Application of Reinforcement Learning in Algorithmic Trading

Jonathan Garvey BSc (06744885)
School of Computer Science
National University of Ireland Galway

Supervisor
Dr. Patrick Mannion

In partial fulfillment of the requirements for the degree of
MSc in Computer Science (Artificial Intelligence - Online)
September 2022

DECLARATION I, Jonathan Garvey, do hereby declare that this thesis entitled "Application of Reinforcement Learning in Algorithmic Trading" is a bonafide record of research work done by me for the award of MSc in Computer Science (Artificial Intelligence - Online) from National University of Ireland Galway. It has not been previously submitted, in part or whole, to any university or institution for any degree, diploma, or other qualification.

Signature: Jonathan Garvey

Abstract

Algorithmic trading is a process in which an algorithm coded as a piece of software, trades autonomously on behalf of a user or institution. One ML paradigm with much promise in this area is Reinforcement Learning (RL). Constrained RL is analogous to RL with guardrails and has been effective in mitigating catastrophic outcomes in high-risk environments. Q-Learning is a type of RL and has been successfully applied to problem-solving in various industries.

In this study, experiments are performed to determine if RL variation in Q-Learning parameters affects generalisation qualities of Agents on new financial markets. Both constrained and unconstrained RL approaches are taken with a view to assessing impact of constraint addition. The best learned policies are used to trade alongside a buy and hold strategy.

Results show that variation in the rate of Q-Learning exploration has a statistically significant effect on policy generalisation on previously unused data. The addition of constraints to mitigate risk leads to more profitable policies and the results are backed by statistical significance. Best-performing policies learned during a training phase outperform the buy and hold strategy in three popular markets.

Keywords: Reinforcement Learning, Q-Learning, Constrained Reinforcement Learning, Algorithmic Trading

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

CRL Constrained Reinforcement Learning. ix, 3, 69

CSV Comma Separated Values. 17

DC Directional Change. 12, 27

EMA Exponential Moving Average. 6, 9, 10, 25, 26

FOREX Foreign Exchange. 6, 11

HFT High-Frequency Trading. 11, 78

MA Moving Average. 8, 9

MACD Moving Average Convergence Divergence. 2, 6, 10, 25, 26

ML Machine Learning. iii, 1

NaN Not a Number. 31

NASDAQ National Association of Securities Dealers Automated Quotations. 17, 19, 22

OBV On Balance Volume. 10, 11, 25, 26, 78

OHLC Open High Low Close. 17

OS Overshoot. 6

RC Regime Change. 12

RL Reinforcement Learning. iii, ix, xii, 1–3, 12, 14, 29, 40, 42, 44, 45, 47, 48, 52–54, 57, 61, 65–68, 74, 76, 77

RSI Relative Strength Index. 3, 6

SMA Simple Moving Average. 8

TD Temporal Difference. 12

USD United States Dollars. 17

Chapter 1

Introduction

Algorithmic trading of financial instruments using Machine Learning (ML) techniques is becoming increasingly popular [1]–[5]. Conditions such as fundamental [6] and technical [7] can be used to analyse markets before writing an algorithm. Fundamental analysis establishes the health of an entity by its reviewing financial elements [6]. Technical analysis uses statistics and other numerical means to derive trends, indicators and oscillators [8] which may aid decision making when trading.

Reinforcement Learning (RL) is paradigm of Machine Learning (ML) and has been successfully applied to algorithmic trading scenarios[9]. RL is conceptually similar to how humans learn by trial and error[10] and in fact, RL has its roots in psychology[11, p. 13]. Perhaps this is one of the reasons for the success of RL in algorithmic trading. While ML techniques involve some degree of human input to produce a trained model, RL is a framework whereby Agents learn from their environment and operate autonomously without the need for human intervention[12].

There are several different approaches to, or types of RL[13]. This study uses the Q-Learning technique. For now, suffice it to say that Q-Learning does not require a complete model of its environment[14]. It provides a mechanism to control, in a probabilistic sense, how often random actions are selected[15]. This is critical. While humans, in some ways, benefit from venturing into the unknown, RL Agents also learn from experimenting with alternative actions to the ones they know are best at a point in time[16]. Accordingly, as environment knowledge is built up, Agents can begin to exploit prior learning to begin to master their surroundings[16]–[18]. As experience increases, the need to explore as opposed to exploit, can be decreased[19]. In certain environments, it is absolutely possible that an Agent reaches a stage where it no longer needs to explore and can therefore refer to its bedrock of optimal actions as a policy[20]. But what about taking a policy and using it to operate in different, but similar environment? Does the rate of exploration during learning have any impact on how the Agent performs in similar environments? In the context of financial instrument trading, this is one of the questions this study aims to address.

The term 'environment' has been mentioned several times thus far. In RL, an algorithmic environment is one that provides provides Agents with a representation of the financial market[21]. Agents take action on the market through the environment and receive a reward, or penalty[22]. From a technical, yet basic standpoint, financial markets are represented by price and volume[23]. Price refers to the cost of a financial instrument and volume refers to the amount of that instrument that has been traded in a specific time-period. Price and volume can be represented using mathematical and statistical operations on the data[24]. The term 'technical indicator' describes such encoding and many technical indicators exist; Stochastics, Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI) to name a few[25]. Some technical indicators provide market representations that denote if a financial instrument is overbought or oversold[26], or if there are changes in trend[27].

Given that Agents start off with no knowledge of their environment, it is absolutely possible that in an algorithmic trading scenario that Agents lose money, at least initially, in situations where technical indicators clearly indicate that taking a specific action may not be desirable[28]. But what if there was a way to implement safety guardrails, or constraints in RL? An area with much promise in dealing with this very problem is Constrained Reinforcement Learning (CRL). Constrained Reinforcement Learning (CRL) implements prior learning, heuristics, or safety rules, from the outset of learning, to prevent catastrophic behaviour[29]. This study involves experimentation with CRL in an algorithmic trading scenario to determine if the addition of constraints can improve profitability of Agents.

Countless human-designed algorithmic trading strategies have been created since the inception of algorithmic trading[1]–[5]. Perhaps one of the most common trading strategies for beginners is the ‘buy and hold’ strategy[30], [31]. In buy and hold, an order is issued to purchase a financial instrument and it is held on to until the owner decides to sell it. While there is inherent risk with this approach[32], it has proven time and time again to be a successful strategy[33]. A litmus test for any algorithmic trading strategy is to compare results with the buy and hold strategy[34], [35]. Of interest to this study, is to Agents’ learned knowledge on new, previously unused financial data, to see how they perform. In particular, can rulesets, or policies, learned by Agents, outperform the buy and hold strategy on some of the most common financial markets in existence?

1.1 Research Questions

1. RQ1. How does variation in the Exploration Rate during training affect performance on unseen data?

2. RQ2. Can constrained Reinforcement Learning be applied to improve performance of Reinforcement Learning Agents in an algorithmic trading scenario?

3. RQ3. Can Reinforcement Learning generate trading strategies that outperform a buy and hold approach?

Chapter 2

Background and Related Work

2.1 Algorithmic Trading

In the context of finance, algorithmic trading is a type of trading whereby buy and sell signals are generated and executed as trades when market conditions meet certain criteria in a ruleset, or algorithm. Algorithmic trading has existed as long as both computers and the stock market have co-existed. However, it is estimated that over the last twenty years, algorithmic trading has become so prevalent in trading that an estimated 80% of all trades transacted on the stock market are said to originate from automated algorithms.

Any worthwhile treatment of algorithmic trading must consider the types of variables that can present themselves in algorithms destined for automation on a financial exchange. There are two main categories of input variables that are prevalent both in the literature and in practice; endogenous and exogenous. Often referred to as fundamental analysis, the study of exogenous factors involves researching macro-economic elements and financial documentation pertaining to a particular instrument. The term exogenous is so called as the factors which it describes are external to the low-level ebb and flow of the spot price within a market. A review of work on exogenous factors is outside the scope of this study. The focus is on algorithmic trading based on technical indicators, i.e. endogenous variables.

2.1.1 Endogenous Factors

Endogenous studies of financial instruments involves the technical analysis of statistics and indicator readings derived from price and volume movement of the instrument. New indicators are still being developed[36], but some of the classic indicators deal with phenomena such as momentum[37], overbought/oversold[26] and strength of direction[38]. In this study, the following timeseries technical indicators are of interest: Relative Strength Index (RSI), Stochastic RSI, Exponential Moving Average (EMA), and Moving Average Convergence Divergence (MACD). In addition, Overshoot (OS) and DC events, which are intrinsic-time events generated by applying the DC Framework to financial instrument data, are included for performance analysis. Descriptions and motivation of said indicator selection follows.

2.1.1.1 Relative Strength Index

Sudden and consistent buying of a stock such that the recent buying trend is relatively stronger than the time-period preceding it can result in the stock being considered ‘overbought’. Similarly, a stock can be considered ‘oversold’ if the recent trading trend is biased towards the sell-side. The Relative Strength Index (RSI) is a numeric method for quantifying this phenomenon and was derived by Welles Wilder [39, p. 63]. Studies have shown that strategies using RSI can be immensely profitable on FOREX markets[40] and such is the reason for its inclusion in market representation in this study. It is formally defined as[41]:

2.1.1.1.1 First Calculation

$$Average\ Gain = \tfrac{1}{n}\textstyle\sum_{i=1}^{n} gain_i$$

$$Average\ Loss = \tfrac{1}{n}\textstyle\sum_{i=1}^{n} loss_i$$

$$RS = \tfrac{Average\ Gain}{Average\ Loss}$$

$$RSI = 100 - \tfrac{100}{1+RS}$$

2.1.1.1.2 Subsequent Calculations

Same as First Calculation, with the following updates for Average Gain and Loss:

$$Average\ Gain = (Average\ Gain_{previous} \times 13) + (Average\ Gain_{currrent})$$

$$Average\ Loss = (Average\ Loss_{previous} \times 13) + (Average\ Loss_{currrent})$$

2.1.1.1.3 Usage values oscillate between 0 and 100. Readings above 70 indicate that a financial instrument is overbought. An RSI reading of 30 or less signifies that the instrument is oversold. Possible usage could be to buy an instrument when it is oversold and sell when it is overbought.

2.1.1.2 Stochastic RSI

George Lane developed what is known as the Stochastic Oscillator [42]. Its purpose, among other uses, is to predict inflection points in a financial instrument's momentum. It achieves this using support and resistance levels. Later, Stanley Kroll and Tushar Chande developed the Stochastic RSI [43]. The Stochastic RSI is derived from RSI. In particular, the Stochastic equation is applied to the RSI output to achieve readings between 0 and 100. Hence the Stochastic RSI is itself an oscillator and the readings represent where the RSI sits on the overbought-oversold spectrum. Its inclusion in this study is on the basis of a study that proved how the addition of Stochastic RSI to a trading strategy on the Karachi stock exchange resulted in out-performance of a buy and hold strategy, even with transaction fees taken into account[44]. Its inclusion in this study Stochastic RSI is defined as follows[45]:

$$StochasticRSI = \frac{RSI_{current} - RSI_{lowest}}{RSI_{max} - RSI_{lowest}}$$

where $RSI_{current}$ is the RSI for the current period, RSI_{lowest} is the lowest RSI reading over the last 14 periods, and RSI_{max} is the maximum RSI reading over the last 14 periods.

Stochastic RSI, when visualised, is plotted as two lines; *%K* and *%D*. They are formally defined as follows:

$$\%K = StochasticRSI$$

$$\%D = MA_{3period}(\%K)$$

%K and *%D* are also referred to as the 'fast stochastic' and 'slow stochastic', respectively. When *%K* or *%D* is above 70, the financial instrument is overbought. *%K* or *%D* levels below 30 indicates that the instrument is oversold. Several trading strategies have implemented the Stochastic RSI indicator[46], [47].

2.1.1.3 Moving Average

Price movement of financial instruments, by nature, is volatile and generates noisy data. Smoothing techniques are applied to the underlying timeseries data to increase the interoperability of the data. More importantly, smoothing techniques encode data from current and previous timesteps through aggregation functions. By specifying the number of timeperiods into the past to interpolate, it is possible to control how much historic data is encoded into the current smoothed function reading. The most common smoothing function is the Moving Average (MA), often referred to as the Simple Moving Average (SMA). It is defined as follows[48]:

$$SMA = \frac{1}{n}\sum_{i=1}^{n} price_i$$

where n is the number of sequential datapoints to aggregate, and $price_i$ is the price of a stock at position i in the sequence. The SMA provides equal cadence to all datapoints in the sequence.

2.1.1.3.1 Exponential Moving Average

An alternative approach is to apply a function to each datapoint such that datapoints closest to the current datapoint have a greater effect on the average than those that occurred earlier in the sequence. In the context of trading, this could be crucial as traders consider recent phenomenon as being of particular interest. One such algorithm is the Exponential Moving Average (EMA) and is derived from the Simple Moving Average. It is defined as[48]:

$$EMA_t = (price_t \times k) + (EMA_{t-1} \times (1 - k))$$

where k is a smoothing function as is defined as:

$$k = \frac{2}{N+1}$$

where N is the number of size of the EMA window in datapoints.

2.1.1.3.1.1 Golden Cross

When the 50-period MA crosses above the 200-period MA, it is knows as a *Golden Cross*[49]. There is generally positive sentiment around Golden Crosses as it represents the point where the medium-term price trend has surpassed the long-term price trend. This could signal impending buy action and subsequent price increases, offering profitable trade opportunities to speculators. Likewise, when the 50-period MA crosses below the 200-period MA, this could signal impending sell action and price decreases. Golden Crosses can also be calculated on EMA trends. In this study, Golden Crosses on EMA trends are used to encode favourable buying and selling conditions in the relevant market.

