypothesis #5: Between 2006 and 2012, the average daily change in Treynor Ratio was the same for PPO and DDPG.	1.88244 x10^-6	**Yes**
Hypothesis #6: Between 2021 and 2024, the average daily change in Treynor Ratio was the same for PPO and DDPG.	0.760353	**No**

TABLE 5.9: t-test results for the comparisons between daily Sharpe, Treynor, and Sortino ratios.

Table 5.9 shows that there was a statistically significant difference between the daily increments in the Sharpe, Treynor, and Sortino ratios over the 2006-2012 period, highlighting DDPG's superior ability to capitalise on the opportunities in a growing market to leverage higher returns. However such a difference did not exist in the 2021-2024 period, indicating that both models performed similarly under the more volatile and unpredictable market conditions, suggesting that DDPG's ability to capitalise on such opportunities is much weaker in under these conditions This explains how PPO was able to achieve higher absolute returns through it's more risk-driven approach over this period.

Wilcoxon Signed-Rank Test		
Hypothesis	p-Value	Reject Hypothesis?
Hypothesis #7: PPO and DDPG maintained the same average portfolio volatility between 2006 and 2012.	≈ 0	**Yes**
Hypothesis #8: PPO and DDPG maintained the same average portfolio volatility between 2021 and 2024.	≈ 0	**Yes**
Hypothesis #9: PPO and DDPG had the same average WMAP between 2006 and 2012.	≈ 0	**Yes**
Hypothesis #10: PPO and DDPG had the same average WMAP between 2021 and 2024.	≈ 0	**Yes**
Hypothesis #11: PPO and DDPG had the same average portfolio entropy between 2006 and 2012.	≈ 0	**Yes**
Hypothesis #12: PPO and DDPG had the same average portfolio entropy between 2021 and 2024.	≈ 0	**Yes**

TABLE 5.10: Wilcoxon Signed-Rank Test results for the comparisons between volatility, WMAP, and entropy.

The results from the Wilcoxon Signed-Rank Test in Table 5.10 provide a much clearer consensus than the results in Table 5.9. These results show that DDPG, across both test periods, managed a portfolio that had a significantly:

- Lower average asset volatility
- Higher average WMAP
- Higher average entropy

These are all characteristics that are synonymous with a more effective trading strategy. Although the differences in risk-adjusted returns were not substantial in the second testing period, the asset selection strategies were significantly different. From the hypotheses presented in this section, it can be concluded that **DDPG's approach to trading is statistically superior to that of PPO.**

5.2.7 Final Model Portfolio Composition

The next step of model analysis is to examine the composition of the portfolio in relation to the asset features described in Section 3.5. These features are crucial, as they provide the information the model uses to make trading decisions. Therefore, it is essential to investigate the differences between the weighted average of these features in the portfolio over the testing period and the market average for the entire asset universe sample. This comparison will reveal how the model's decisions deviate from the broader market - providing insights into its asset selection strategy.

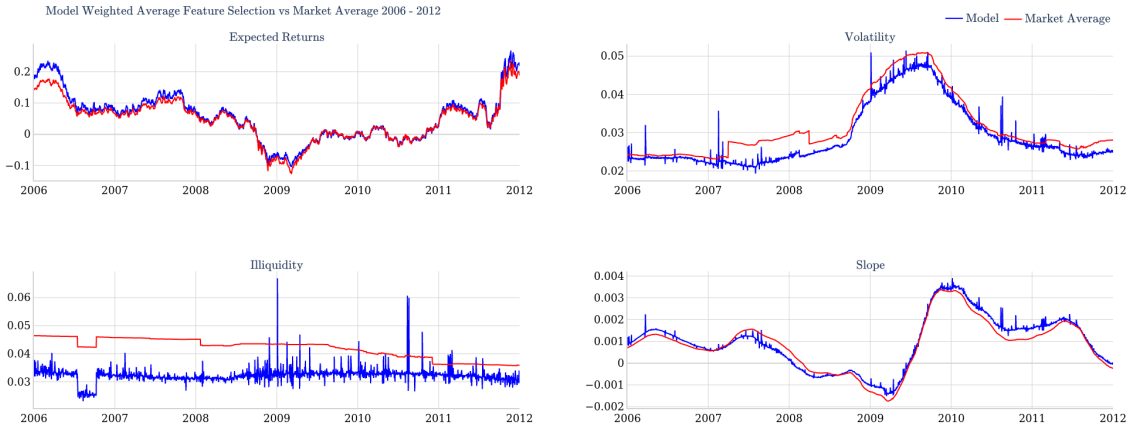


FIGURE 5.3: Weighted average of asset features in the DDPG portfolio relative to the market average from 2006 - 2012

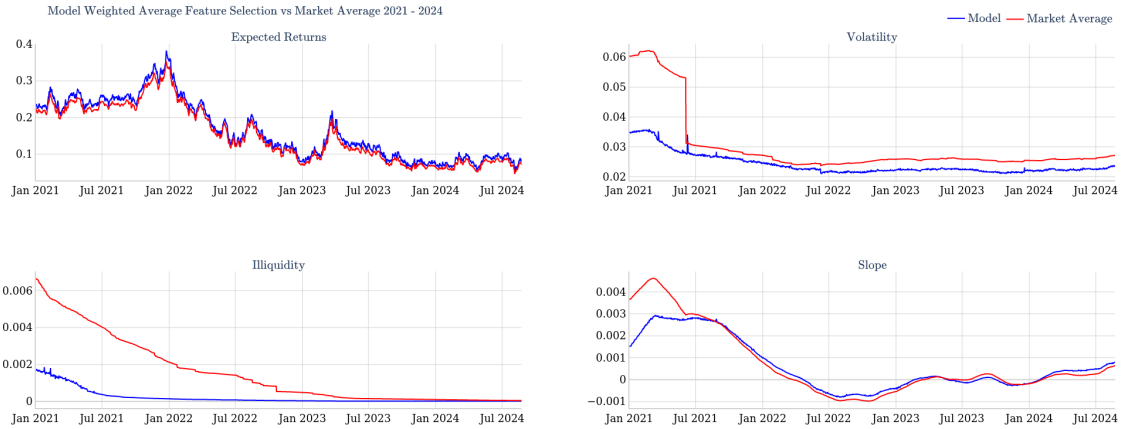


FIGURE 5.4: Weighted average of asset features in the DDPG portfolio relative to the market average from 2021 - 2024

The DDPG model followed a consistent asset selection strategy across the two testing periods. Specifically, it prioritised assets with higher expected returns as calculated by the CAPM, lower levels of volatility, and high liquidity. This approach is logical, as choosing assets with strong expected returns while reducing risk caused by asset volatility creates a stable foundation for the portfolio. Selecting assets with a high liquidity further protects the portfolio from the risks associated with sudden price jumps due to illiquidity, ensuring that the portfolio can be adjusted quickly in response to market changes. This strategy not only aligns with sound investment principles but also demonstrates the model's ability to adapt its asset selections based on key financial metrics, justifying their inclusion in the observation space and leading to a robust and diversified portfolio.

Unfortunately, the fourth feature, regression slope, showed no trend relative to the market average, indicating that the model does not factor in asset momentum. Consequently, this feature will be

either be adjusted or removed from future iterations of the model, as it currently only serves as noise without contributing to effective asset selection.

In addition to this, the fact that the DDPG model consistently applied the same asset selection strategy across both test periods, despite the wide array of market conditions, indicates a significant limitation in its adaptability. Although it selected different portfolios in each period, the adherence to the same overall strategy, even one that is fundamentally sound, suggests that the model lacks the ability to adapt dynamically to changing market environments. This indicates poor market generalisation, as the model does not adjust its strategy based on evolving market conditions or trends. This lack of adaptability could result in suboptimal performance in unexpected or rapidly changing market scenarios and it may also lead to missed key investment opportunities. This conslusion is further supported by the lack of variance in portfolio entropy discussed in Section 5.2.4.

5.3 Reward Function Sensitivity for Chosen Model

The results presented in Section 5.2 represent the optimal discovered configurations of both models with regards to maximising the average daily increase of Sharpe Ratio over both testing period. The optimal reward configuration for the final DDPG model was:

$$r_t = 0.7 \cdot \mathbb{E}(\mathrm{ShR}_t) + 0.1 \cdot \mathbb{E}(\mathrm{TR}_t) + 0.03 \cdot \mathbb{E}(\mathrm{SoR}_t) + 0.12 \cdot H_t + 0.05 \cdot \mathrm{ROI}_t \tag{5.1}$$

The primary objective of the model is to maximise the Sharpe Ratio, as this is the most general expression of risk-adjusted returns. Unsurprisingly, the model that achieved the highest cumulative Sharpe Ratio had the highest weighting on this factor. The other coefficients, such as those for TR, SoR, Entropy, and ROI, are included to help guide the model towards a well-rounded strategy. By considering different forms of risk, ensuring diversification through entropy, and acknowledging the potential for higher returns with higher risk, these additional metrics enable the model to balance various aspects of portfolio performance. This approach not only focuses on risk-adjusted returns but also fosters a comprehensive understanding of how to manage risk and maximise overall returns effectively.