2.1.1.3.2 Moving Average Convergence Divergence

Moving Average Convergence Divergence (MACD) is a momentum indicator that encodes the difference between the 12-period EMA and the 26-period EMA of the price as a line called the MACD. The second element of the MACD indicator is a signal line, so-called as traders often trade on the basis of the MACD line crossing the signal line. A calculation of the signal line is performed by computing a nine-day EMA of the MACD line. Formally, MACD is defined as[50]:

$$MACD = EMA_{12period}(price) - EMA_{26period}(price)$$
$$signal = EMA_{9period}(MACD)$$

The MACD indicator is capable of capturing changes in buying and selling momentum. In particular, momentum shifts from selling to buying when the MACD line crosses above the signal line. Conversely, momentum shifts from buying to selling when the MACD line crosses below the signal line. Many trading strategies have been implemented using the MACD indicator, inclduing [51], [52].

2.1.1.4 On Balance Volume

On Balance Volume (OBV) is a momentum indicator that represents how much net volume is present in a market at a point in time. This is achieved by summing for all periods in the instrument's history, the volume for periods where an uptick in price occurred since the previous price, and subtracting the volume for all periods where a decrease in price occurred since the previous period. It is formally defined as[53]:

$$OBV = OBV_{prev} + \begin{cases} volume, & \text{if } close > close_{prev} \\ 0, & \text{if } close = close_{prev} \\ -volume, & \text{if } close < close_{prev} \end{cases}$$

With consistent buying momentum, OBV creates an increasing trend in the positive direction. This is not to be confused with a positive reading of OBV, which signifies that cumulatively, there is more positive than negative volume traded in the market. Conversely, consistent selling momentum causes OBV to trend in the negative direction. A net-negative reading of OBV is reached when the total negative volume traded in a system is greater than the net positive volume traded. Said trends are of high importance to traders and indeed the OBV has proven to be a profitable indicator[54], hence its inclusion in this study.

2.1.1.5 Directional Change

In technical analysis, data is primarily time-series in nature [55, p. 2] and altering time scales can produce different results [56]. In algorithmic trading, charts are generally viewed in intervals of one hour or less, but near real-time algorithmic trading known as High-Frequency Trading (HFT) is commonplace. Yearly time periods are often viewed in daily intervals which highlight the effects of major political and economic events on financial markets. Tsang and Chen [57] showed this in their analysis of the impact of Brexit on popular Foreign Exchange (FOREX) markets.

Regime Change is an important phenomenon of market behaviour. As market participants' sentiment changes, their new evaluation of how much the financial instrument is worth is manifested through financial instrument trading, leading to movements in price [55, p. 2]. If the collective sentiment changes, one would expect a significant change in the overall price trend. Detecting major shifts in a financial market could be useful [57].

Glattfelder et al. [58] proposed an effective approach to RC detection called Directional Change which uses scaling laws to operate in an intrinsic time paradigm instead of time-series. Building on this pioneering work, a complete and practical artefact on the topic is a book by Chen and Tsang [55]. Machine Learning has helped DC transition from a set of scaling laws to having practical use cases [55, pp. 79-91], [59]. In any case, given the RC detection qualities of the DC framework, DC indicators are included to co-represent the state-space of environments created during this study.

2.2 Reinforcement Learning

Reinforcement Learning is a paradigm of Machine Learning in which agents navigate through an environment and learn from rewards received from taking actions along the way. Humans may design a reward structure to encourage Agents to achieve a goal but critically, the Agents are not told how to achieve the goal. Instead, through trial and error, they learn the ‘how’, and eventually solve the creator’s problem. The definition of an Agent is entire body of work in itself. Having said that, it is common to refer to Agents as being autonomous and reactive[12].

2.2.1 Q-Learning

Q-Learning is an off-policy Temporal Difference (TD) RL algorithm and the central algorithm in this experiment. It is applied to financial market data in a trading capacity in order to learn the actions that maximise profit. Q-Learning achieves this through the use of dynamic programming. Several attempts have been made to apply Q-Learning to algorithmic trading[28], [60]–[62]. As Q-Learning is TD RL algorithm, it does not require a complete model of the environment. As an Agent navigates through the environment, it chooses an action,

a, for the state, s, it is currently in. It then transitions to a new state, s', and receives a reward, r. The value of choosing action a in state s is given by:

$$Q(s,a) \leftarrow Q(s,\ a) + \alpha[r + \gamma\ \max_a Q(\mathrm{s}',\ a)\ \text{-}\ Q(s,\ a)]$$

where α and γ are the learning rate and discount factor, respectively.

The learning rate controls how quickly the algorithm learns. As $\alpha \to 0$, $Q(s,\ a)$ is assigned to itself after each transition. As $\alpha \to 1$, $Q(s,\ a)$ is assigned the value of $r + \gamma\ \max_a Q(\mathrm{s}',\ a)$. In other words, the value of $Q(s,\ a)$ is not considered in the update and the new value of $Q(s,\ a)$ becomes a function of the reward and the discounted value of the best action to take in s'.

The discount factor controls how much of the expected maximum future value to include in the update step. As $\gamma \to 0$, the expected maximum future value is not taken into account and $Q(s,\ a)$ is increased only by $\alpha[r\ \text{-}\ Q(s,\ a)]$. As $\gamma \to 1$, the entirety of the expected future reward, $\max_a Q(\mathrm{s}',\ a)$, is taken into account during the update and $Q(s,\ a)$ is increased by $\alpha[r + \max_a Q(\mathrm{s}',\ a)\ \text{-}\ Q(s,\ a)]$.
A step by step representation of the Q-Learning algorithm follows[11]:

Algorithm 1 Q-Learning Algorithm

```
Initialize $Q(s,\ a)$, $\forall s \in S$, $a \in A(s)$, arbitrarily, and $Q(\text{terminal}, \cdot) = 0$
repeat(for each episode):
    Initialise $s$
    repeat(for each step of episode):
        Choose $a$ from $s$ using policy derived from $Q$ (e.g., $\epsilon$-greedy)
        Take action $a$, observe $r$, $s'$
        $Q(s,a) \leftarrow Q(s,\ a) + \alpha[r + \gamma\ \max_a Q(\mathrm{s}',\ a)\ \text{-}\ Q(s,\ a)]$
        $s \leftarrow \mathrm{s}'$
        until $s$ is terminal
```

2.2.2 Constrained Reinforcement Learning

In certain scenarios, it is not always ideal, or even safe, to implement Agents with no action restrictions. In particular, by providing the ability to Agents to take any action $a \in A$ in every state $s \in S$, inherent risk is introduced if there exists at least one action that leads to potentially harmful, or highly adverse consequences. Autonomous vehicles is an area where this is prevalent. Autonomous driving is an area where the risk of harmful consequences as a result of taking undesirable actions is prevalent[29]. Constrained Reinforcement Learning aims to tackle this issue. In constrained RL, actions are limited to those that do not lead to such unfavourable consequences. In doing so, Agents learn as they do in a constrained approach, exploring and exploiting as per their configuration, yet their action-set is limited in some states to mitigate risk.

Chapter 3

Data

3.1 Source Data

Trading data is typically available in time-series format at a specific temporal granularity, e.g. hourly / daily. For each time interval, the opening, lowest, highest and closing prices, are supplied. They refer to the financial instrument price at interval onset, the highest price reached during the interval, the lowest price reached during the interval, and the very last price of the instrument before the interval terminates. Said prices are synonymous with algorithmic trading and are referred to as OHLC prices. Trading volume, which is the total unit quantity of a financial instrument traded in a time interval is often, though not always, provided with each interval.

Table 3.1 shows OHLC data obtained for and consumed in this study. BTC-USD and EOS-USD data were both obtained from cryptodatadownload.com [63], [64]. NASDAQ-USD data was obtained from backtestmarket.com [65].

Instrument	Granularity	Period	Columns	Format
BTC-USD	Minute	1/1/2015 - 31/7/2022	• Date • Unix • [OHLC] • Volume BTC • Volume USD	CSV
EOS-USD	Hourly	15/5/2018 to 31/7/2022	• Date • Unix • [OHLC] • Volume EOS • Volume USD	CSV
NASDAQ-USD	Hourly	1/4/2007 - 19/7/2022	• Date • Time • [OHLC] • Volume	CSV

Table 3.1: Source Data Description.

3.1.1 Data Description

3.1.1.1 BTC-USD Data

This dataset contains historic Bitcoin spot prices from 1/1/2015 to 31/7/2022. It comes in CSV format and can be downloaded from [63]. A separate file is provided for each year. Data is provided at a granularity of one minute, with each interval of one minute corresponding to one row in the file. For each interval, a timestamp is provided, along with a Unix date, Open High Low Close (OHLC) prices, and volume traded in that interval, both in Bitcoin and United States Dollars (USD).

3.1.1.2 EOS-USD Data

This dataset contains historic EOS spot prices from 15/5/2018 to 31/7/2022. It comes in CSV format and can be downloaded from [64]. Data is provided at an hourly granularity, with each hourly interval corresponding to one row in the file. For each interval, a timestamp is provided, along with a Unix date, OHLC prices, and volume traded in that interval, both in EOS and USD.

3.1.1.3 NASDAQ-USD Data

This dataset contains historic NASDAQ spot prices from 1/4/2007 to 19/7/2022. It comes in CSV format and can be downloaded from [65]. Data is provided at an hourly granularity, with each hourly interval corresponding to one row in the file. For each interval, a date is provided, along with a time, OHLC prices, and volume traded in that interval, in USD.

3.1.2 Data Preview

A preview of the data is shown below.

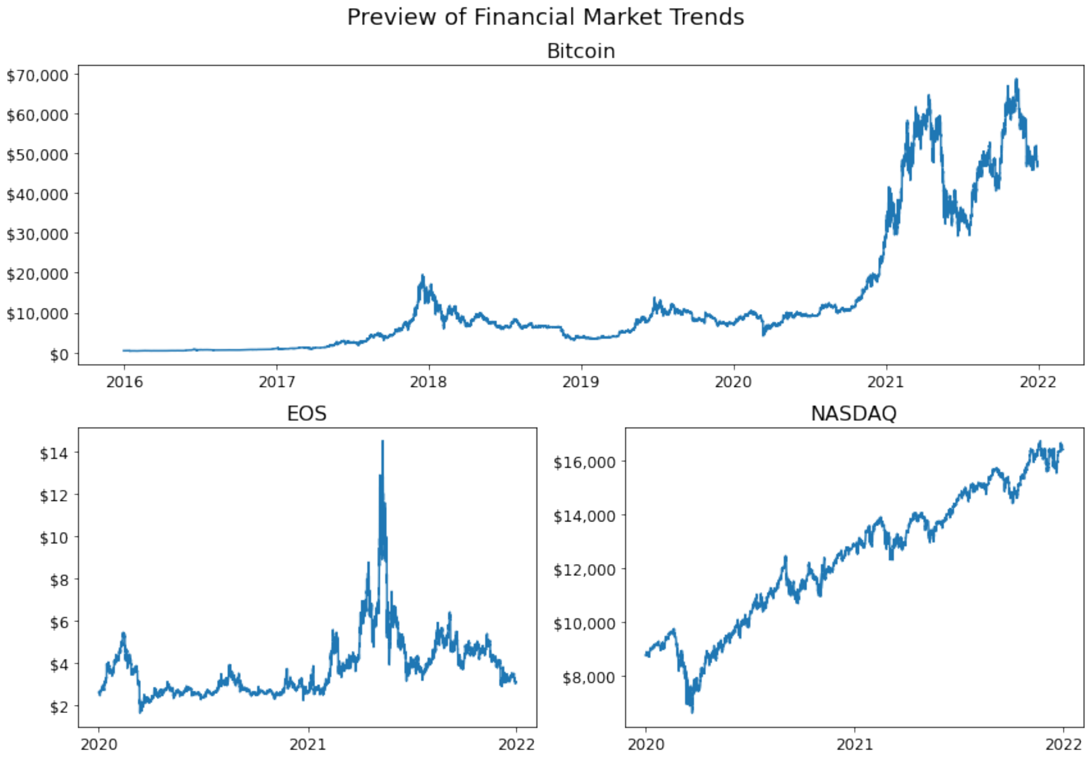

Figure 3.1: Financial Markets Trends Preview

3.1.3 Data Selection Motivation

While several thousand cryptocurrencies have been created and are traded daily, Bitcoin remains the original, most dominant, and widely known cryptocurrency. It makes sense therefore to select a Bitcoin market as the benchmark market for this study. EOS is another well-known cryptocurrency and critically, as the trends in Figure 3.1 show, closing prices for BTC-USD and EOS-USD are effectively uncorrelated with each other. This is positive in the context of this study as generalisation on unseen, uncorrelated markets is desirable.

The National Association of Securities Dealers Automated Quotations (NASDAQ) is one of the world's largest stock exchanges. Large companies from sectors such as technology, healthcare and consumer services are listed on the exchange. A measure of how well the companies are performing as a whole is available through the NASDAQ Composite Index, also known as "the NASDAQ". This study includes the NASDAQ in its scope. It was selected for two reasons. Firstly, it is of interest to this study to observe and analyse trained Agents' performance on financial markets both inside and outside the cryptocurrency space. Since the NASDAQ is is not a cryptocurrency, it meets said criteria. Secondly, the majority of companies listed on the NASDAQ are tech companies. And while generalisation on unseen and uncorrelated markets to those used during training is of course preferred, the NASDAQ with its technical prevalence, is a logical next step for progressive generalisation testing. Figure 3.2 highlights the high, positive correlation between BTC-USD and NASDAQ-USD. This is not ideal, yet for the aforementioned reasons the study will proceed with the NASDAQ in scope.