The next section of the study looks at how variations in the reward function affect the model's portfolio management strategy. The first set of variations focuses on what happens when the dominant weighting is shifted to the other four measurements. This will show how the model adjusts its strategy when prioritising different aspects of risk, diversification, and absolute returns. The second set of variations look at what happens when the balance of weightings is removed entirely and the reward function is determined solely by a single measurement. In these variations, the reward function will be set to the expected Sharpe Ratio followed by the portfolio ROI. Hopefully this will reveal the limitations concentrating on a single performance indicator - justifying the compound reward structure used in this research.

Due to the significant time and computational resources required for each model run, only a finite number of runs can be performed, which prevents the investigating of coefficients as continuous variables. The proposed coefficient combinations can be seen in Table 5.11. As DDPG was proven to only develop a single strategy per model, the proposed model configurations will be evaluated over the 2021 - 2024 test period. The overall results from these evaluations across the key evaluation metrics are presented in Table 5.12. In addition to this, the complete set of plotted results from each of these reward functions can be found in Section A.1 of the Appendix.

	Coefficient				
Reward Function	E(ShR)	E(TR)	E(SoR)	Entropy (H)	ROI
Preferred Treynor	0.1	**0.7**	0.03	0.12	0.05
Preferred Sortino	0.03	0.1	**0.7**	0.12	0.05
Preferred Entropy	0.12	0.1	0.03	**0.7**	0.05
Preferred ROI	0.05	0.1	0.03	0.12	**0.7**
Only Sharpe	1.0	0.0	0.0	0.0	0.0
Only ROI	0.0	0.0	0.0	0.0	1.0
Final Model	0.7	0.1	0.03	0.12	0.05

TABLE 5.11: Summary of reward function weighting combinations for the sensitivity analysis

	Reward Function						
Metric	Preferred Treynor	Preferred Sortino	Preferred Entropy	Preferred ROI	Only Sharpe	Only ROI	Final Model
Overall ROI	0.1342	0.2007	0.1219	0.1551	0.1326	0.2310	0.3974
Cumulative Sharpe Ratio	64.2501	75.864259	74.1566	66.64928	63.5968	68.2453	99.71256
Cumulative Treynor Ratio	1.1728	1.314335	1.3009	1.2698	1.3129	1.1170	1.7662
Cumulative Sortino Ratio	39.9377	49.2804	42.1117	43.1087	51.9675	37.0151	60.4495
Mean Portfolio Asset Volatility	0.02985	0.03148	0.02873	0.03098	0.03118	0.02339	0.02343
Portfolio Asset Volatility SD	0.01138	0.014074	0.007337	0.0130	0.006309	0.00306	0.003431
Mean WMAP	0.4993	0.4988	0.4972	0.4972	0.4903	0.5081	0.5375
WMAP SD	0.003961	0.01295	0.00935	0.005047	0.05878	0.02977	0.01243
Mean Portfolio Entropy	5.5641	5.2064	5.4312	5.5270	3.1248	4.5250	4.9344

TABLE 5.12: Results of the different reward function weighting combinations

Preferred Treynor Out of the four alternative reward functions that maintained a compound reward function approach, this was the one that performed the worst overall, achieving a lower cumulative Sharpe Ratio than all the baseline methods. This is likely due to its primary objective being to minimise systematic risk, which only captures the sensitivity of the portfolio to broad market movements. During Test Period #2, market conditions were highly volatile, with significant idiosyncratic risks. As a result, despite its high portfolio entropy, the model's cumulative Sortino Ratio was low. This outcome suggests that while the model diversified across different assets, it did not adequately consider asset-specific risks based on historical returns, but rather focused on

systematic risk, looking at price variations in relation to the overall market. Consequently, it was exposed to significant negative returns due to the idiosyncratic risks that were prominent in that volatile market environment.

Preferred Sortino This was the most successful alternate reward function candidate overall. It consistently delivered strong risk-adjusted returns across all three ratios, along with a respectable overall ROI. Although this approach did not achieve the highest cumulative Sortino Ratio among the alternatives, its negative-return-oriented reward function guided its asset selection process to achieve a respectable mean WMAP with low variance. Additionally, it maintained high portfolio entropy, demonstrating a solid understanding of the benefits of diversification. Interestingly, it had the highest average portfolio volatility and volatility variance, which is unexpected for a model with a reward function focused on minimising negative returns. The fact that it still managed to maintain a respectable WMAP and strong risk-adjusted returns suggests that it leveraged this excess risk effectively to yield high returns.

Preferred Entropy Similar to the Preferred Sortino strategy, this reward function resulted in a model that managed a well-diversified portfolio capable of delivering good risk-adjusted returns. However, this model appears to have suffered from over-diversification, much like the Preferred Treynor method, though to a lesser extent. To reiterate, if a portfolio is too widely diversified without sufficient focus on its asset selection strategy, it will incur significant exposure to poorly performing assets. This is evidenced by the low ROI, the below-average Sortino Ratio, and the below-average mean WMAP. The model clearly invested capital in underperforming assets for the sake of diversification, leading to substantial negative exposure.

Preferred ROI he Preferred ROI strategy provides perhaps the most surprising results in this investigation. Despite having a reward structure that prioritises absolute returns, with less regard for risk management or diversification, it managed one of the most diversified portfolios among all the alternatives. This unexpected outcome may be due to the model recognising that concentrated high-risk portfolios are more vulnerable to market swings, which could adversely affect ROI. Additionally, the slight weightings for risk-adjusted returns and entropy likely guided the model towards maintaining a more balanced portfolio. For the most part, this led to behavior similar to the UCRP baseline, which had a very simple asset selection technique - not factoring in risk considerations. This similarity explains why the cumulative risk-adjusted ratios are on the lower side.

Only Sharpe This was certainly the weakest alternative out of the considered reward functions, obtaining the lowest overall ROI and cumulative Sharpe Ratio. In addition to this, it had by far the lowest level of portfolio entropy, leading to a highly changeable WMAP. At first glance, these results may seem contradictory: how does high volatility and a low overall ROI result in a high Sortino Ratio? To understand this, the progression of the ROI, as shown in Table A.5, needs to be studied. The high volatility of the assets led to a sharp rise in portfolio value at the start of the period, with minimal negative returns, which significantly boosted the Sortino Ratio. After this initial spike, the ROI stabilised at around 0.5. Following this, the portfolio value rapidly declined to -0.05 over the next six months before recovering to 0.13 by the end of the period. This early sharp rise allowed the model to maintain an above-average Sortino Ratio for the remainder of the period, despite its overall poor and volatile performance - similar to the FtW strategy in this period.

Only ROI Given the conclusions reached with the Preferred ROI strategy, the behaviour of this model can be easily explained. Without the additional incentives for risk management and diversification, the model opted for a much more concentrated portfolio. This portfolio included a mix of consistently high-performing and low-performing assets, effectively embracing both aspects of the FtW and FtL strategies. This explains the high WMAP variance and the low average asset volatility and volatility variance. Additionally, the presence of consistently poorly performing assets in the portfolio led to significant downside exposure, resulting in a very low cumulative Sortino Ratio. The polarised nature of the assets in this portfolio also accounts for its ability to achieve the highest ROI, despite having the lowest Sortino Ratio.

5.4 Evaluation of the Integrated Financial Analysis Approach

This research introduced a novel approach by integrating financial analysis features into the observation space, moving beyond the traditional reliance on raw financial data alone. This method aimed to reduce noise and enable the consideration of more comprehensive historical information in decision-making processes. To assess the efficacy of this method, let's look back to the examples reviewed in Section 2.4.