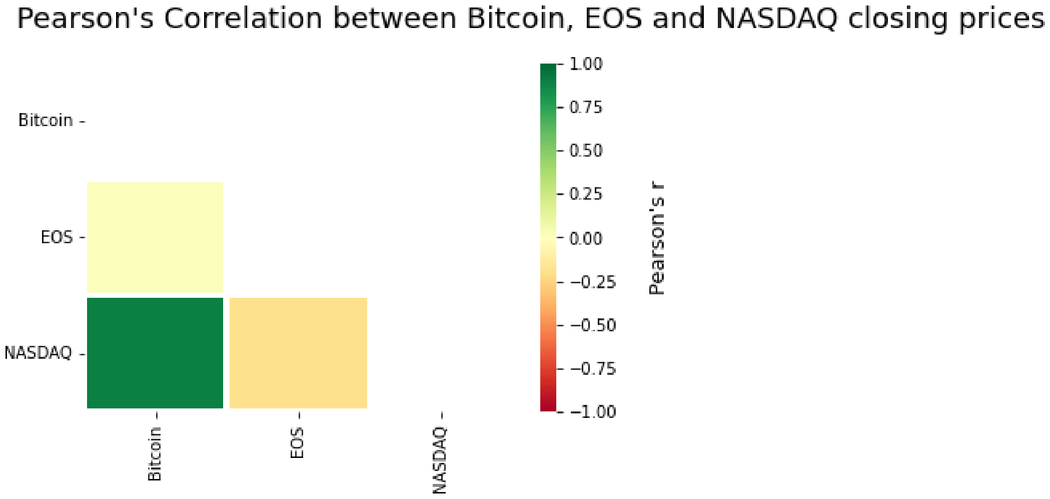

Figure 3.2: Correlation between BTC-USD, EOS-USD and NASDAQ-USD

3.2 Preprocessing

Requirements:

1. Produce three tables, each representing timeseries data from one of the three financial instruments in scope.

2. Tables should have an identical schema and structure, free of duplicates and missing values, and contain data pertaining to a common date period.

3. Each table should contain a common set of financial indicator readings.

3.2.1 Preprocessing Actions

Source files for each of the three financial instruments were loaded into separate tables ("DataFrames") using the *read_csv()* function of the *Pandas* Python library. Table 3.2 shows the complete set of data preprocessing steps applied to the imported data. Descriptions of same are provided in sections 3.2.2 - 3.2.4.

Source Market	Column(s)	Action
BTC-USD	date	Set as DataFrame index
	unix, symbol, open, high, low, Volume BTC	Dropped
	N/A	Duplicate rows removed
	Volume USD	Renamed to ‘volume’
	Rows with missing ‘close’ values	Removed
	Entire DataFrame	Time-Resampled from minute to hourly
	Entire DataFrame	Time-Resampled from hourly to daily
	Hourly DataFrame	Indicators and DC events added
	Daily DataFrame	Indicators added
EOS-USD	date	Set as DataFrame index
	unix, symbol, open, high, low, Volume EOS	Dropped
	N/A	Duplicate rows removed
	Volume USD	Renamed to ‘volume’
	Rows with missing ‘close’ values	Removed
	Entire DataFrame	Time-Resampled from minute to hourly
	Entire DataFrame	Time-Resampled from hourly to daily
	Hourly DataFrame	Indicators and DC events added
	Daily DataFrame	Indicators added
NASDAQ-USD	date, time	Combined and used to set DataFrame index
	date, time, open, high, low	Dropped
	N/A	Duplicate rows removed
	Rows with missing ‘close’ values	Removed
	Hourly	1/1/2015 - 7/31/2022
	Entire DataFrame	Time-Resampled from hourly to daily
	Hourly DataFrame	Indicators and DC events added
	Daily DataFrame	Indicators added

Table 3.2: Data Preprocessing Steps

Where date and time were stored in seperate columns at source, they were combined into a single datetime column for experimentation. Finally, the datetime column datatype was changed from text to timestamp to enable interaction with the data using datetime functionality. All columns except for datetime and close were dropped as they did not form part of the data requirements.

3.2.2 Indexing

Indexing is a common optimisation practice when storing and processing large datasets. In the context of this experiment, data sizes were not large enough to warrant the application of extensive indexing techniques. However, the Pandas library provides functionality which enables trivial and expedient processing of, and interaction with, timestamps. To avail of this functionality, Pandas requires that DataFrames have datetime indices. As such, a datetime index was created for each of the three tables. For Bitcoin and EOS, a 'date' column, which contained a date and time combined, was available in the source dataset and was used as the index. For the NASDAQ data, the date and time were provided at source as two separate columns. These were combined to produce the index.

3.2.3 Removal of Unused Columns

Data exploration showed instances of duplicate records in some of the source datasets. As such, data-deduplication was applied as a preprocessing step to nullify the impact of duplicate records in experimentation.

3.2.4 Data Deduplication

Data exploration showed instances of duplicate records in some of the source datasets. As such, data-deduplication was applied as a preprocessing step to nullify the impact of duplicate records in experimentation.

3.2.5 Missing Value Handling

Rows with missing values in the ‘close’ column in each DataFrame were removed. Removal was preferred over replacement with aggregated or smoothed values. This is because the selection of a smoothed function is data and market dependent, and somewhat arbitrary, given the stochastic nature of financial instrument data. In addition, the BTC-USD source data is at a minute-granularity, and since the time-resampling to an hourly granularity only considers the last closing price for the hour, the removal of missing values within-hour did not affect the training data quality.

3.2.6 Time Resampling

Timeseries technical indicators can be added to timeseries data of any granularity. This study used indicators at hourly and daily timeframes for each financial instrument. Resampling of the source data was required to fulfill this requirement. The *DataFrame.resample()* function from the *Pandas* library was utilised for resampling. *DataFrame.resample()* takes the to-be interval, the value to aggregate and the aggregation operations, as parameters. Table 3.3 shows the specification of the resampling carried out.

Instrument	Source Granularity	Resample Specification
BTC-USD	Minute	• Closing Price (Last) per hour • Volume (Sum) per hour • Closing Price (Last) per day • Volume (Sum) per day
EOS-USD	Hourly	• Closing Price (Last) per day • Volume (Sum) per day
NASDAQ	Hourly	• Closing Price (Last) per day • Volume (Sum) per day

Table 3.3: Time Resampling

3.2.7 Technical Indicator Addition

As described in Chapter 4, the state space formulation is compromised of a number of financial indicators. It was therefore required to compute readings for all indicators in scope, for each interval in each financial instrument dataset. Indicators are categorised as being either timeseries or time-intrinsic and a different approach for computation for each indicator category was taken and is described below.

3.2.7.1 Time Series Indicators

Time-series indicators as described in Chapter 2 were computed for each financial instrument's dataset, both at hourly and daily granularity. This was a two-step process.

3.2.7.1.1 Step 1 - Preparation

Firstly, a Python library called *TA-Lib* was used for computing the indicators. Each indicator was added as a column to both the hourly and daily DataFrame for each financial instrument. Table 3.4 shows the indicators computed for during this step.

Indicator Computed	Granularity
RSI	Hourly, Daily
Stochastic RSI (i.e. %K and %D)	Hourly, Daily
$EMA_{50period}(close)$	Hourly, Daily
$EMA_{200period}(close)$	Hourly, Daily
MACD	Hourly, Daily
OBV	Hourly, Daily

Table 3.4: Timeseries Indicators

3.2.7.1.2 Step 2 - Computation of State-Space Variables

Table 3.5 shows the additional steps applied to transform the core indicator readings into the Q-Learning environment state-space variables.

Input Indicator	Applied Step	Reasoning	Output Indicator(s)
RSI	Discretised to Integer.	To avoid complexity in handling continuous RSI readings.	RSI discrete (hourly, daily)
Stochastic RSI (i.e. %K and %D)	Encoded as %K having crossed over or under %D in the last 5 periods. Boolean.	To align with Stochastic RSI crossovers signalling changes in buying and selling momentum.	Stochastic RSI cross-up recent (hourly, daily), Stochastic RSI cross-down recent (hourly, daily)
$EMA_{50period}(close)$	Encoded as a Golden Cross in either updward or downward direction having occurred in last 5 periods. Boolean.	To align with Golden Crosses signalling positive or negative sentiment milestones of a financial instrument.	EMA cross-up recent (hourly, daily), EMA cross-down recent (hourly, daily)
$EMA_{200period}(close)$			
MACD	Encoded as MACD having crossed over or under signal line in last 5 periods. Boolean.	To align with MACD signal-line cross-overs indicating buying or selling momentum shifts.	MACD cross-up recent (hourly, daily), MACD cross-down recent (hourly, daily)
OBV	Normalised and discretised to Float.	To avoid curse of dimensionality, and complexity in handling continuous OBV readings.	MACD OBV normalised (hourly, daily)

Table 3.5: Indicator Computation - Step 2

3.2.7.2 Directional Change Events

The following DC events were computed for each financial instrument. Since DC is an intrinsic-time framework, DC events are time-interval agnostic. However, DC events were computed added to each financial instrument's hourly DataFrame. This decision was made to mitigate the risk of missing out on intra-day DC events. Table 3.6 shows the DC events computed using the hourly datasets, and the DC threshold chosen.

Event	DC Threshold
Directional Change Event	7.5%
Overshoot Event Upward Direction	
Overshoot Event Downward Direction	

Table 3.6: Directional Change Events

7.5% was selected as the DC threshold as it sits between day-trading (1-5% increases \ decreases in stock price) and position-trading (>10% increases \ decreases in stock price). By selecting a value in between said ranges, the DC framework captures short to medium-term Directional Change. This is ideal for algorithmic trading.

Chapter 4

Methodology

4.1 High-Level Approach

Q-Learning environments are proposed for agents to assimilate market conditions and gain experience. There are several indicators and DC events which could form the state space. Therefore a brute force approach to experiment with all possible state space representations is taken.

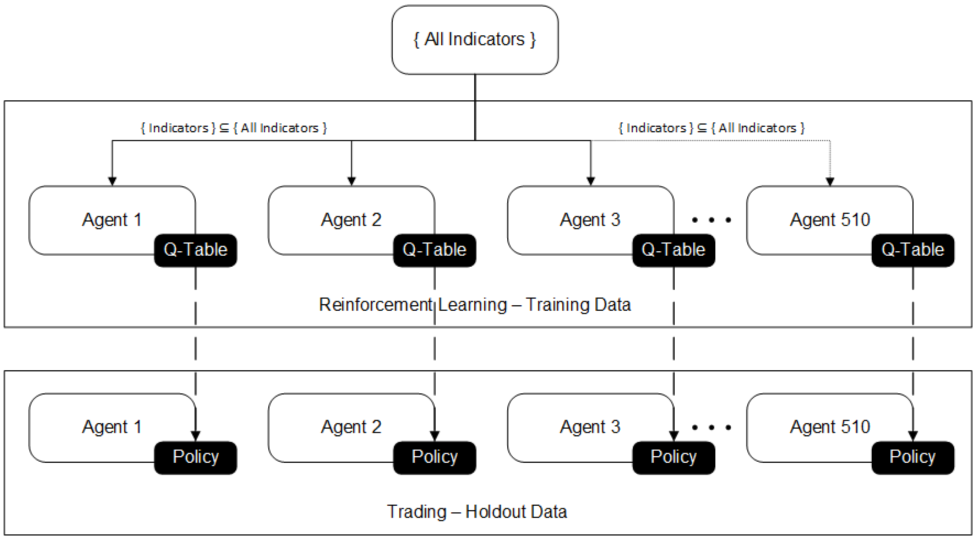

Figure 4.1: Q-Learning for Trading - High Level Approach

4.2 State-Space Combinations

RL Agents are created and assigned a state-space representing one subset of all possible indicator and DC event combinations. The number of state-space combinations is calculated using the *combinations()* function of the Python *itertools* library. It takes a list of entries and combination length as inputs and outputs a list of combinations. The following is the list of indicators used:

1. (stoch_RSI_cross_up_recent_hour, stoch_RSI_cross_down_recent_hour)

2. (stoch_RSI_cross_up_recent_day, stoch_RSI_cross_down_recent_day)

3. (EMA_cross_up_recent_hour, EMA_cross_down_recent_hour)

4. (EMA_cross_up_recent_day, EMA_cross_down_recent_day)

5. (MACD_cross_up_recent_hour, MACD_cross_down_recent_hour)

6. (MACD_cross_up_recent_day, MACD_cross_down_recent_day)

7. (is_OS_event_up, is_OS_event_down, is_DC_event)

8. (RSI_discrete_hour, RSI_discrete_day)

9. (OBV_norm_hour, OBV_norm_day)

Note, where indicators differ only by the nature of their direction, they are grouped together. For example, Stochastic RSI upward and downward crosses are treated as one entry in the indicator list for combination calculation purposes. There are two exceptions to this approach: the hourly and daily discretised RSI are grouped together, as are all of the DC event indicators. This decision is driven by the need to avoid an arbitrarily large number of state-space combinations, which greatly increases the time and compute resources required for experimentation.

Starting with an indicator combination length of 1 and increasing until 9 (i.e. the length of the list of indicator groups), *itertools.combinations()* creates 510 combinations, which is equal to the number of all possible combinations from the list of indicator groups.

Out of the three markets in scope in this study, data from a single market is used for training. However, not all of the market's data is used for training. In particular, a data subset pertaining to the period of 1/1/2016 to 31/12/2019 is selected. The Agents' goals are to learn trading policies that maximise a reward, in this case - % profit. Agents and associated policies are then assigned to trade on previously unused data, i.e. data which is set aside for testing. Figure 4.1 depicts a conceptual model of this approach.

A state-space combination pertain to all possible indicator combinations

4.2.1 ϵ-Greedy Policy

At each state, the algorithm chooses either a random action (*explore*) or an action that exploits prior learning (*exploit*). In this experiment, an epsilon-greedy ("ϵ-greedy") policy is used. ϵ-greedy is a probabilistic approach for choosing an action which uses a parameter, ϵ, to control the probability of selecting a random action. An Agent operating in a Q-Learning environment with $\epsilon = 0.2$ will select a random action 20% of the time. 80% of the time it will select the action corresponding to the highest Q-value in the current state.

4.2.1.1 Q-Value Storage

NumPy is a Python library for performing mathematical computation on arrays. For each Agent, a two-dimensional *NumPy* array is created to store Q-values.

The number of rows is equal to the total number of states, $\|S\|$. A column is created for each action a in A. However, as each Agent could have a potentially different number of states, the Q-table size is standardised in order to make use of *NumPy's* computational performance gains through vectorisation. The Agent with the largest number of distinct states is selected and this number is assigned to the rowcount of the standard array size. The number of columns is equal to the number of actions, i.e. three. While each Agent is assigned a standard Q-table size, Agents that do not encounter a sufficient number of states as to utilise all rows the array, will have the unused rows masked to *NumPy's Not a Number (NaN)* type. Not a Number is a valid value in a *NumPy* array and is used to represent numbers that are not known. In this case, the use of NaN is for masking purposes and to achieve array size standardisation.

Q-tables are grouped by exploration rate specification. This means that while each Agent is assigned its own Q-table (two-dimensional array), an exploration rate specification is assigned a three-dimensional array; an array of Q-tables.

4.2.1.2 Q-Value Estimation

A tabular approach for Q-value storage and estimation is taken for two reasons. Firstly, indicator readings for training and test datasets are computed before experimentation. This is made possible through the nature of the experiment, i.e. data is historic and therefore, all states in the context of the data are known. This is significant as the number of distinct states, hence Q-table size, can be calculated. Secondly, the values of the state-space variables in this experiment are discretised. This reduces the number of possible states of the environment when compared to the scenario where unadulterated, continuous values of the underlying indicators are used. Furthermore, discretisation of values helps mitigate the curse of dimensionality.

A neural network approach for Q-value estimation was considered for this experiment and ultimately deemed inappropriate. This would be better suited to a live environment where previously unseen state-action pairs are encountered and there exists a necessity to estimate Q-values for same.

4.3 Training

The BTC-USD financial market is selected as the Reinforcement Learning environment. Technical indicators represent the market state. Furthermore, for each technical indicator combination, an Agent is created. Specifically, there is a one:one mapping of technical indicator combination to Agent. The state-action space is formulated with the following actions which are available for each state:

- Buy
- Sell
- Do Nothing

4.3.1 Exploration Rate Variation

As per research question RQ2, this study aims to investigate the impact of exploration rate variation during training on the performance of Agents on unseen data. There, six distinct settings of the exploration rate parameter, ϵ, are selected (see Experimental Settings). For each exploration rate setting, a separate training run is executed. This produces six distinct result sets. A training run is defined as one pass over training data for all Agents for one setting of ϵ.

4.4 Testing

On training completion, there are six sets of Q-tables, corresponding to six exploration rate specification IDs. Each Q-table in a Q-table set corresponds to a learned policy for one Agent. The policies are then used for trading on holdout data that was not used during training. The exploration rate specification ID pertaining to each Agent is recorded during testing, along with the policy performance data. By doing so, impact of exploration rate variation during training on testing performance can be quantified.

In this experiment, holdout data is made available for the Bitcoin, EOS and NASDAQ financial instruments (Table 3.1). This is in order to experiment on the generalisation qualities of policies learned from a particular market; not just on the same market, but also on other, previously unseen markets. The time period for each holdout dataset is 1/1/2020 to 31/12/2021.

For each market-exploration rate specification-Agent combination, a test is performed. At each state, s, the action a, to perform is retrieved from the policy and the environment transitions into a new state, s'. Rewards and portfolio balance are recorded after each transition. However, no updates to $Q(s, a)$ are performed.

Experimental settings and intended analysis are detailed in Experimental Settings.

Chapter 5

Experimental Settings

5.1 Exploration Rate Specification

Critically, the study aims to determine if variation in exploration rate affects performance on unseen data. Consequentially, the following exploration rate specification is utilised:

Exploration Rate Specification ID	ϵ	ϵ Decay
1	0.9	True
2	0.7	True
3	0.3	True
4	0.1	False
5	0.05	False
6	0.025	False

Table 5.1: ϵ Exploration Rate Specification

Each row in Table 5.1 represents an exploration rate specification. There are two parameters for the exploration rate specification:

1. ϵ - A value between 0 and 1 which controls the probability of selection an action at random.

2. ϵ **Decay** - A Boolean value which specifies whether or not ϵ will decay over time.

    - When **True**, ϵ will decay by $\frac{1}{2}$ at the end of each year.
    - When **False**, ϵ will not decay and retain the initial value throughout learning.

5.2 Learning Rate and Discount Factor

The learning rate and discount factor are set to 0.9 and 0.8 respectively.

5.3 Utility Scheme

There are three distinct actions that Agents can take in this experiment:

1. Buy
2. Sell
3. Do Nothing

A standard trading fee on financial markets is 0.1% of the order value. This applies to buying and selling. With buying, the effective portfolio balance is reduced by 0.1%, if the entire balance is used to trade. When selling, profit can be measured in a percentage amount, including the trading fee. To emulate a live environment, Agents will receive the following rewards for buying and selling:

- Buying: -0.001
- Selling: [(sell price - purchase price) / purchase price] - 0.001

It does not cost anything to simply take no action on a financial market. There are exceptions to this, but the goal of this experiment is not to emulate every exchange intricacy. In this study, to encourage Agents to be active traders, a decision is made to penalise the Agents for doing nothing. Hence a reward of -0.0005 is selected as the reward for choosing to do nothing. This amount is exactly half of the fee and while it is not a significant amount, it will provide stimulation in certain situations to buy or sell.

5.4 Experiments

Experiments are categorised by the research question they aim to address. Reference is made to 'training' and 'testing' in each of the experiments. Descriptions of said processes are provided below as template experimentation steps that may be reused in several experiments.

5.4.1 Experimentation Patterns

5.4.1.1 Training

In the context of this experiment, training involves assigning Agents to a financial market environment where they learn how to trade. As there are 6 exploration rate specifications (Table 5.1) and 510 possible state-space combinations, a total of 3,060 Agents are trained for any experiment that requires training. This number is further increased by the number of financial markets involved in experimentation. Specifically the total number of trained Agents, for all financial markets in training scope is $\|\text{financial markets}\| \times 3,060$.

During training, at the end of each trading day, each training experiment will record, for each market in scope, data for the following attributes:

- Date
- Exploration Rate Specification ID
- Cumulative Reward
- Current Balance

5.4.1.2 Testing

Resultant Q-Tables from training experimentation are considered policies for testing. It is important to note that each training exploration rate specification produces a set of unique policies, i.e. one policy per Agent. Therefore, the total number of policies is: $\|\text{exploration rate specifications}\| \times \|\text{agents per specification}\|$, or $6 \times 510 = 3,060$ policies. This number, as in training, is increased as the number of financial markets to be tested, increases.

Critically, exploration and learning are disabled for testing. Agents exploit prior learning at every state. This is the basis for analysing the generalisation qualities of policies on unseen markets.

5.4.1.2.1 Testing Experiment Description

For each of the 6 exploration rate specification groups and their 510 policies, a testing experiment is executed on previously unused data for each of the three markets. Therefore, 18 tests in total are performed. Agents read the state at each timestep, and lookup the action to take from the policy. The Agent moves to the next step and a reward is received. This is repeated until the Agent reaches the terminal state. Similar to training, at the end of each day in the specified time-period during testing, each testing experiment, for each market, will record data for the following attributes:

- Date
- Learned from Exploration Rate Specification ID
- Cumulative Reward
- Current Balance

A time-series plot for each *Learned from Exploration Rate Specification ID* is created for analysis, with the Current Balance on the y-axis. As a result, it is possible to visualise the effect of exploration rate variation on performance on unseen data across different markets.

5.4.2 Standalone Experimentation

5.4.2.1 Exploration Rate Specification Difference

RQ1 pertains to the effect that exploration rate variation during training has on Agent performance on unseen data. An experiment consisting of one training and testing procedure, as defined in the previous section, for each financial market. Results are analysed and interpreted for signs of exploration rate variation effect on testing results. Following this step, two significance tests are required to address the core research question. The first test is an ANOVA test and will compare the performance across all 6 exploration rate specifications to see if at least any two means are significantly different to one another. The second test is a T-test will check for a significant differences between two selected means, if the results from the first test are significant.

5.4.2.1.0.1 ANOVA Test

To facilitate this test, it is necessary to produce a sample of means for each exploration rate configuration for a selected market. The sample means are used in an ANOVA test to determine if a significance difference between any of the means exists. Since learned policies will produce the same results on unseen data for each episode, it is necessary to execute, for each market-learning rate specification combination, a series of training-testing episodes. This will provide samples with variation required to carry out the ANOVA test. Bitcoin is selected as the market to carry out the ANOVA test. With 6 exploration rate specifications and a sample size of 30, a total of 180 training-testing episodes are required.

At the end of each training step, exploration and learning are disabled as before. Agents treat the Q-tables as policies during testing. At the end of each episode, the average Agent closing balance per market-exploration rate specification is recorded. The means are then used into the ANOVA test for significance.

A p-value of 0.05 will be used for the ANOVA test.

5.4.2.1.0.2 T-test

If the p-value from the ANOVA test is less than 0.05, a T-test will be carried out. The T-test will check for significant differences between two selected means. A one-sided T-test will be done if the mean Agent performance of one exploration rate specification appears to be greater than another in a single episode. If there is so clear difference in the mean Agent performance of two exploration rate specifications, a two-sided T-test will be performed.

A p-value of 0.05 will be used for the T-test.

5.4.2.2 Effect of Constrained Reinforcement Learning

RQ2 pertains to investigating the effect of adding RL constraints has on algorithmic trading profitability in Q-Learning environment. To explore this question, a significance test is required. The test consumes sample means from both the constrained and unconstrained side and determine if the difference between them, if one exists, is significant.

5.4.2.2.1 T-test

The T-test will check for significant differences between the mean Agent performance of the same exploration rate specification on the same market, under constrained and unconstrained environments. In other words, sample means from both sides will be generated and consumed by the T-test to check for significance. For the same reasons outlined for the previous experiment, Bitcoin-USD is selected as the market used in this experiment. To prepare the data for experimentation, just like the ANOVA test in the previous experiment, 30 training-testing episodes for the Bitcoin market are carried out, and the mean Agent closing balance recorded. This is done with constraints turned on, and turned off. If one side appears to be generating higher means than the other during experimentation, a one-sided T-test will be carried out. Otherwise a two-sided T-test will be executed.

A p-value of 0.05 will be used for the T-test.

5.4.2.3 Benchmark Testing

RQ3 is concerned with determining if policies learned through Reinforcement Learning are capable of outperforming a buy and hold strategy. In this experiment, the best-performing Agents from the testing phase (i.e. policies) are selected and their results are compared with a buy and hold strategy. The buy and hold strategy will spend its entire opening balance on the first trading period and hold its position until the final period. Closing portfolio values will be compared after execution. In particular, the following tests will be performed in this experiment:

1. Best Constrained Policy vs Buy and Hold (BTC-USD)
2. Best Unconstrained Policy vs Buy and Hold (BTC-USD)
3. Best Constrained Policy vs Buy and Hold (EOS-USD)
4. Best Unconstrained Policy vs Buy and Hold (EOS-USD)
5. Best Constrained Policy vs Buy and Hold (NASDAQ-USD)
6. Best Unconstrained Policy vs Buy and Hold (NASDAQ-USD)

The experiment will consider the results of the above tests at face value, with analysis and interpretation also taking the constrained vs unconstrained differences into account. It is assumed that through the selection of the best performing Agent out out of over 3,000 Agents, any closing balance difference between that of the Agent and the buy and hold strategy will hold up during significance testing. Therefore, significance testing is not performed in this experiment.

Chapter 6

Results

At the core of this study is the effect of constrained RL in an algorithmic trading scenario. Accordingly, results are categorised by constrained and unconstrained RL approaches. References to and comparison with the constrained approach are made in the analysis and interpretation of the unconstrained RL results.

6.1 Training

6.1.1 Constrained Reinforcement Learning

6.1.1.1 Results

Figure 6.1 shows the results of Reinforcement Learning under 6 distinct exploration rate specifications. Actions were limited by constrained RL. For each specification, 510 Agents were created and assigned a unique state-space formulation. Every line on the plots represents the performance of a single Agent. Also highlighted are the best performing Agent for each exploration rate specification, as well as the mean performance, i.e. the average Agent balance per timestep.