Firstly, let us consider the work done by Hieu in comparing PPO, DDPG and GDPG in the field of stock-based optimisation [52]. Like this research, Hieu concludes that DDPG is the most suitable method among the given candidates, yielding the highest returns. Hieu's approach involves using a discrete wavelet transform to de-noise the historical financial data, with each asset observation comprising three historical pricing series: the close price; high price; and the close price after wavelet transform. However, while this study demonstrates that DDPG can generate returns greater than the UCRP, FtW, and FtL strategies, Hieu's asset universe is limited to only 50 assets, which represents just 10% of the asset universe used in this study. This limitation raises concerns about the computational feasibility of Hieu's approach when applied to more comprehensive domains. Furthermore, Hieu's methodology involves manually selecting the initial portfolio, which consists of six assets that the RL model is expected to handle from there on. Initial iterations with these selected assets led to poor results due to suboptimal asset selection, with the model only generating returns after being provided with an initial portfolio of high-performing stocks from diverse industries. This suggests that the model struggled to capture relevant patterns from the market data and failed to grasp any core investment strategies such as diversification and risk management. Consequently, this reliance on manual selection highlights a clear shortcoming in the model's ability to autonomously develop effective investment strategies based solely on historical market data. In contrast to this, the method presented in this study demonstrates a robust ability to handle a larger and more diverse asset universe without requiring manual intervention. Unlike Hieu's approach, the model in this study is always initialised with a random set of asset weightings, in both training and testing, spread across the entire asset universe, rather than being limited to a small, manually selected subset of assets. This shows that the model is truly capable of identifying effective patterns and portfolio management strategies based off the financial metrics it is being supplied through the observation space by it's ability to steer the portfolio from a randomised initial state to a state where it is generating stable, risk-adjusted returns. Furthermore, this study's model operates on 500 assets which is ten times the size of Hieu's asset universe, showcasing its superior scalability and sample efficiency.

Now let's consider the second work reviewed in Section 2.4 that covers a comparison of PPO and DDPG in this field [50]. In this research Liang *et al* apply DDPG and PPO to the Chinese Stock Market to optimise a portfolio of assets. They experiment over a different combinations of historical, opening, closing, high and low pricing time series with little to no pre-processing, something they identity as a key area for improvement in their future work section, in an attempt to determine which feature combination leads to the best performance. During the selection of their asset universe they ensure that each asset has at least 1200 days of historical data, implying that each asset observation will contain at least one long financial time series - potentially more for feature combinations. As a result of this complexity, the RL models are only tested in asset universes comprised of 5 assets - 1% of the size of the portfolio sized used in this investigation. By performing numerous training and testing iterations - they did prove that this technique could be used to consistently generate strong absolute and risk-adjusted returns relative to the UCRP, FtW and FtL strategies in these micro asset universes, the fact that these micro asset universes are so small shows that lack of scalability of this method to larger state spaces - something that has been significantly improved upon in the novel approach adopted by this research.

Through these comparisons to similar published research also looking to apply PPO and DDPG to to portfolio optimisation, it is clear that the novel approach to the pre-processing of historical data and the presentation of the asset observations has had a positive impact on this research. This can be proven by the fact that this research achieved similar levels of performance relative the UCRP, FtW, and FtL strategies as the other studies - but it managed to do so while operating on a **significantly** larger asset universe. Furthermore, the fact that clear asset selection strategies were identified in Section 5.2.7, shows that this more refined and compact representation of assets financial history has helped the model to develop theoretically sound trading strategies.

6. Conclusions and Further Work

6.1 Evaluation of Project Deliverables

In this section, the project deliverables that were outlined in Section 1.2.3 will be revisited to assess not only how, but also the extent to which they were all achieved

Deliverable #1: Market simulation framework. This was the most fundamental deliverable, the market simulation environment was implemented using an object-orientated architecture as seen in Chapter 4. This design significantly enhanced the maintainability and scalability of the simulation environment by making testing straightforward due to its modular nature. It also made it easier to identify and optimise bottlenecks using flame graphs. An example of this optimisation can be seen in Section A.2 of the Appendix. This architecture was combined with the with the OpenAI gymnasium environment interface to create a flow of time through a simulated historical market by advancing the simulation by one time step with each call of `env.step()`

Deliverable #2: Reinforcement leaning model. To achieve this deliverable, the PPO and DDPG models were implemented using the Stable-Baselines3 (SB3) library. These implementations were able to interface directly with the simulated market environments described in Deliverable #1, taking actions and receiving feedback in the form of state updates and rewards, thereby facilitating effective training and evaluation of their performance. The model parameters were fine-tuned to ensure a stable learning process, and the reward function was iteratively developed to maximise cumulative risk-adjusted returns.

Deliverable #3: Functionality to evaluate model performance relative to baseline strategies To complete this, a range of fixed-heuristic baseline strategies had to be created. These were the uniform buy-and-hold (UBAH), the uniform constantly rebalanced portfolio (UCRP), the follow-the-winner (FtW) and follow-the-loser (FtL) strategies. Each of these techniques follows a unique approach, demonstrating varying levels of success across different market environments. Subsequently, a test suite was designed, enabling both the baseline portfolio management strategies and the model's portfolio management strategy to interact with a market environment as described in Deliverable #1 over a a fixed test period of unseen market data. This setup allowed for a direct comparison between the baseline methods and the RL model - allowing for a clear evaluation of their relative performance.

Deliverable #4: Analysis of learned strategies. This analysis was carried out in Section 5.2.7. As mentioned at the start of Section 3.5, the novelty in this approach lies in the pre-processing of the financial data to create to create more tangible set of features, as opposed to previous pieces of research in this area which just use the raw historical pricing data. This approach was chosen as it provides a much more effective and concise viewpoint of an asset's economic status with a lot less noise by summarising years worth of data into a series of single-value observations. An additional benefit of this is that it removes the 'blackbox' element that is synonymous with deep-learning techniques. By looking at the features of the assets the model had put into the portfolio, a clear strategy could be identified - with the final DDPG model choosing to maintain a portfolio of low-volatility, high-liquidity assets with above average expected returns.

Deliverable #5: Sensitivity analysis of a model's reward function. While an analysis was conducted to compare various reward functions in Section 5.3, it does not do so in the most effective manner. Ideally, a sensitivity analysis treats the variables as continuous, progressively altering them to model the change in output in relation to changes in the input. However, due to the complexity of the training process in this study, only a limited number of models could be run, restricting the scope and depth of potential findings from this investigation. Nevertheless, the results from this investigation did reveal the profound effects the composition of the reward function has on not only the quality of the returns but also the selected portfolio management strategy.

6.2 Evaluating Project Objectives Criteria

The final stage of the project evaluation is to look back to the initial project objectives in Table 1.1 to determine which have had their respective success criteria met.

6.2.1 Primary Objective Evaluation

Primary Objective: Assessing RL Model Efficacy

Success criteria: "Successful if the RL model achieves higher absolute returns and risk-adjusted returns than all the benchmark strategies"

Evaluation: The final DDPG model outperformed PPO and all baseline models consistently with respect to risk adjusted returns across both test periods, showing it's applicability over a wide range of market environments. The one exception was the FtL method having a higher cumulative Sortino Ratio over the 2006-2012 period. This suggests that further work may be required to refine the model's downside risk management capabilities during adverse market conditions. These results justify the use of reinforcement learning for portfolio management, demonstrating its capacity to outperform traditional heuristic strategies in most scenarios. The success of the final DDPG model in achieving higher absolute and risk-adjusted returns as well as demonstrating better risk management capabilities than the baseline strategies solidifies the case for further exploration and application of RL techniques in optimising investment strategies.

6.2.2 Secondary Objectives Evaluation

Secondary Objective #1: Strategic Behaviour Identification

Success criteria: "Successful if the model identifies and implements a range of strategic behaviours, including established techniques and/or novel strategies, as evidenced by improved performance metrics and adaptability to a range of simulated market scenarios."

Evaluation: The question of whether or not this objective has been satisfied is less clear than the Primary Objective. The final model did display a range of impressive behaviours, synonymous with good portfolio management. Namely, it maintained a portfolio with high levels of diversification, centred around assets with high liquidity and expected returns as well as low volatility. Furthermore, where applicable it utilised increased exposure to risk to leverage higher returns. This certainly qualifies as a good range of strategic behaviours. However, it failed to meet the last point of the success criteria. The model ultimately showed little adaptability to different market environments. While this lack of adaptability did not have an adverse effect on the results due to the sound nature of the portfolio's primary investment strategy, this shortcoming could lead to missing emerging opportunities or failing to adjust to adverse conditions, potentially leading to suboptimal performances. While the model demonstrated the ability to identify suitable strategies in line with a given reward structure, it showed limited adaptability to deviate from its chosen strategy. This suggests that while the model can align with predefined goals, it lacks the flexibility to adjust dynamically to changing market conditions. For these reasons, this objective can only be considered partially achieved.

Secondary Objective #2: Response to Extreme Market Events

Success criteria: "Successful if the model maintains stability and manages risk effectively. It must avoid significant loses - greater than the baseline strategies."