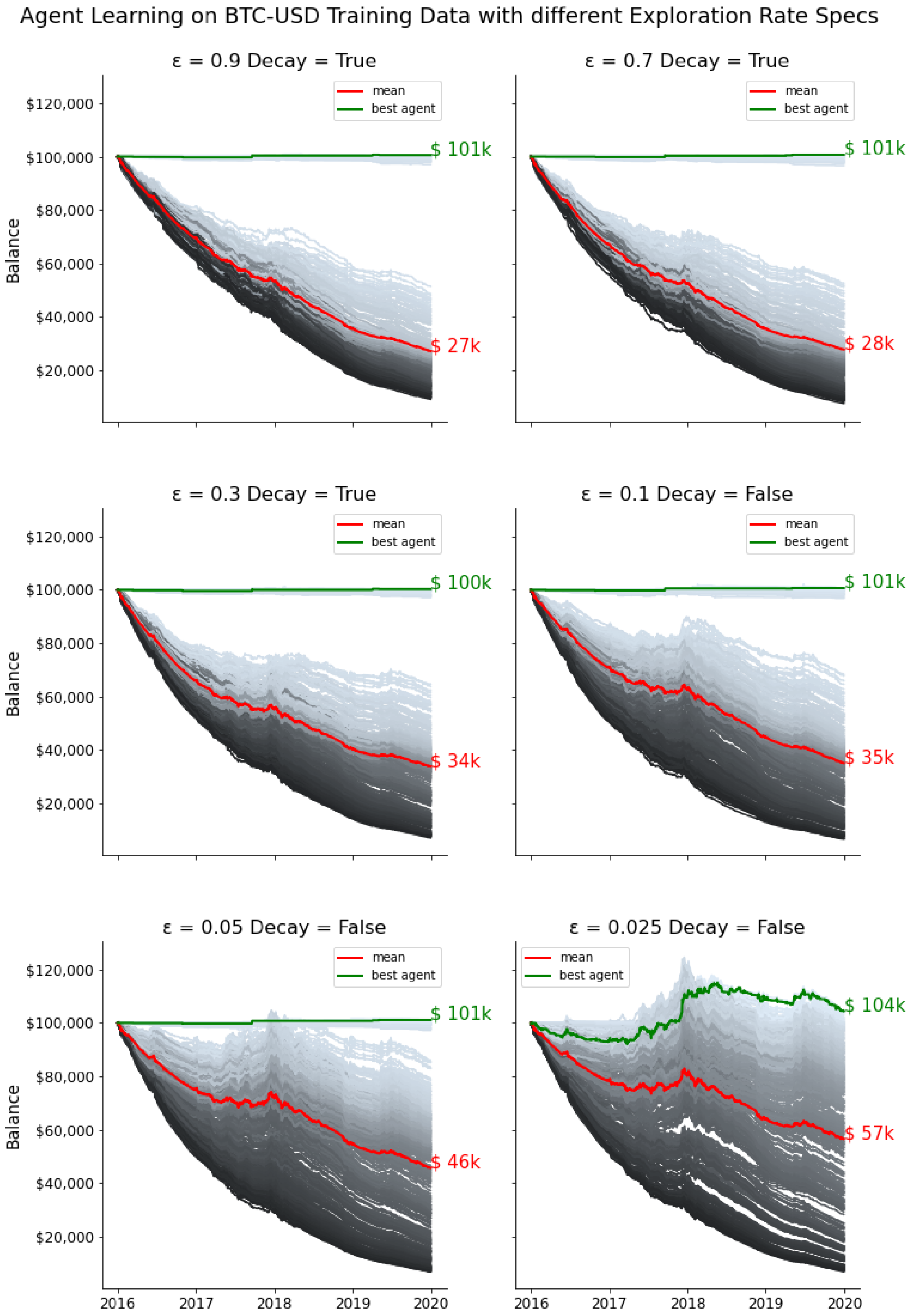

Figure 6.1: Constrained RL Training Performance Results

6.1.1.2 Analysis

As the initial exploration rate is reduced in the constrained RL approach, a slight improvement in performance is achieved. Specifically, there is a gradual improvement in the mean Agent-closing balance. This is evident in Figure 6.1. Plots are read from left to right, then top to bottom. The mean closing balance increased from \$27,000 in specification 1 (top-left plot), to \$57,000 in specification 6 (bottom-right plot). There is also an increasing separation, or variance, in the plots, starting from the top-left plot ($\epsilon = 0.9$, decay = True) and working towards the bottom-right plot ($\epsilon = 0.025$, decay = False). Notably, there is little variance in the best-performing Agents under each exploration rate specification. The closing balance of the majority of the best-performing Agents is only \$101k, i.e. a 1% increase on the opening balance. Interestingly, the best-performing Agents, with the exception of pertaining to the final plot, have very variance in their trends. It appears that very few actions were taken, yet the Agents still managed to make a profit. This is promising, considering that only 1% of the portfolio balance could be traded at a time.

Regardless of the exploration rate or decay thereof, a band of very poor-performing Agents exists in each training run. These are identified by a sharp and continuous drop-off in balance in each plot.

6.1.1.3 Interpretation

Albeit Figure 6.1 exhibits variation in mean Agent performance across the exploration rate specifications, it is of importance to highlight that this does imply superiority of one exploration rate specification setting over another. Adjusting ϵ simply controls the probability of selecting a random action. Hence, the variation in depicted mean trends is predominately due to variation in epsilon and says nothing about generalisation qualities. There is an open research question as to whether those Agents which were subjected to a higher number of random Actions may ultimately achieve a more profitable policy. Therefore, before reaching any conclusion, the Agents pertaining to each exploration rate specification run must be subjected to performance analysis on previously unused data.

6.1.2 Unconstrained Reinforcement Learning

6.1.2.1 Results

Figure 6.2 shows the results of Reinforcement Learning under 6 distinct exploration rate specifications. No limitations on action selection were applied as this was an unconstrained RL experiment. For each specification, 510 Agents were created and assigned a unique state-space formulation. Every line on the plots represents the performance of a single Agent. Also highlighted are the best performing Agents for each exploration rate specification, as well as the mean performance, i.e. the average Agent balance per timestep.

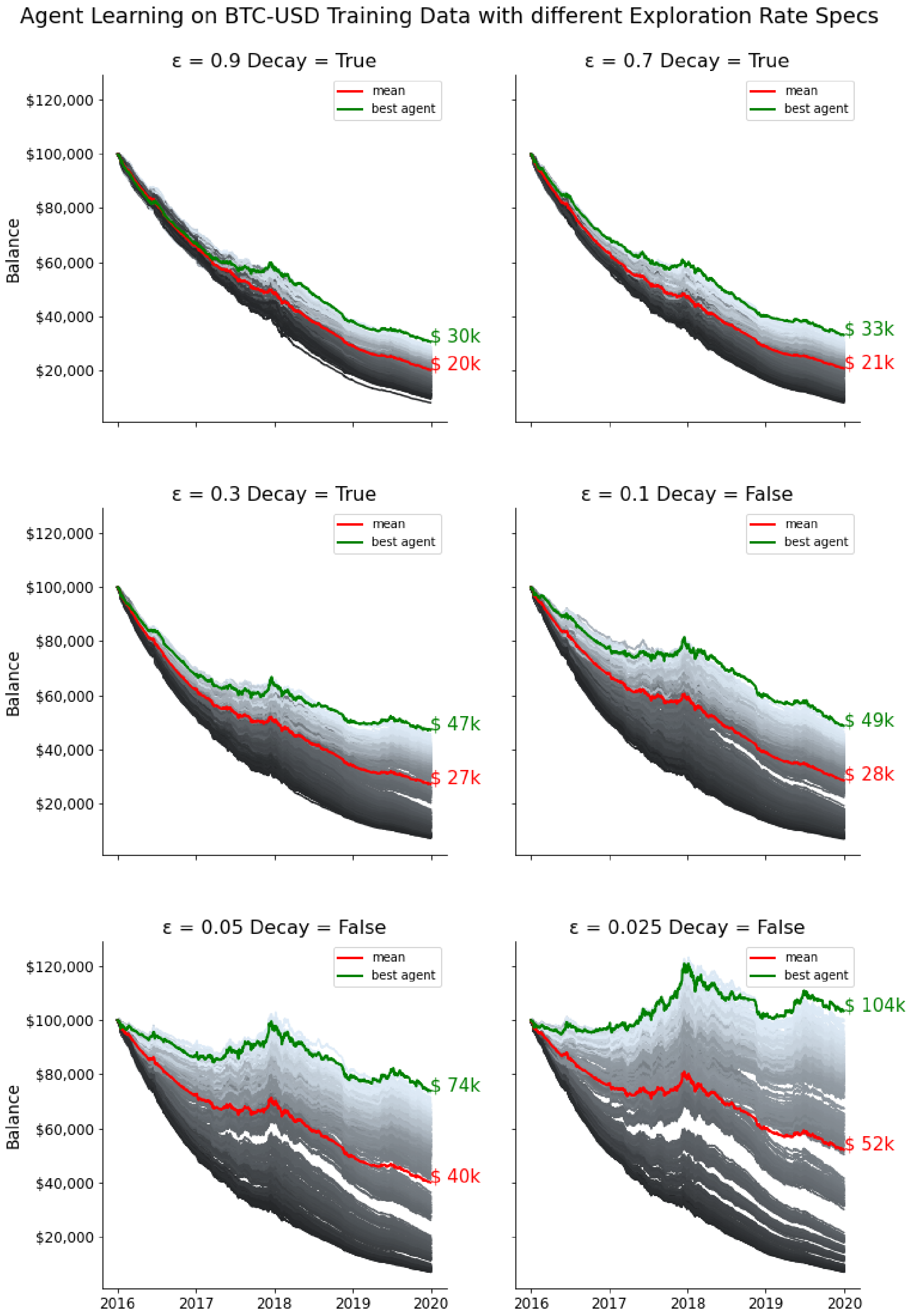

Figure 6.2: Unconstrained RL Training Performance Results

6.1.2.2 Analysis

Similar to the constrained RL results, there exists a trend in terms of the mean Agent performance as the initial exploration rate is reduced. The mean closing balance increased from $20,000 in specification 1 (top-left plot), to $52,000 in specification 6 (bottom-right plot). However, the mean closing balance for each exploration rate specification in the constrained RL approach is greater that the mean of the corresponding exploration rate specification in the unconstrained RL approach. Further testing and analysis are required to check for significance.

In terms of the best-performing Agents, results from the unconstrained RL differ to the constrained RL results. In Figure 6.2, there appears to be a correlation between reduction in exploration rate versus an increase in closing balance of the best-performing Agent. To be precise, starting from the top-left plot ($\epsilon =$ 0.9, decay = True) and working towards the bottom-right plot ($\epsilon = 0.025$, decay = False), the closing balance of the best performing Agent increased in each subsequent setting. The first exploration rate specification resulted in a best-performing Agent closing balance of $30,000. A closing balance of $104,000 was achieved by the best-performing Agent of specification 6, i.e. a 247% increase. Although this is a substantial difference, it does not indicate that Agents trained as per specification 6 will generalise better than Agents trained with different exploration rate settings.

6.1.2.3 Interpretation

Removing constraints from the possible actions at specific states appears to have caused a reduction in mean and best-Agent training performance. This is likely due to Agents, especially those with a high exploration rate, taking undesirable actions which would otherwise have been constrained, or restricted. This is clear from the first four plots in Figure 6.2. None of the unconstrained RL training plots exhibit best-Agent behaviour like their counterparts in the constrained RL plots. In particular, the constraint-bound best-performing Agents exhibit a near-horizontal trend, indicative of not holding a position for the majority of the training period. The horizontal trends also indicate that said Agents placed a higher value on doing nothing in the majority of states. The fact that the unconstrained RL Agents did not exhibit this behaviour might suggest that the addition of constraints may have been the cause of the low-trading, yet profitable activity of the constrained RL Agents.

6.2 Testing

6.2.1 Constrained Reinforcement Learning

6.2.1.1 Results

6.2.1.1.1 Performance on Holdout Data

This section contains constrained RL results of trading on previously unused data for each of the three markets in scope. All 510 Agents from each of the 6 exploration rate settings in training were allowed to trade. However, unlike training, the Agents were not allowed to learn from or explore the the testing environments. In other words, the results Q_tables from training are used as policies for testing. The testing time-period is 01/01/2020 to 31/12/2021. Another major difference in experimental configuration between training and testing, is that in testing, the Agents are allowed to use all of their available balance for trading. Figures 6.3, 6.4 and 6.5 depict policy performance on Bitcoin, EOS and NASDAQ holdout data, respectively, and for each of the exploration rate specifications used during training.