Evaluation: The final DDPG model displayed exemplary risk management by generally selecting assets with low volatility and high liquidity. The most extreme market event encountered during the testing phase was the 2008 financial crisis, which spanned from mid-2008 to early 2009. During this period, the net ROI of the DDPG portfolio fell from 0.31 to -0.21. Although this decline is not ideal, it represents the smallest reduction in portfolio value among all the considered methods,

demonstrating the effectiveness of DDPG's investment strategy in reducing losses during significant market downturns. For these reasons, this objective can be considered successfully achieved.

Secondary Objective #3: Impact of Reward Structures

Success criteria: "Successful if variations in reward parameters lead to discernible changes in the model's investment behaviours. These changes should align with the specific financial goals behind that the given reward structure"

Evaluation: Section 5.3 examined six distinct alternative reward function configurations and conducted a results-driven investigation to determine why specific changes in the reward function led the model to adopt strategies that produced the observed outcomes. While this analysis did yield a few surprising results, the variations in reward parameters did lead to clear and identifiable changes in investment behaviours, demonstrating the model's responsiveness to different reward incentives. These behavioural changes aligned with the financial objectives associated with each reward structure, such as attempting to maximise returns, minimising risk, or aiming for portfolio diversification. Therefore, this objective can be considered successfully achieved, as the model's strategies adapted logically in response to the changes in reward structures.

6.3 Future Work

This study has provided a promising proof of concept that reinforcement learning can be successfully applied on a large scale in the field of portfolio management. However, several areas can be further explored to enhance the robustness, effectiveness, and adaptability of these models. This section outlines potential directions for future work that could refine this research, enabling future iterations of this technique to develop more sophisticated and adaptive investment strategies.

Currently, the only component of the training process that leverages the advanced processing power of a GPU is the internal training routines of the Stable-Baselines3 (SB3) algorithms. These algorithms have built-in optimisations for GPU acceleration, including support for CUDA instructions [65], which enable efficient parallel processing and significantly speed up computations related to neural network training and gradient descent. This means a very large bottle neck exists in the training process in the form of the object orientated architecture demonstrated in Chapter 4 used to evaluate asset features. This significantly hinders scalability of this research. To mitigate this, the current codebase needs to be refactored using Python libraries that support GPU computations. This would involve the conversion of all numerical calculations to be tensor-based operations, though libraries such as PyTorch [64]. This approach would allow for end-to-end acceleration of the entire training pipeline, including data pre-processing, feature extraction, and the step process, enabling the model to handle larger datasets and more complex simulations more efficiently. This would allow for a much more extensive training process, hopefully allowing for the identification of deeper economic patterns therefore increasing model adaptability. Furthermore, it could also enable the expansion of the observation space to include a more comprehensive suite of features.

Upon seeing the final results and the poor level of generalisation and adaptability displayed by the final model, it was clear that changes need to be made to training process to correct this. The first change that would be made in future versions of this project would be diversify the observation space. Currently during the training process the composition of the observation space is essentially constant - with each row representing the state of specific asset. This creates a nasty local minimum in the training process. Rather than necessarily utilising the individual feature values to drive financial decisions, it could simply learn which assets (which rows of the observation space) generally perform better preventing any deeper patterns from being recognised. This would explain the lack of adaptability displayed by the final model. To address this one of two things need to happen at the end of a training episode. Either re-order the asset universe, so these no geometric relationships can be created, or completely change the assets within the asset universe all together (this can only work if the current asset universe is a sample of the market). This batch-based strategy would force the model to look for deeper patterns found in the values of the features themselves, hopefully increasing its market generalisation abilities.

While an extensive investigation was conducted into reward function parameter optimisation in Section 5.3 due to the major effect it had in the output, there are two other groupings of hyperparameters that need to optimised in future developments of this research. These groups were not addressed in this study due to time constraints and the lengthy process required to train models. Optimising these additional hyperparameters will be essential for further improving the model's performance and robustness. The first group is the DDPG model parameters. These parameters were adjusted during the initial stages of this project to achieve a configuration that allowed for stable learning. For instance, if model's learning rate was too high, the actor-critic loss functions would produce values that were too large meaining the network parameters would change excessively with each step resulting in a highly unstable learning process. These parameters have a major effect on the training process, therefore a full sensitivity analysis - similar to the one in Section 5.3 needs to be conducted on them. The second group of hyperparameters is the look back periods on the respective feature calculations. These hyperparameters were alluded to in Section 3.5. Each feature considers a segment of historical financial data prior to the calculation date. The length of this period determines the longevity of the observation window and could, therefore, have a significant impact on the model's strategy selection. Adjusting the look-back periods may influence how the model balances short-term versus long-term rewards, ultimately shaping its investment decisions. The extent to which these parameters can affect model behaviour needs to be determined based on empirical evidence, as they may significantly enhance the model's responsiveness to varying market conditions and investment opportunities.

Currently, the model is required to invest all capital into the markets, which can be suboptimal during periods of significant market turmoil, such as the 2008 financial crisis, when profitable strategies are difficult to find. To address this, future models should incorporate an expanded action space that allows for the option to hold a portion of funds an a simulated interest-bearing bank account or other non-invested forms. This would create a more realistic trading environment and provide the flexibility to limit losses during adverse market conditions, ultimately enhancing the potential for higher overall returns.

6.4 Conclusion

This study set out to explore the application of reinforcement learning (RL) in financial portfolio management, aiming to develop strategies that can outperform traditional heuristic-based approaches. By leveraging RL models the research aimed to maximise risk-adjusted returns while effectively managing various forms of risk, including systematic, idiosyncratic, and downside risks.

The methodology involved the development of a novel data pre-processing and feature presentation technique using financial and statistical analysis. A range of fixed-heuristic baseline strategies were also implemented, including the uniform buy-and-hold (UBAH), uniform constantly rebalanced portfolio (UCRP), follow-the-winner (FtW), and follow-the-loser (FtL), to serve as benchmarks for evaluating the RL models' performance.

The results demonstrated that the final DDPG model developed sound investment principles to manage risk and deliver the most consistent risk-adjusted returns compared to PPO and the baseline strategies. These principles allowed the DDPG model to effectively manage risk by selecting low-volatility, high-liquidity assets and maintaining higher portfolio entropy, which facilitated diversification. However the study also identified limitations, such as the model's lack of adaptability to changing market conditions, which could be addressed in future research. The novel approach to feature presentation enabled the model to operate effectively over a much larger asset universe than the current state of the art.

In conclusion, this research has successfully demonstrated that reinforcement learning can be a viable and effective approach to portfolio management, capable of outperforming traditional strategies under various market conditions. The findings provide a strong proof of concept, highlighting the potential of RL models to develop sophisticated and effective investment strategies.

Bibliography

[1] Jonathan Law and John Smullen. *Amsterdam Stock Exchange*. 2008. DOI: https://doi.org/10.1093/acref/9780199229741.013.0124.

[2] *The history of LSEG*. URL: https://www.lseg.com/en/about-us/history.

[3] White. "Economic prediction using neural networks: the case of IBM daily stock returns". In: *IEEE 1988 International Conference on Neural Networks*. 1988, 451–458 vol.2. DOI: 10.1109/ICNN.1988.23959.

[4] J. Moody and M. Saffell. "Learning to trade via direct reinforcement". In: *IEEE Transactions on Neural Networks* 12.4 (2001), pp. 875–889. DOI: https://doi.org/10.1109/72.935097.

[5] Bojana Novićević Čečević, Ljilja Antić, and Adrijana Jevtić. "Stock Price Prediction of the Largest Automotive Competitors Based on the Monte Carlo Method". In: *Economic Themes* 61.3 (2023), pp. 419–441. DOI: https://doi.org/10.2478/ethemes-2023-0022.

[6] L. Bachelier. "Théorie de la spéculation". fr. In: *Annales scientifiques de l'École Normale Supérieure* 3e série, 17 (1900), pp. 21–86. DOI: 10.24033/asens.476. URL: http://www.numdam.org/articles/10.24033/asens.476/.

[7] Harry Markowitz. "Portfolio Selection". In: *The Journal of Finance* 7.1 (1952), pp. 77–91. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/2975974 (visited on 08/14/2024).

[8] Edwin J Elton and Martin J Gruber. "Modern portfolio theory, 1950 to date". In: *Journal of Banking Finance* 21.11 (1997), pp. 1743–1759. ISSN: 0378-4266. DOI: https://doi.org/10.1016/S0378-4266(97)00048-4.

[9] Taariq G.H. Surtee and Imhotep Paul Alagidede. "A novel approach to using modern portfolio theory". In: *Borsa Istanbul Review* 23.3 (2023), pp. 527–540. ISSN: 2214-8450. DOI: https://doi.org/10.1016/j.bir.2022.12.005.