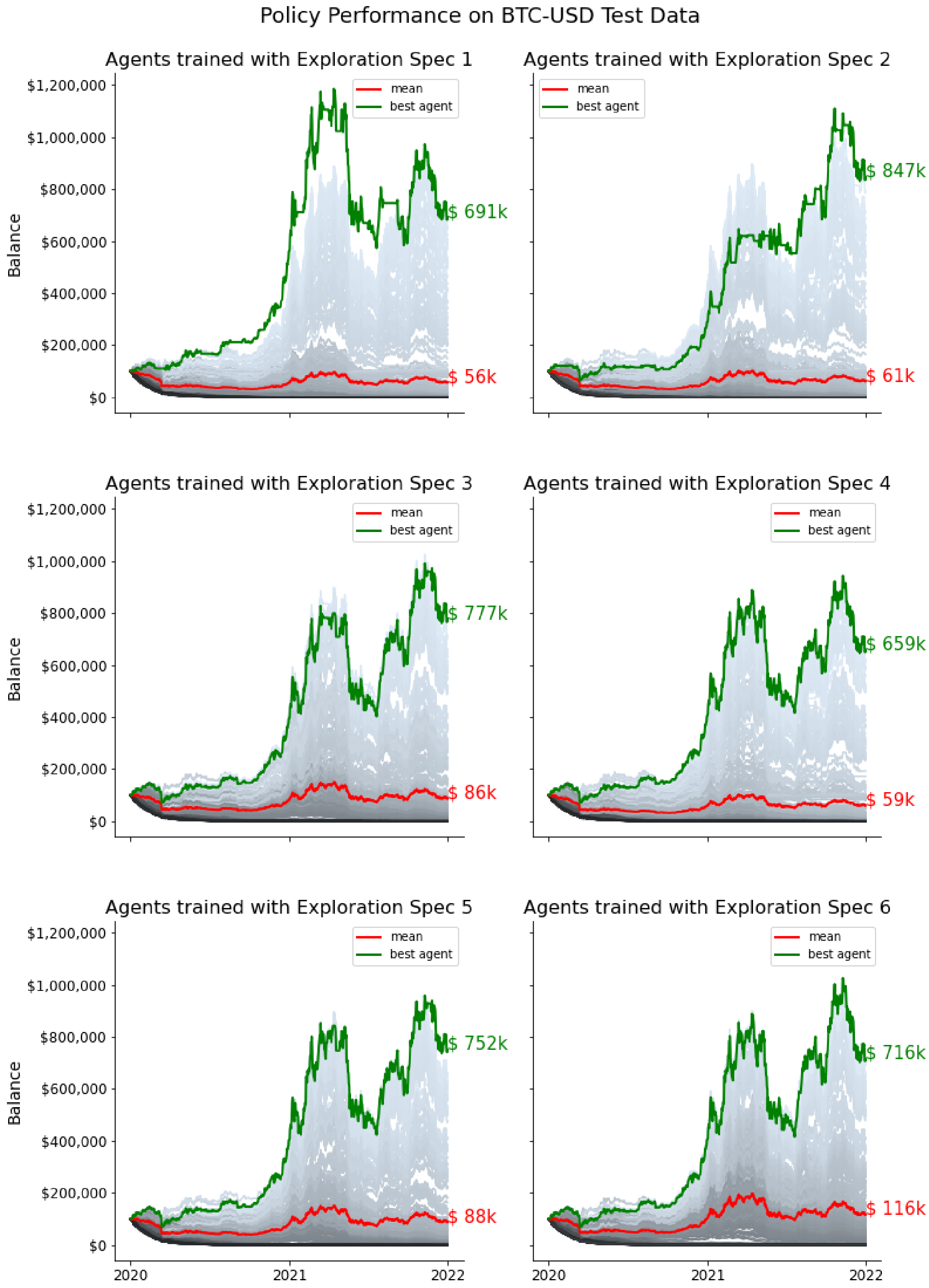

Figure 6.3: Policy Performance on Bitcoin Holdout Data

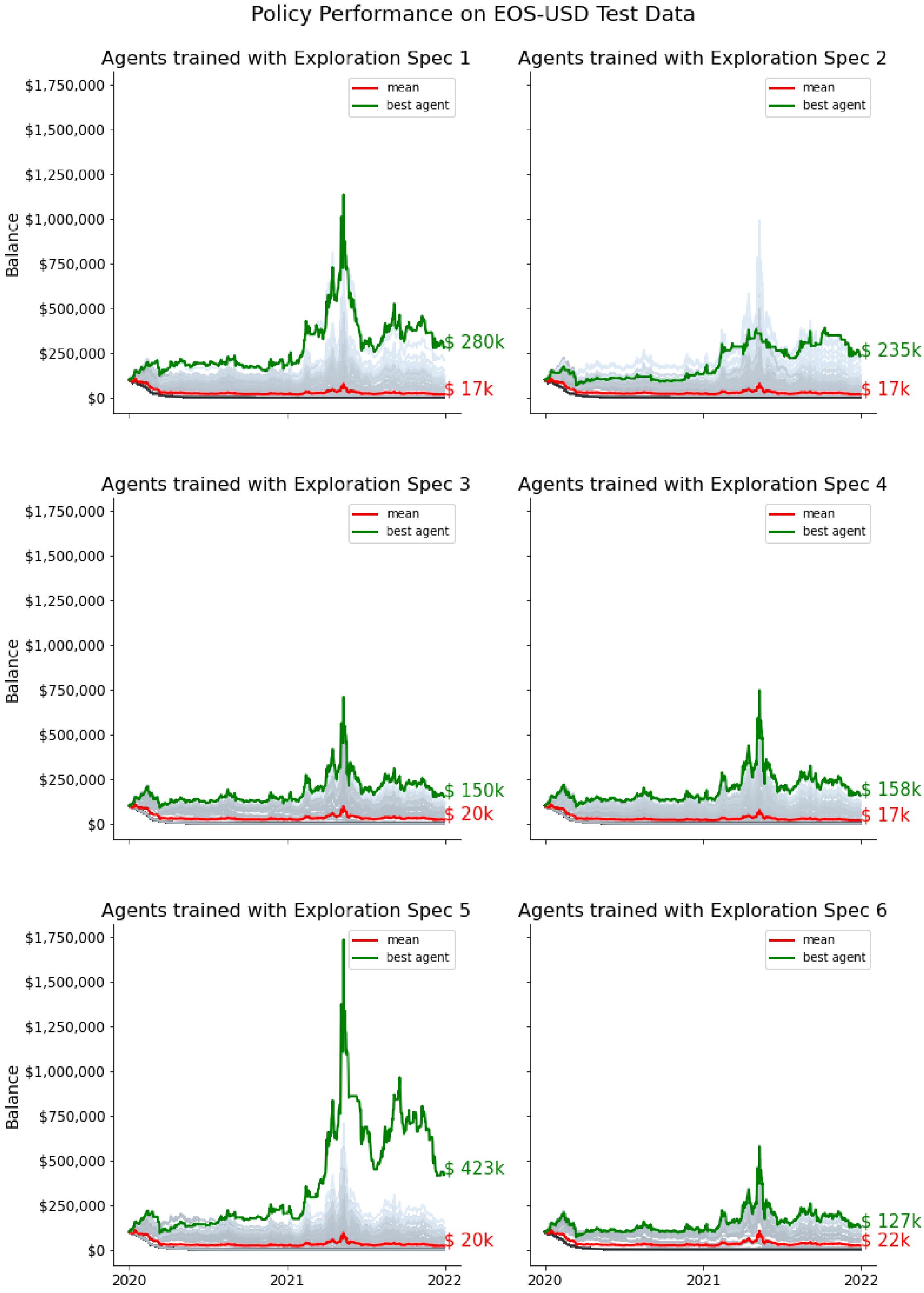

Figure 6.4: Policy Performance on EOS Holdout Data

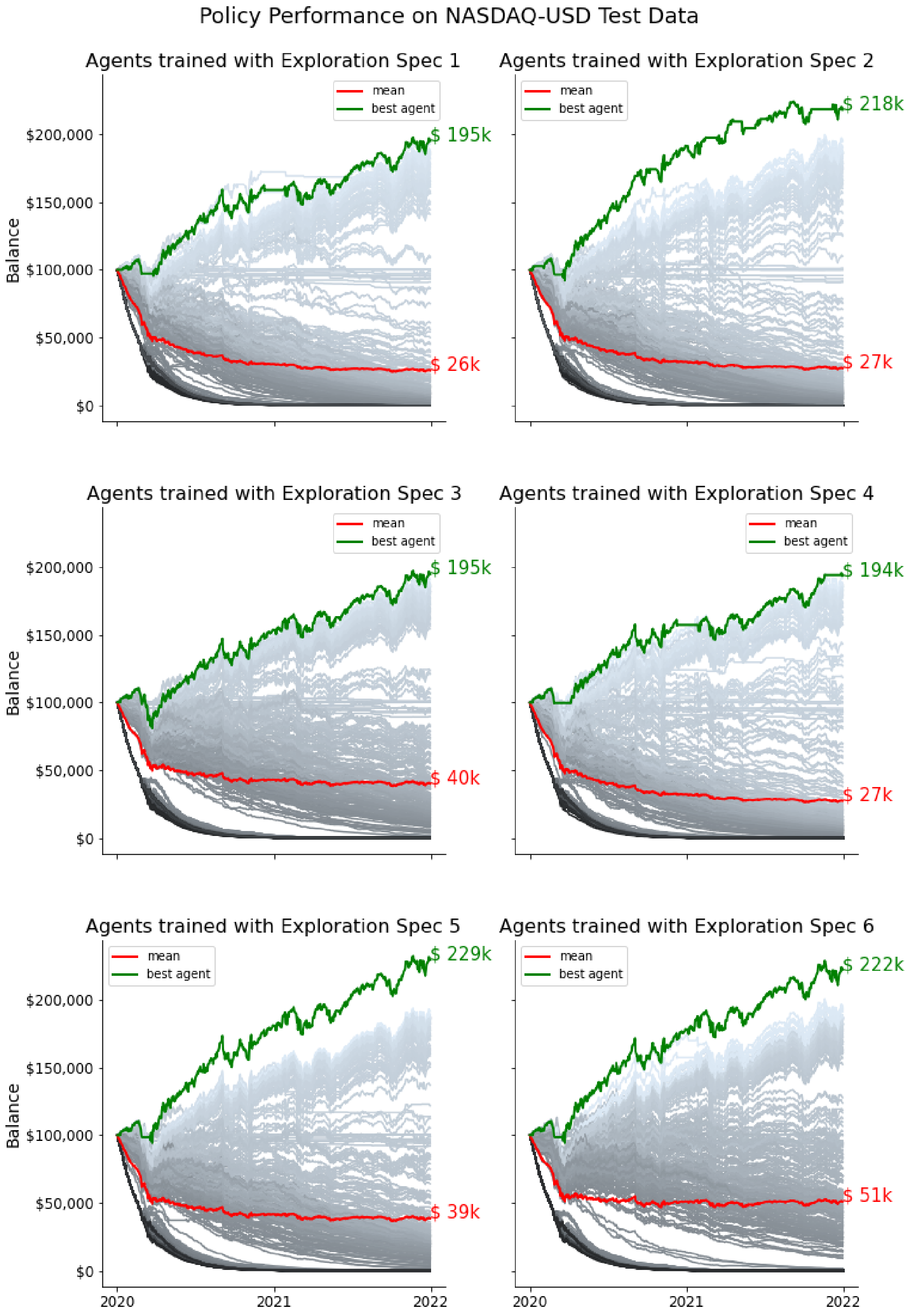

Figure 6.5: Policy Performance on NASDAQ Holdout Data

6.2.1.1.2 Mean Performance on Holdout Data - Detailed

Figures 6.6, 6.7 and 6.8 provide an intuitive, market-specific comparison between the mean Agent balance per timestep for each group (training specification) of Agents, under a constrained RL approach.

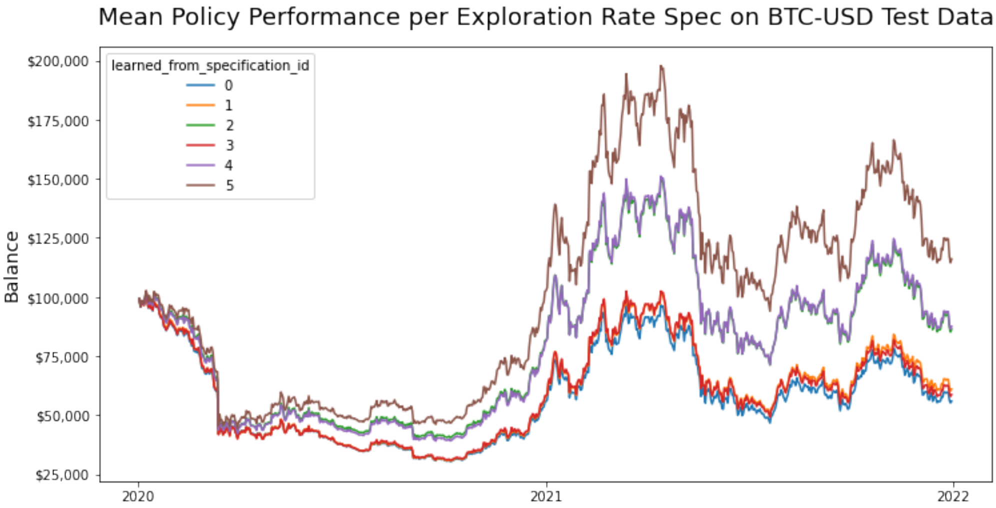

Figure 6.6: Mean Agent Performance on Bitcoin Holdout Data

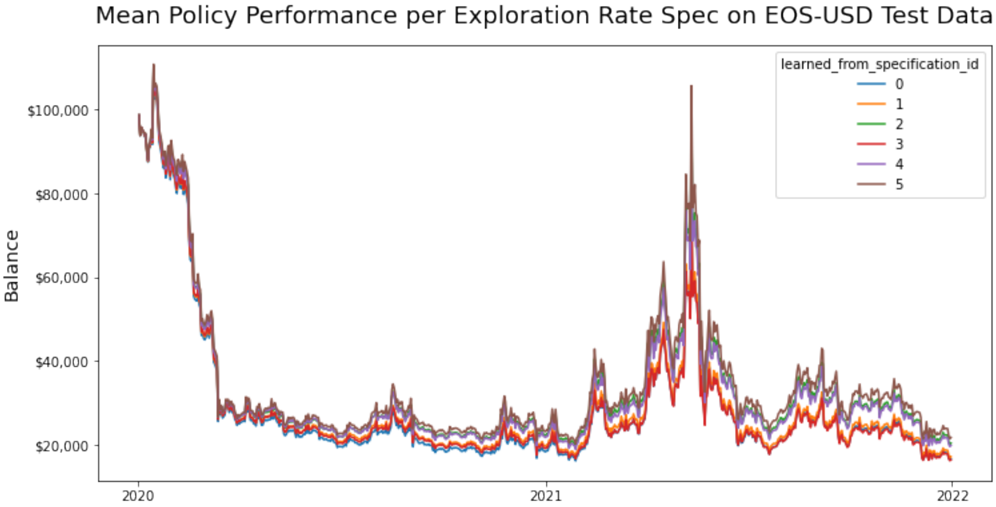

Figure 6.7: Mean Agent Performance on EOS Holdout Data

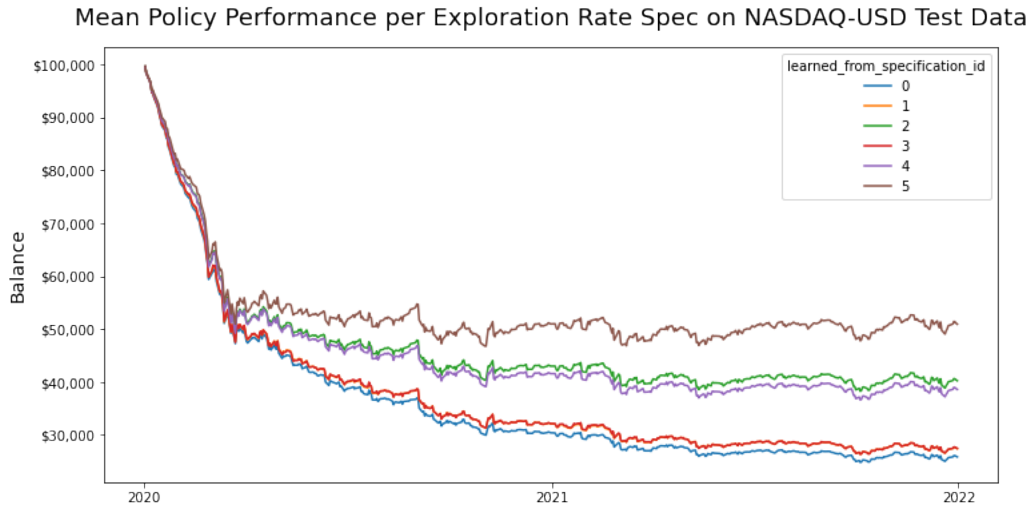

Figure 6.8: Mean Agent Performance on NASDAQ Holdout Data

6.2.1.1.2.1 ANOVA Test

Summary statistics for 30 training-test runs on BTC-USD data for each exploration rate specification are shown in table 6.6.

Exploration Rate Spec. ID	N	Mean	SD	SE	95% Conf.	Interval
1	30	51,907	3,931	718	50,439	53,375
2	30	63,152	3,715	678	617,65	64,540
3	30	94,207	3,588	655	92,867	95,547
4	30	57,946	3,390	619	56,680	59,212
5	30	85,192	3,475	635	83,895	86,490
6	30	115,307	4,523	826	113,618	116,996

Table 6.1: Mean Closing Balance Summary Statistics - Constrained RL

Table 6.7 shows the results of a one-way ANOVA test for significance between closing-balance means per exploration rate specification from training-test runs on the BTC-USD market.

Hypothesis

- H_0**:** All mean closing balance are equal.
- H_A**:** All mean closing balance are not equal.

F-statistic	p-value
1257.5	1.6e-134

Table 6.2: One-Way ANOVA Test Result

6.2.1.1.2.2 One-Sided T-test

One-sided T-test results for the difference between the average balances for exploration rate specifications 6 and 3 is shown in Table 6.8.

Hypothesis

- H_0**:** Mean closing balance for exploration rate specification 6 is not greater than the mean closing balance for specification 3.
- H_A**:** Mean closing balance for exploration rate specification 6 is greater than the mean closing balance for exploration rate specification 3.

t-statistic	p-value
20.1	5e-28

Table 6.3: One-Sided T-test Results - Constrained RL

6.2.1.2 Analysis

6.2.1.2.1 Overall Performance

6.2.1.2.1.1 Bitcoin

In reference to Figure 6.3, there is variation between mean and best Agent testing performance across the exploration rate specification groupings. Agents trained with Exploration Rate specification 6, i.e. with the lowest exploration rate, generalised the best, on average, on previously unused BTC-USD data. The mean Agent closing balance for this specification is $116,000. This is 32% greater than the next highest mean balance of $88,000 from exploration rate specification 5. A statistical significance test is needed to determine if this difference in significant.

When considering only the best Agent from each plot, the best performing Agent overall belongs to plot 2, i.e. trained with exploration rate of 0.7 with decay. This Agent achieved a final closing balance of $847,000 which equates to a 13% increase over the next best-performing Agent that finished with a closing balance of $752,000. Looking at Figure 6.3, it is clear that the best-performing Agent from exploration rate specification 2 has a higher presence of horizontal lines in the trend of its balance. The significance of this is discussed in the interpretation.

Figure 6.6 shows, more intuitively than Figure 6.3, the comparison between mean Agent testing performance for all exploration rate specifications applied during training. It appears that small differences between the trends near the start of the testing period ultimately led to large differences by testing completion. Note that specification IDs are zero-indexed in Figure 6.3.

6.2.1.2.1.2 EOS

Results from the EOS-USD testing phase show that again, exploration rate specification 6 led to the highest mean Agent closing balance during on the EOS-USD market. This is evident in Figure 6.4. A mean Agent closing balance of \$22,000 was achieved which is 10% greater than the mean Agent closing balance of \$20,000, achieved from both specifications 5. This is the same placement as from the BTC-USD results. A trend beings to emerge here and reinforces the need to carry out a statistical significance test.

The best performing Agent overall was trained with exploration rate specification 5 and achieved a closing balance of \$423,000. This is much greater (+51%) than the next best performing Agent (specification 1), which achieved a closing balance of \$280,000.

Figure 6.7 compares the mean Agent performances trends for the EOS-USD testing phase. Although there visible separation in the trends, it much less evident than in the corresponding BTC-USD plot. Indeed it is already known from Figure 6.4 that the difference between the worst and best mean Agent performance is \$22,000 - \$17,000 = \$5,000. In other words, The mean Agent performance from specification 6 is 29% greater than that of specification 3. This pales in comparison to the same lowest to highest mean Agent performance difference from the BTC-USD testing phase, which is \$116,000 - \$56,000 = \$60,000 (+107%).

6.2.1.2.1.3 NASDAQ

Figure 6.5 shows the constrained RL results for the NASDAQ-USD market. The results exhibit the same trend in mean Agent performance in that Agents trained with exploration rate specification 6, on average, appeared to have performed better in testing that Agents trained with alternative exploration rate specifications. This specification resulted in a mean Agent closing balance of \$51,000. Again, specification 5 finished in second place with a mean Agent closing balance of \$39,000. This equates to 31% increase from specification 5 to 6.

The best-performing Agent overall arose from exploration rate specification 5. It achieved a final closing balance of \$229,000 which represents a 3% increase over specification 5, with a best-performing Agent closing balance of \$222,000.

Figure 6.8 compares the mean Agent performances trends for the NASDAQ-USD testing phase. There is huge separation between specification 6 (zero-indexed as id 5 in plot) and specifications 3 and 5. Trends for specifications 1, 2 and 4 are similar, and together they forming a losing pack from circa April 2020.

6.2.1.2.2 Statistical Analysis

There is a clear pattern of policy performance on holdout data for the three markets. Specifications 6 and 5 consistently finished in the top two positions in terms of highest Agent mean closing balance. To further investigate these findings, a significance test was carried out. It is assumed, given the initial analysis, that significant differences detected in performance on the BTC-USD market would be repeated in the other two markets. As such, a significance test is only carried out on the BTC-USD market.

For each exploration rate specification, a training phase, followed by a testing phase was carried out. The Q-tables generated from training were used as policies for testing. The mean Agent closing balance per exploration rate specification was recorded at the end of testing. This entire process was repeated 30 times for each exploration rate specification. Table 6.6 shows the summary statistics from this step.

A One-Way ANOVA test for a significant difference between any pair of mean Agent closing balances is carried out. The Null Hypothesis states that no significant difference exists between any of the means. The alternative hypothesis states that at least one difference exists. Table 6.7 shows the results of the ANOVA test. The F-statistic is very high at 1,257.5, which denotes very high variation between sample means versus within-sample variation. The p-value of 1.6e-0134 is much less than 0.05 and therefore the null hypothesis that there is no difference between the means is rejected.

Considering exploration rate specifications 6 and 5 led to, average, the most performant Agents on Bitcoin holdout data, yet their closing balances differed by 32%, a logical step is to perform a one-sided T-test between their sample means. However, examining Table 6.6., the training-test sample statistics, it transpires that the mean of means for specification 3 finished ahead of that of specification 5. Therefore, specification 3 is selected as the second element in the T-test.

The ANOVA test already detected a significant difference between *one* pair of samples. If a significant difference is detected between results for exploration rate specifications 6 and 3, it can be deduced that the means for exploration rate specifications 1, 2, 4 and 5 are also significantly different than that of exploration rate specification 6. The Null Hypothesis states that mean Agent performance on holdout data for specification 6 is not greater than mean Agent performance on holdout data for specification 3. The alternative hypothesis states that mean Agent performance on holdout data for specification 6 is greater than mean Agent performance on holdout data for specification 3. A one-sided t-test produced a p-value of 5e-28 which is much less than the accepted level of 0.05. This means that the null hypothesis, i.e. that mean Agent performance on holdout data for specification 6 is not greater than mean Agent performance on holdout data for specification 3, is rejected.

6.2.1.3 Interpretation

6.2.1.3.1 RQ1 - Exploration Rate Variation Effect

Exploration rate specifications 6 ($\epsilon = 0.025$ with decay) and 5 ($\epsilon = 0.05$ without decay) performed the best during testing on holdout data for each of the three markets. Both of these exploration rates are lower than the other specifications, with the exception of specification 4, which after decay, provided an exploration rate of 0.0375 to the fourth year of training.

There is an argument to be made that these low values of ϵ for four years of training contributed, at least in part, to the performance of the Agents pertaining to these exploration rate specifications, over the other four. Indeed, when considering the amount of training data available (hourly intervals for four years), it is conceivable that after assimilating one to two years of data, Agents have enough experience to start comfortably exploiting knowledge. Therefore, Agents that learned with exploration rate specifications 6 and 5 spent more time exploiting knowledge than Agents from the other four specifications. Critically, the near-avoidance of taking random actions in latter stages of training, mitigated the risk of adversely affecting Q-values when the optimal actions were already learned. It is therefore plausible that Agents from exploration rate specifications 1 to 4 suffered from this exact phenomenon.

6.2.1.3.1.1 Statistical Significance

To further quantify the differences in mean performance, reference is made to the ANOVA test and T-test results as documented in Section 6.2.1.1.2. The ANOVA showed that at least one mean is statistically different to another. However, a 30-sample mean Agent closing balance for each exploration rate specification showed, on average, that exploration rate specification 3 performed better than exploration rate specification 5. Regardless, this study is not concerned with the best exploration rate specification, but rather that variation in exploration rate can have a statistically significant impact on testing performance. A one-sided T-test showed that the mean Agent closing balance for exploration rate specification 6 on the BTC-USD market is greater than that of exploration rate specification 3, from a statistical significance standpoint. This is perhaps not surprising, given the difference of 22% in their mean of means as seen in Table 6.6. Indeed, we can further deduce from this finding that exploration rate specifications 1, 2, 4 and 5 are also significantly less than exploration rate specification 6 in terms of mean Agent closing balance.

It is reasonable to conclude, considering the statistically significant differences shown by the ANOVA and T-test, that variation in exploration rate during training has an affect on profitability when applying learned policies to unseen data.

6.2.2 Unconstrained Reinforcement Learning

This section contains experimental results of trading performance after RL constraints were removed. While Research Question 1 deals with the effect of exploration rate variation during training has on generalisation, Research Question 2 deals with the effect of adding constraints to the RL state-action space. As such, results, analysis and interpretation in this section are performed to determine differences, if any, between the constrained and unconstrained approaches. Therefore, considerations of the affects of exploration rate variation in an unconstrained scenario are out of scope in this section. The use of Figures and Tables are in accordance. Aside from the removal of RL constraints, all other experimental settings in comparison to the constrained approach are the same.

6.2.2.1 Results

6.2.2.1.1 Performance on Holdout Data

Figures 6.9, 6.10 and 6.11 depict policy performance on Bitcoin, EOS and NASDAQ holdout data, respectively, and for each of the exploration rate specifications used during training.