[10] William F. Sharpe. "Capital Asset Prices: A Theory of Market Equilibrium Under Consitions of Risk". In: *The Journal of Finance* 19.3 (1964), pp. 425–442. DOI: https://doi.org/10.1111/j.1540-6261.1964.tb02865.x.

[11] John Lintner. "The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets". In: *The Review of Economics and Statistics* 47.1 (1965), pp. 13–37. ISSN: 00346535, 15309142. URL: http://www.jstor.org/stable/1924119 (visited on 08/15/2024).

[12] Fischer Black. "Capital Market Equilibrium with Restricted Borrowing". In: *The Journal of Business* 45.3 (1972), pp. 444–455. ISSN: 00219398, 15375374. URL: http://www.jstor.org/stable/2351499 (visited on 08/15/2024).

[13] Haim Levy. "The Capital Asset Pricing Model". In: *The Capital Asset Pricing Model in the 21st Century: Analytical, Empirical, and Behavioral Perspectives*. Cambridge University Press, 2011, pp. 117–155.

[14] Eugene F. Fama and Kenneth R. French. "The Cross-Section of Expected Stock Returns". In: *The Journal of Finance* 47.2 (1992), pp. 427–465. DOI: https://doi.org/10.1111/j.1540-6261.1992.tb04398.x.

[15] Keith S.K Lam. "The relationship between size, book-to-market equity ratio, earnings–price ratio, and return for the Hong Kong stock market". In: *Global Finance Journal* 13.2 (2002), pp. 163–179. ISSN: 1044-0283. DOI: https://doi.org/10.1016/S1044-0283(02)00049-2.

[16] Dennis Stattman. "Book values and stock returns". In: *The Chicago MBA: A journal of selected papers* 4.1 (1980), pp. 25–45.

[17] Louis K. C. Chan, Yasushi Hamao, and Josef Lakonishok. "Fundamentals and Stock Returns in Japan". In: *The Journal of Finance* 46.5 (1991), pp. 1739–1764. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/2328571 (visited on 08/15/2024).

[18] Rolf W. Banz. "The relationship between return and market value of common stocks". In: *Journal of Financial Economics* 9.1 (1981), pp. 3–18. ISSN: 0304-405X. DOI: https://doi.org/10.1016/0304-405X(81)90018-0.

[19] Laxmi Chand Bhandari. "Debt/equity ratio and expected common stock returns: Empirical evidence". In: *The journal of finance* 43.2 (1988), pp. 507–528.

[20] Mahnoor Sattar. "CAPM Vs Fama-French Three-Factor Model: An Evaluation of Effectiveness in Explaining Excess Return in Dhaka Stock Exchange". In: *International Journal of Biometrics* 12 (2017), p. 119. DOI: https://doi.org/10.5539/ijbm.v12n5p119.

[21] Werner F. M. De Bondt and Richard Thaler. "Does the Stock Market Overreact?" In: *The Journal of Finance* 40.3 (1985), pp. 793–805. ISSN: 00221082, 15406261. DOI: https://doi-org.bris.idm.oclc.org/10.2307/2327804. URL: http://www.jstor.org/stable/2327804 (visited on 08/16/2024).

[22] Narasimhan Jegadeesh and Sheridan Titman. "Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency". In: *The Journal of Finance* 48.1 (1993), pp. 65–91. ISSN: 00221082, 15406261. DOI: https://doi-org.bris.idm.oclc.org/10.2307/2328882. URL: http://www.jstor.org/stable/2328882 (visited on 08/16/2024).

[23] Mark M. Carhart. "On Persistence in Mutual Fund Performance". In: *The Journal of Finance* 52.1 (1997), pp. 57–82. DOI: https://doi.org/10.1111/j.1540-6261.1997.tb03808.x.

[24] Kent Daniel and Tobias J. Moskowitz. "Momentum crashes". In: *Journal of Financial Economics* 122.2 (2016), pp. 221–247. ISSN: 0304-405X. DOI: https://doi.org/10.1016/j.jfineco.2015.12.002.

[25] Michael J. Cooper, Roberto C. Gutierrez Jr., and Allaudeen Hameed. "Market States and Momentum". In: *The Journal of Finance* 59.3 (2004), pp. 1345–1365. DOI: https://doi.org/10.1111/j.1540-6261.2004.00665.x.

[26] David A. Lesmond, Michael J. Schill, and Chunsheng Zhou. "The illusory nature of momentum profits". In: *Journal of Financial Economics* 71.2 (2004), pp. 349–380. ISSN: 0304-405X. DOI: https://doi.org/10.1016/S0304-405X(03)00206-X.

[27] William F. Sharpe. "Mutual Fund Performance". In: *The Journal of Business* 39.1 (1966), pp. 119–138. ISSN: 00219398, 15375374. URL: http://www.jstor.org/stable/2351741 (visited on 08/16/2024).

[28] Craig Israelsen. "A Refinement to the Sharpe Ratio and Information Ratio". In: *Journal of Asset Management* 5 (Apr. 2005), pp. 423–427. DOI: https://doi.org/10.1057/palgrave.jam.2240158.

[29] Jack L Treynor. "How to Rate Management of Investment Funds." In: *Harvard business review.* 43.1 (1965). ISSN: 0017-8012.

[30] Richard Roll. "Ambiguity when Performance is Measured by the Securities Market Line". In: *The Journal of Finance* 33.4 (1978), pp. 1051–1069. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/2326939 (visited on 08/16/2024).

[31] Frank A. Sortino and Lee N. Price. "Performance Measurement in a Downside Risk Framework". In: *The Journal of Investing* 3 (3 1994), pp. 59–64. DOI: https://doi.org/10.3905/joi.3.3.59.

[32] C. E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

[33] Angelos Filos. *Reinforcement Learning for Portfolio Management*. 2019. DOI: https://doi.org/10.48550/arXiv.1909.09571.

[34] Richard S Sutton, Andrew G Barto, et al. "Reinforcement learning". In: *Journal of Cognitive Neuroscience* 11.1 (1999), pp. 126–134.

[35] Martin L. Puterman. "Chapter 8 Markov decision processes". In: *Stochastic Models*. Vol. 2. Handbooks in Operations Research and Management Science. Elsevier, 1990, pp. 331–434. DOI: https://doi.org/10.1016/S0927-0507(05)80172-0.

[36] Yuxi Li. "Deep Reinforcement Learning: An Overview". In: (2018). DOI: https://doi.org/10.48550/arXiv.1701.07274.

[37] John Schulman et al. *Proximal Policy Optimization Algorithms*. 2017. DOI: https://doi.org/10.48550/arXiv.1707.06347.

[38] Peter Henderson et al. "Deep Reinforcement Learning That Matters". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32 (Sept. 2017). DOI: https://doi.org/10.1609/aaai.v32i1.11694.

[39] Christopher Watkins. "Learning From Delayed Rewards". In: (Jan. 1989).

[40] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (Feb. 2015), pp. 529–533. ISSN: 1476-4687. DOI: https://doi.org/10.1038/nature14236.

[41] Neda Yousefi. "Deep Reinforcement Learning for Tehran Stock Trading". In: *Journal of Novel Engineering Science and Technology* Vol. 01 (Nov. 2022), pp. 37–42. DOI: https://doi.org/10.56741/jnest.v1i02.171.

[42] Timothy P. Lillicrap et al. *Continuous control with deep reinforcement learning*. 2015. DOI: https://doi.org/10.48550/arXiv.1509.02971.

[43] Peter Henderson et al. "Deep Reinforcement Learning That Matters". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). DOI: https://doi.org/10.1609/aaai.v32i1.11694.

[44] Scott Fujimoto, Herke van Hoof, and David Meger. *Addressing Function Approximation Error in Actor-Critic Methods*. 2018. DOI: https://doi.org/10.48550/arXiv.1802.09477.

[45] David Silver et al. *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*. 2017. DOI: https://doi.org/10.48550/arXiv.1712.01815.

[46] Marc Peter Deisenroth and Carl Edward Rasmussen. "PILCO: a model-based and data-efficient approach to policy search". In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Bellevue, Washington, USA: Omnipress, 2011, pp. 465–472.

[47] Carl Edward Rasmussen. "Gaussian Processes in Machine Learning". In: *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*. Ed. by Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 63–71. ISBN: 978-3-540-28650-9. DOI: https://doi.org/10.1007/978-3-540-28650-9_4.

[48] Andreas Damianou and Neil D. Lawrence. "Deep Gaussian Processes". In: *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Carlos M. Carvalho and Pradeep Ravikumar. Vol. 31. Proceedings of Machine Learning Research. Scottsdale, Arizona, USA: PMLR, 29 Apr–01 May 2013, pp. 207–215. URL: https://proceedings.mlr.press/v31/damianou13a.html.

[49] Niranjan Srinivas et al. "Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting". In: *IEEE Transactions on Information Theory* 58.5 (2012), pp. 3250–3265. DOI: https://doi.org/10.1109/TIT.2011.2182033.

[50] Zhipeng Liang et al. *Adversarial Deep Reinforcement Learning in Portfolio Management*. 2018. DOI: https://doi.org/10.48550/arXiv.1808.09940.

[51] Mu-En Wu et al. "Portfolio management system in equity market neutral using reinforcement learning". In: *Applied Intelligence* 51.11 (Nov. 2021), pp. 8119–8131. ISSN: 1573-7497. DOI: https://doi.org/10.1007/s10489-021-02262-0.

[52] Le Trung Hieu. "Deep Reinforcement Learning for Stock Portfolio Optimization". In: *International Journal of Modeling and Optimization* 10.5 (Oct. 2020), pp. 139–144. ISSN: 2010-3697. DOI: https://doi.org/10.7763/ijmo.2020.v10.761.

[53] Paul Wilmott. *Paul Wilmott introduces quantitative finance.* John Wiley Sons, 2007.

[54] James Chen. "What Is a Liquid Asset, and What Are Some Examples?" In: (2024). URL: https://www.investopedia.com/terms/l/liquidasset.asp.

[55] Adam Hayes. "Slippage: What It Means in Finance, With Examples". In: (2023). URL: https://www.investopedia.com/terms/s/slippage.asp.

[56] Bin Li and Steven C. H. Hoi. *Online Portfolio Selection: A Survey*. 2013. DOI: https://doi.org/10.48550/arXiv.1212.2129.

[57] Ben Taylor. *The Costs of Investing*. 2024. URL: https://www.investopedia.com/investing/costs-investing/.

[58] Elite Data Science. *Overfitting in Machine Learning: What It Is and How to Prevent It*. 2022. URL: https://elitedatascience.com/overfitting-in-machine-learning#signal-vs-noise.

[59] *NASDAQ Composite Index*. URL: https://www.nasdaq.com/market-activity/index/comp.

[60] Hongnan Gao. "Linear Regression from Scratch using Python and its Time Complexity". In: (2021). URL: https://bit.ly/3yYaSc7.

[61] Yakov Amihud. "Illiquidity and stock returns: cross-section and time-series effects". In: *Journal of Financial Markets* 5.1 (2002), pp. 31–56. ISSN: 1386-4181. DOI: https://doi.org/10.1016/S1386-4181(01)00024-6.

[62] *Stable-Baselines3 PPO Implementation*. URL: https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html.

[63] *Stable-Baselines3 DDPG Implementation*. URL: https://stable-baselines3.readthedocs.io/en/master/modules/ddpg.html.

[64] *PyTorch*. URL: https://pytorch.org/.

[65] *CUDA Python*. URL: https://developer.nvidia.com/cuda-python.

[66] *Gymnasium Documentation*. URL: https://gymnasium.farama.org/index.html.

[67] *Google Colaboratory*. URL: https://colab.google/.

[68] Frank Wilcoxon. "Individual Comparisons by Ranking Methods". In: *Biometrics Bulletin* 1.6 (1945), pp. 80–83. ISSN: 00994987. URL: http://www.jstor.org/stable/3001968.

A. Appendix

A.1 Detailed Results From Reward Function Variants

A.1.1 Preferred Treynor

TABLE A.1: Preferred Treynor reward function results across key evaluation metrics

A.1.2 Preferred Sortino

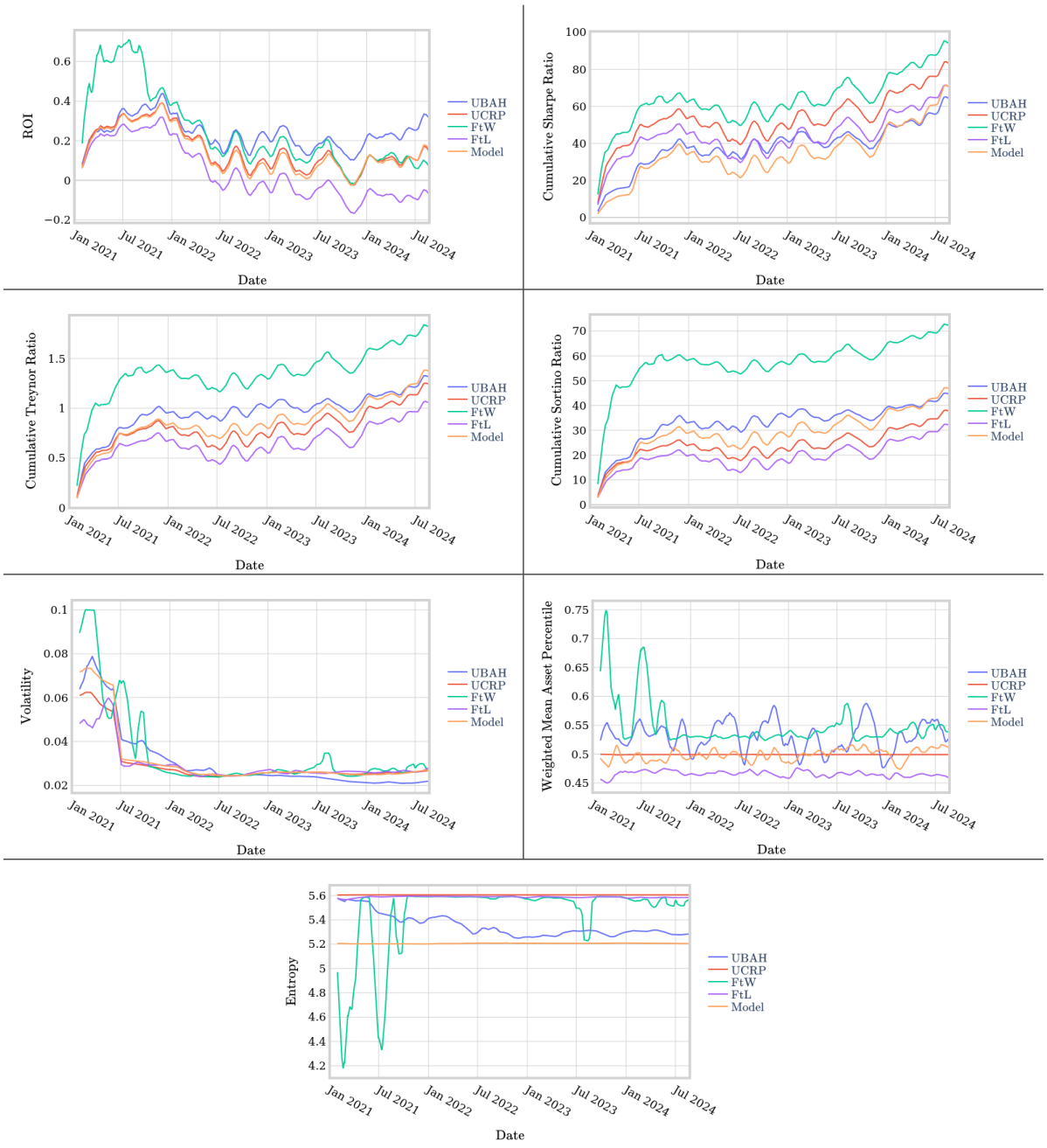


TABLE A.2: Preferred Sortino reward function results across key evaluation metrics

A.1.3 Preferred Entropy

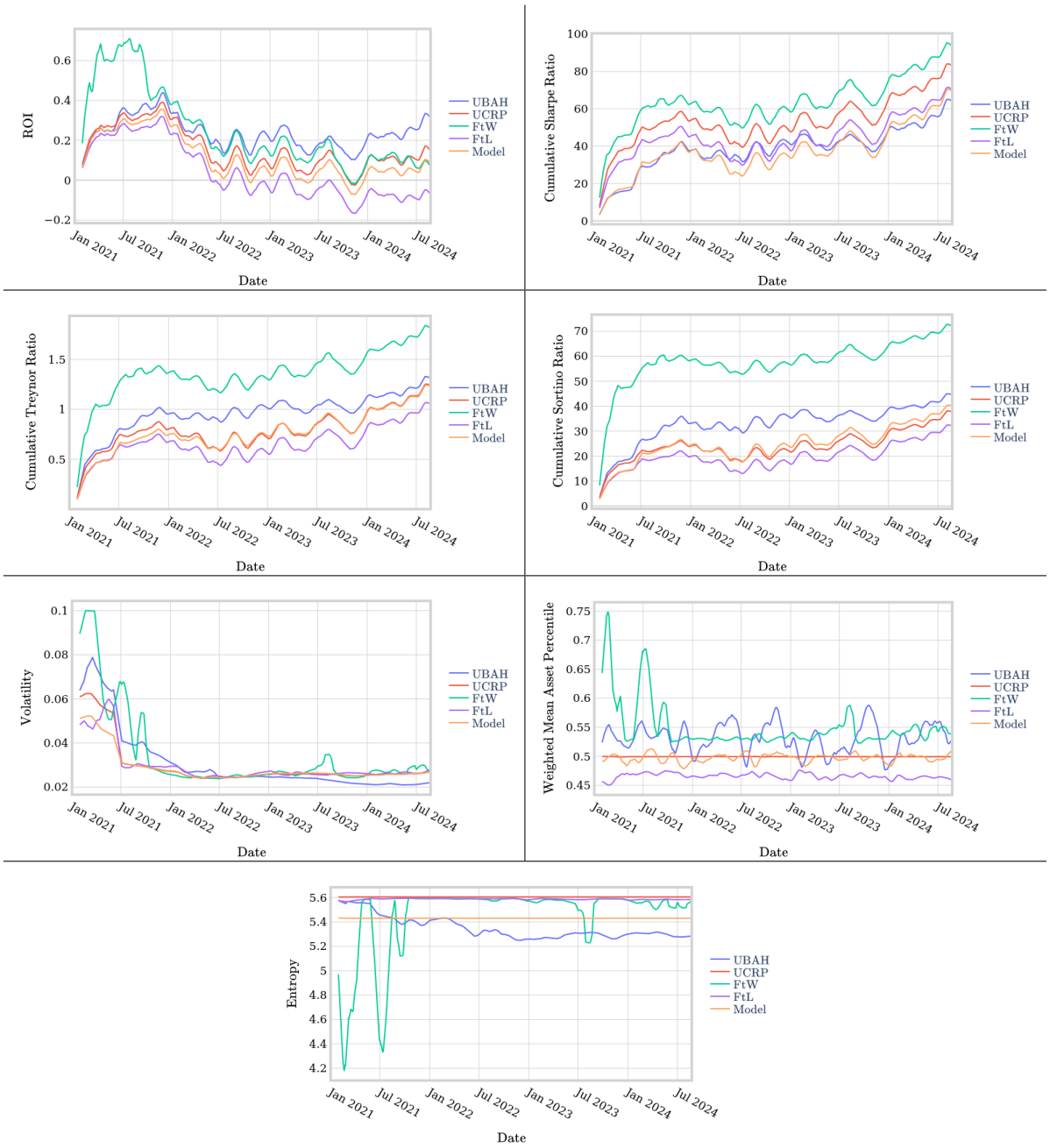


TABLE A.3: Preferred Entropy reward function results across key evaluation metrics

A.1.4 Preferred ROI

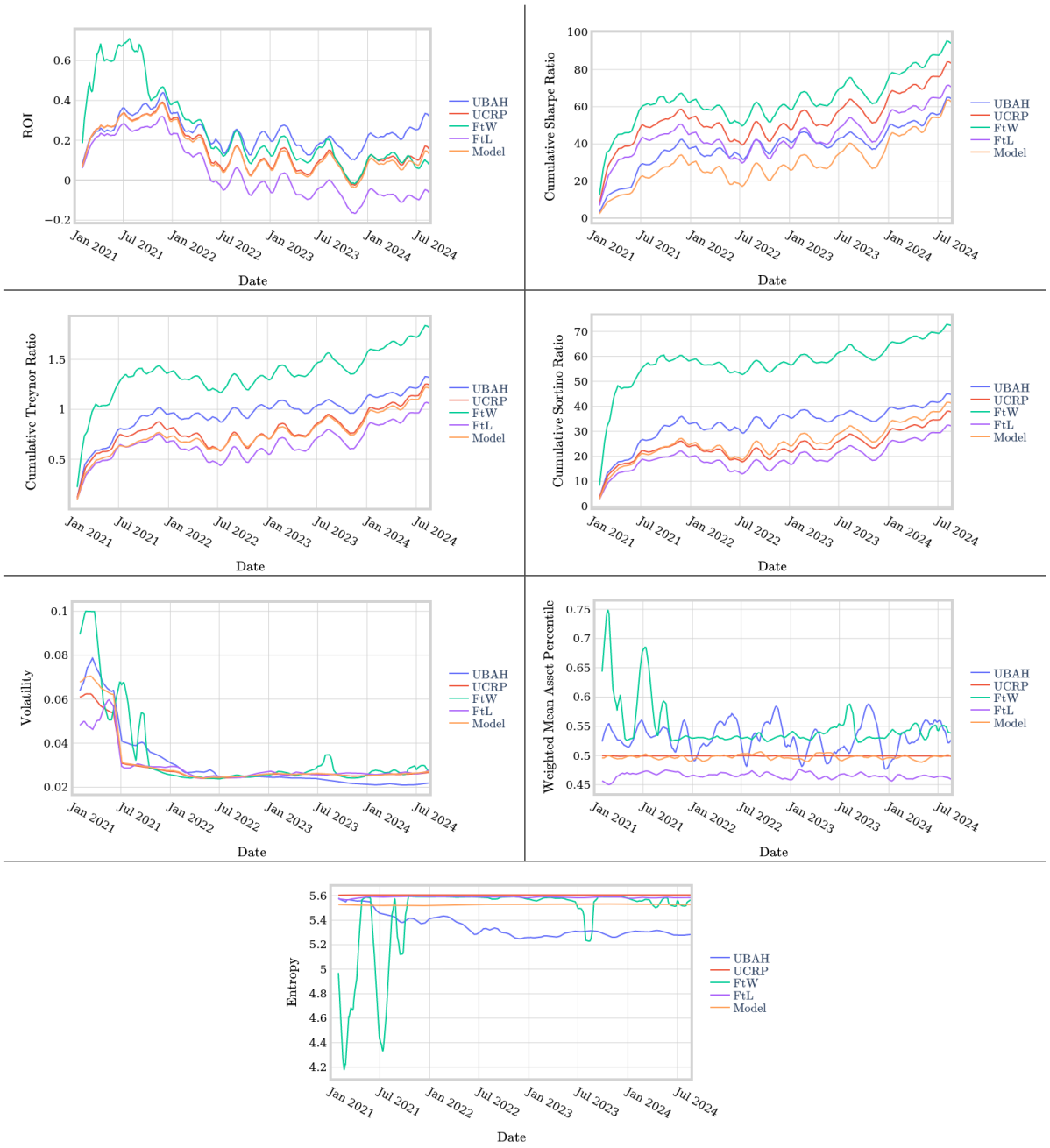


TABLE A.4: Preferred ROI reward function results across key evaluation metrics

A.1.5 Only Sharpe

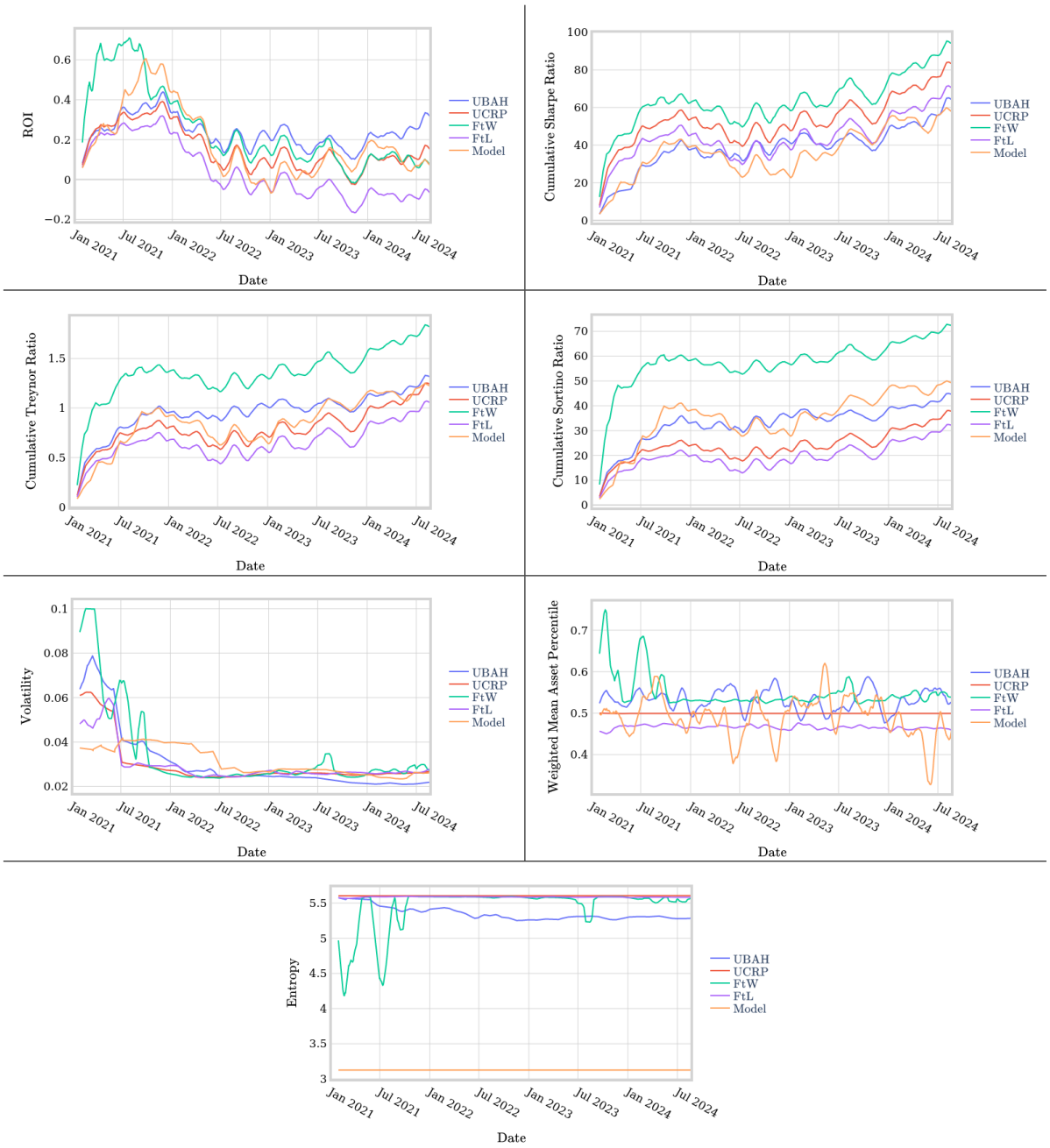


TABLE A.5: Only Sharpe reward function results across key evaluation metrics

A.1.6 Only ROI

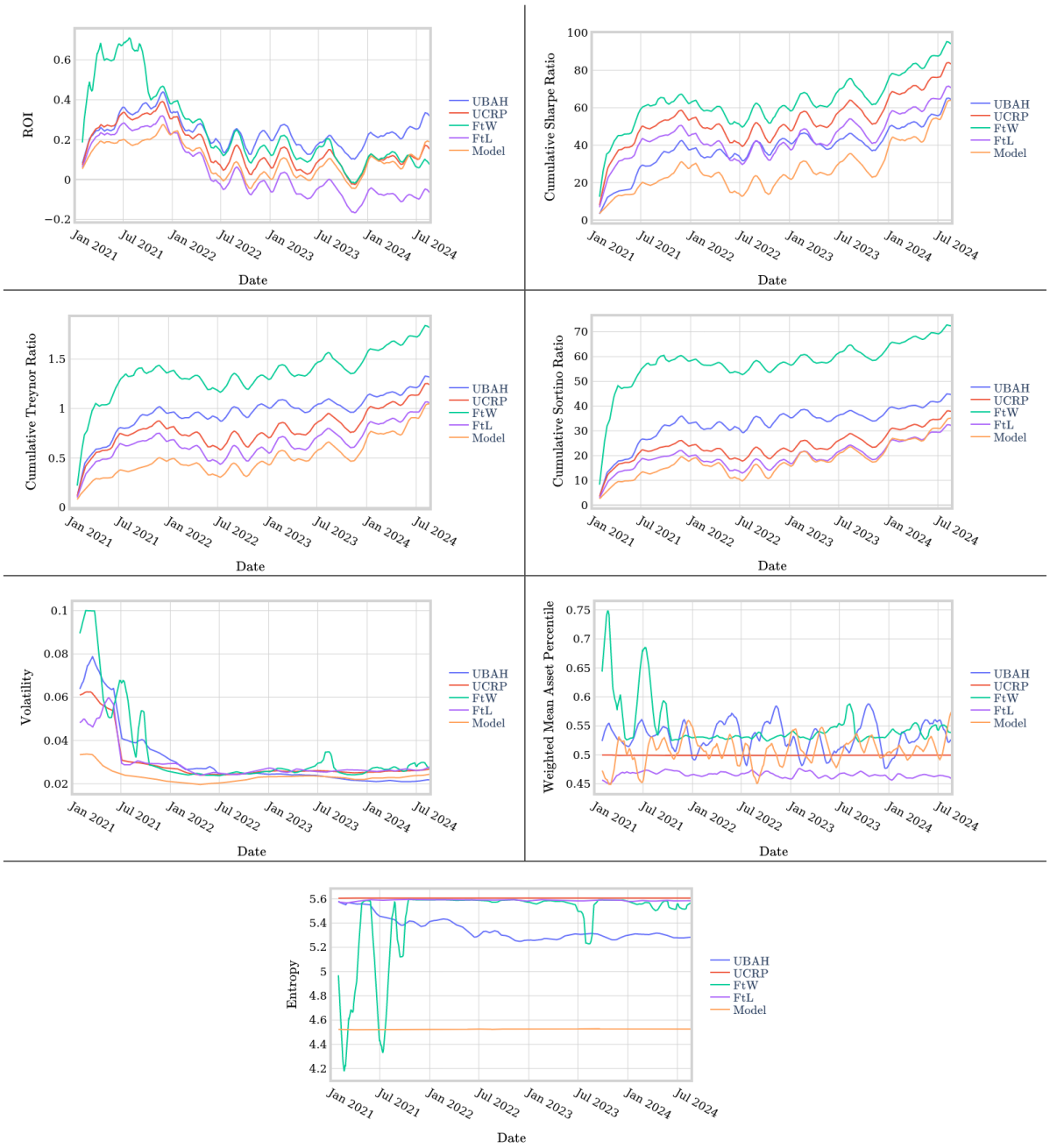


TABLE A.6: Only ROI reward function results across key evaluation metrics

A.2 Optimisations Using Flame Graphs

A flame graph is a visualisation tool used to analyse the performance of software by displaying the hierarchical structure of function calls. It shows how much time is spent in each function, with wider bars representing functions that consume more processing time. Flame graphs help identify bottlenecks in the code, making it easier to optimise performance. Developers can create flame graphs by running their code with a profiler such as cProfile.

A good example of how these plots were used to identify bottlenecks cam early on - where a method of the 'Asset' class called `closest_date_match()` which calculated the closest date in an asset's historical financial time series to a given date - a method used very frequently to ensure that data queries always returned a value - was identified as responsible for nearly 86% of the runtime due to it's use of nested loops. The profile for this code is seen in Figure A.1, with the guilty function being circled.

FIGURE A.1: Flame graph of code pre-optimisation

Once this bottleneck had been identified, it was optimised using search-sorted function with a much more reasonable level of computational complexity. Overall performance was notably better after but a profiler was still run over the model with this new feature. Figure A.2 shows the updated flame graph, with the `closest_date_match()` circled once again. From this it is clear to see the optimisations had a very good effect on code performance - with it now only being responsible for 4.12% of the runtime.

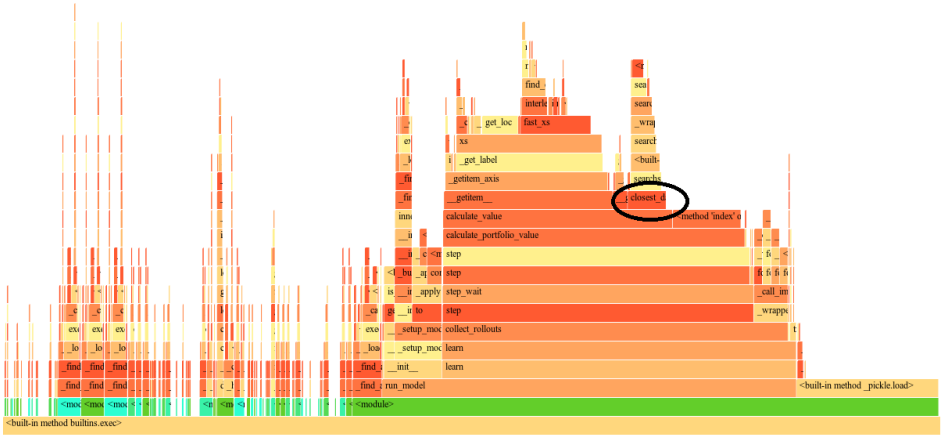


FIGURE A.2: Flame graph of code post-optimisation