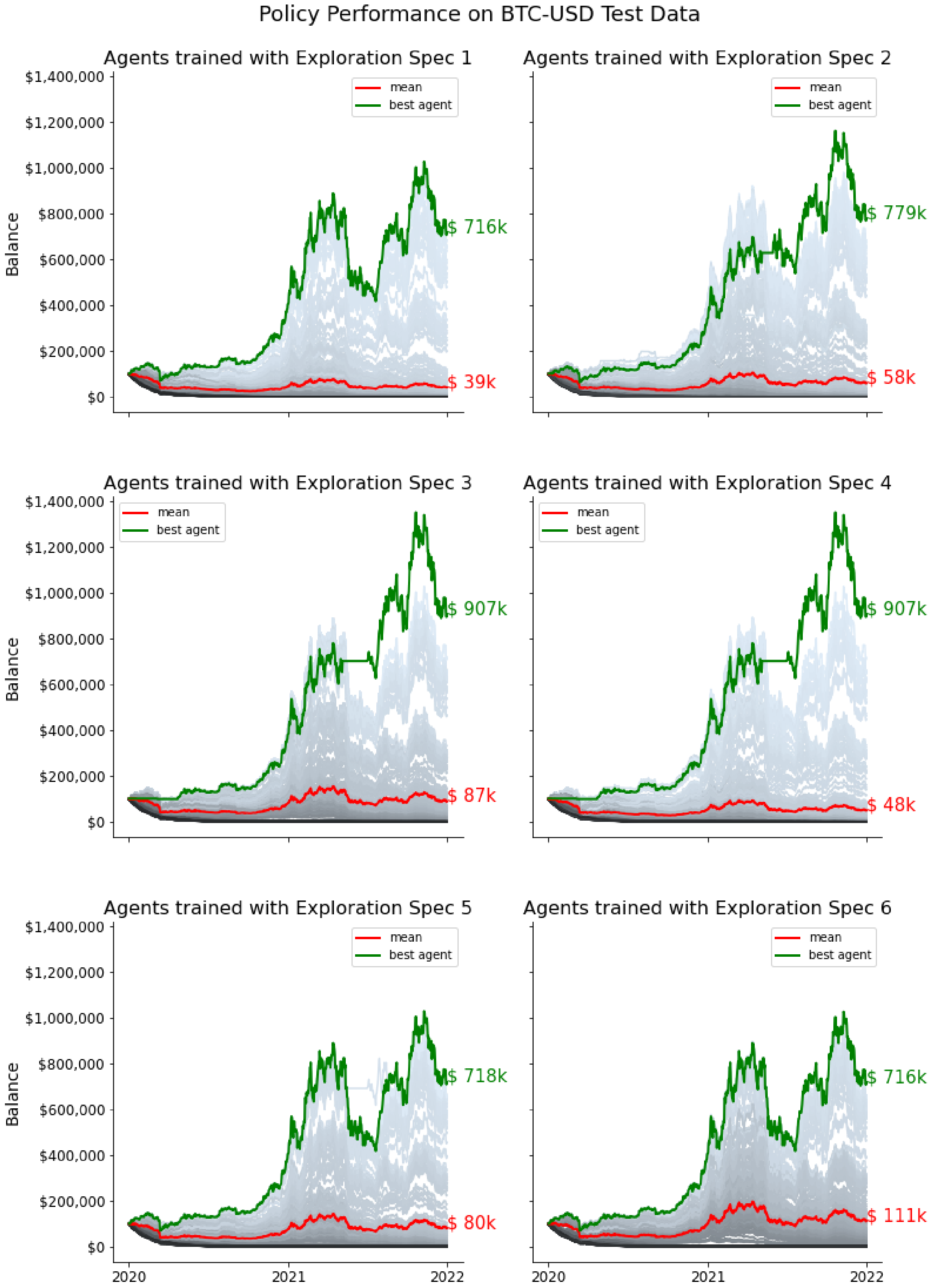

Figure 6.9: Policy Performance on Bitcoin Holdout Data

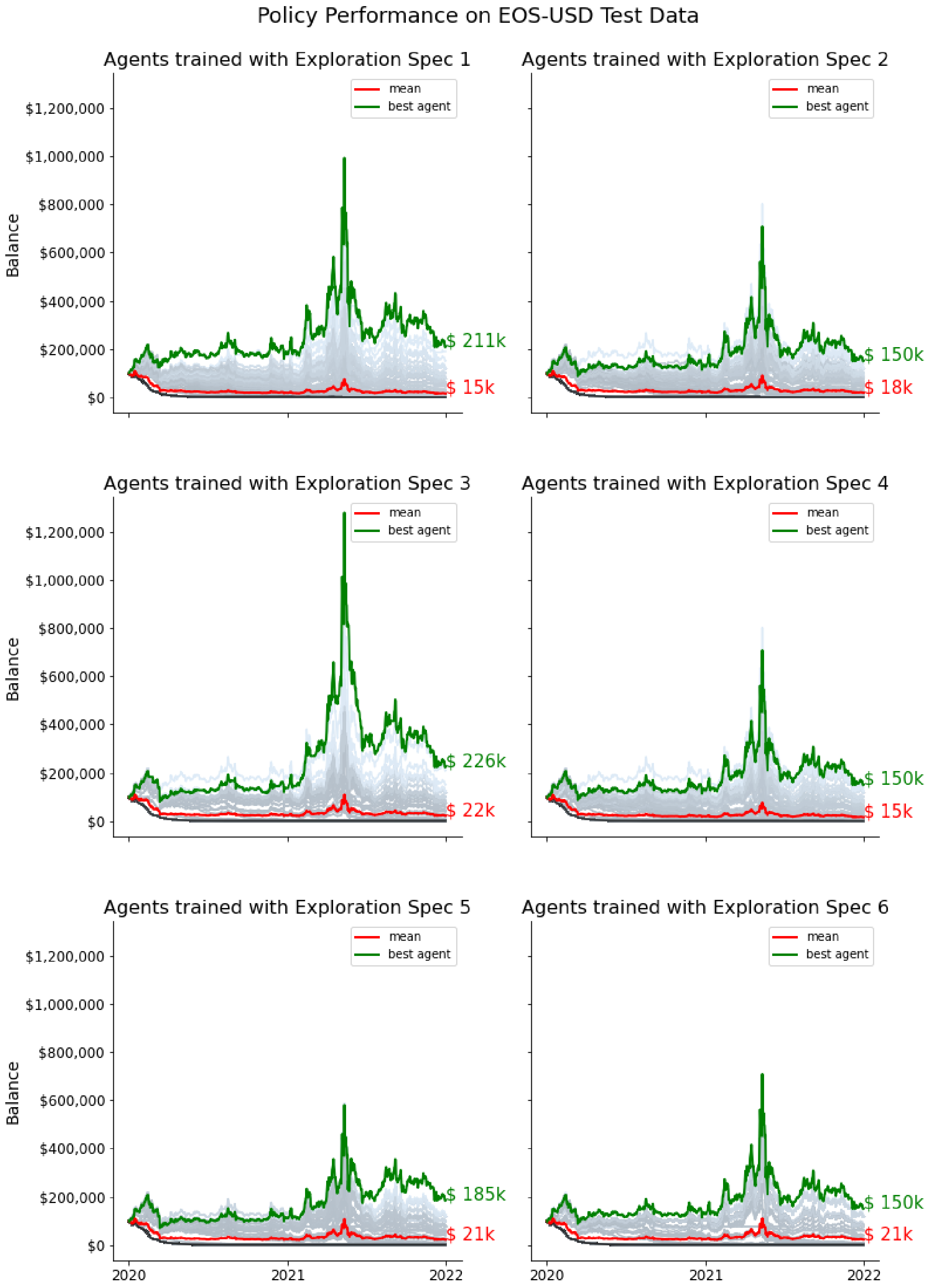

Figure 6.10: Policy Performance on EOS Holdout Data

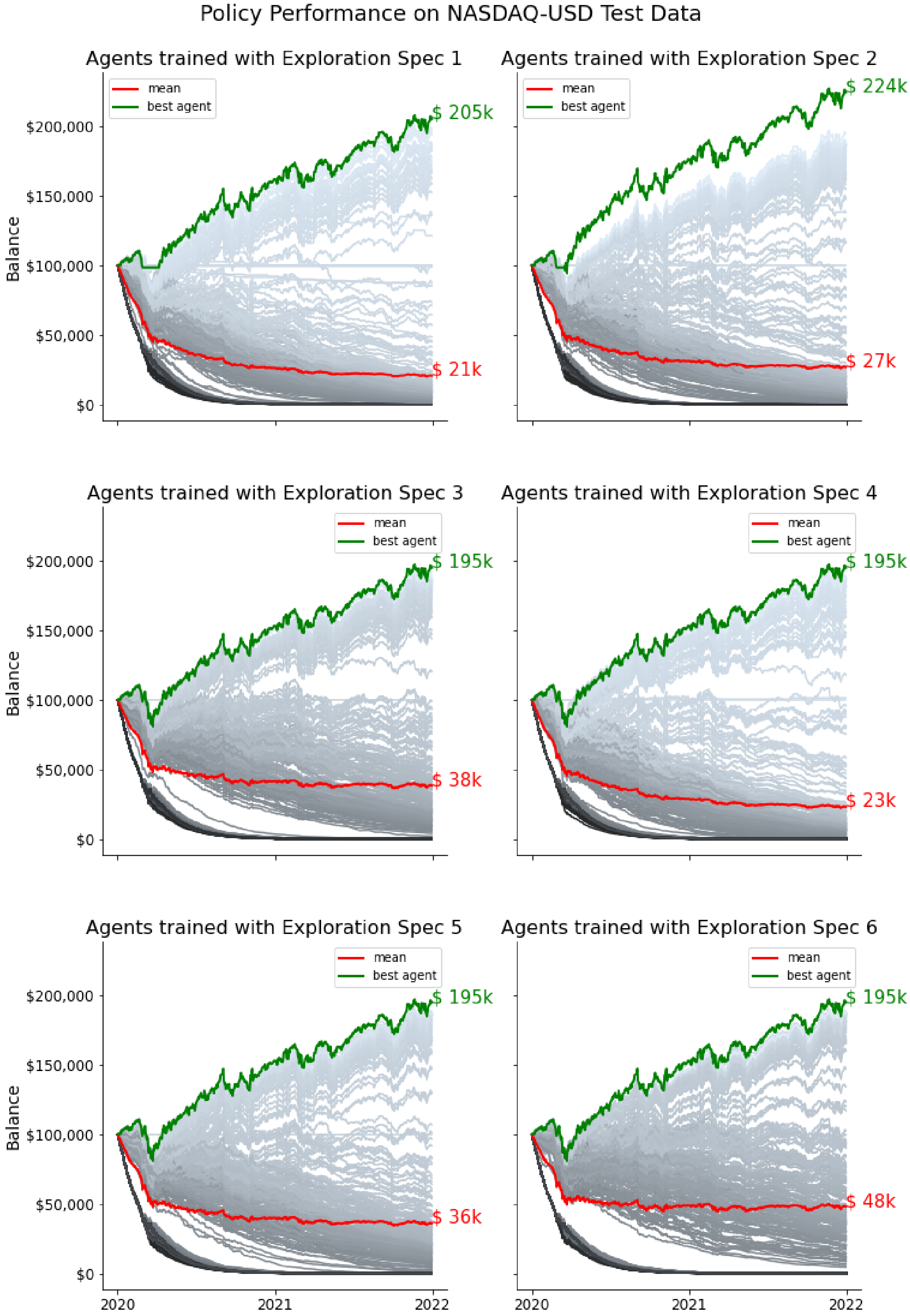

Figure 6.11: Policy Performance on NASDAQ Holdout Data

6.2.2.1.1.1 One-Sided T-test

Summary statistics for 30 training-test runs in unconstrained RL BTC-USD environments for each exploration rate specification are shown in table 6.12.

Exploration Rate Spec. ID	N	Mean	SD	SE	95% Conf.	Interval
1	30	46,446	5,413	988	44,425	48,467
2	30	55,393	4,204	768	53,823	56,963
3	30	89,717	4,025	735	88,215	91,220
4	30	51,680	3,824	698	50,252	53,107
5	30	81,081	3,829	699	79,651	82,511
6	30	110,621	4,817	879	108,823	112,420

Table 6.4: Mean Closing Balance Summary Statistics - Unconstrained RL

A one-sided T-test is performed between the mean Agent testing performance of exploration rate specification 6 from both the constrained and unconstrained approaches. Table 6.13 shows the results of the T-test.

Hypothesis

- H_0**:** Mean closing balance for exploration rate specification 6 in the constrained RL approach is not greater than the mean closing balance for exploration rate specification 6 in the unconstrained RL approach.
- H_A**:** Mean closing balance for exploration rate specification 6 in the constrained RL approach is greater than the mean closing balance for exploration rate specification 6 in the unconstrained RL approach.

t-statistic	p-value
3.9	1.3e-4

Table 6.5: One-Sided T-test Constrained vs Unconstrained RL

6.2.2.2 Analysis

6.2.2.2.1 Constrained vs Unconstrained - High-Level Comparison

6.2.2.2.1.1 Bitcoin

In references to Figures 6.3 and 6.9 (BTC-USD constrained and unconstrained RL results), the removal of constraints resulted in a drop in mean Agent closing balance across the board. An exception to this is exploration rate specification 3, which finished with a mean Agent closing balance of $87,000 (+$1,000) vs the constrained approach. In both the constrained and unconstrained scenarios, exploration rate specification 6 resulted in the highest mean Agent closing balance. The balance from the constrained approach is $5,000 (4.5%) higher than the balance of the unconstrained approach. A statistical significance test is needed to determine if this difference in significant.

Analysing the best Agents, the results appear to be inconsistent. Best-performing Agents from exploration rate specifications 2 and 3 in the constrained approach performed better than their counterparts in the unconstrained approach. However, exploration rate specifications 1,3 and 4 in the unconstrained approach outperformed their counterparts in the constrained approach, in the context of best-performing Agents. Reference is made to exploration rate specification 4 in Figure 6.9. The best-performing Agent here achieved a final closing balance of $907,000, or 38% higher than its constrained counterpart's best closing balance of $659,000. Although this difference is staggering, it could be due to chance. In any case, a t-test mean sample means from both approaches will show up significant differences, if any.

6.2.2.2.1.2 EOS

Results from the EOS-USD unconstrained RL testing phase (Figure 6.10) exhibit balanced outcomes versus the corresponding results from the constrained scenario (Figure 6.4). From a mean Agent performance standpoint, each approach achieved three higher balances that their counterparts. From a best-performing Agent standpoint, the constrained approach outperformed the unconstrained approach, with 4 out of the 6 exploration rate specifications leading to a better top-Agent performance. Of note, the best-performing Agent in exploration rate specification 5 in the constrained approach achieved a final closing balance of $423,000. This is $238,000 (129%) higher than its unconstrained counterpart. Although the outcomes are relatively balanced, it would appear that in general, the performance balance is tipped in favour of the constrained approach, as is the case for the BTC-USD market.

6.2.2.2.1.3 NASDAQ

The NASDAQ-USD unconstrained RL testing phase (Figure 6.11) resulted in 3 higher best-performing Agent closing balances than the constrained approach (Figure 6.5). One closing balance was equal and the constrained approach resulted in two higher best-performing Agent closing balances. Critically, on the mean Agent performance side, the removal of constraints led to degradation in final closing balance in all but one of the exploration rate specifications. This might indicate superiority of the constrained approach over the unconstrained approach - and that the more favourable outcomes of best-performing Agents in the unconstrained approach occurred by chance. Regardless, the NASDAQ-USD market under an unconstrained environment follows the other two markets in terms of its perceived inferiority versus the constrained setup.

6.2.2.2.2 Statistical Analysis

Due to the apparent reduction in mean performance on all three markets through the removal of constraints, a statistical test was carried out to see if the difference could be significant. It is assumed that a significant difference between the constrained and un constrained approaches applied to the BTC-USD market will hold up on the other markets, given the results on mean-Agent performance on all three markets. As such, a significance test is only carried out on the BTC-USD market.

The exact same procedure for generating sample means as the one carried out during the constrained RL testing was carried out. 30 sample means, representing mean Average closing balance for the BTC-USD market, were calculated. Table 6.4 shows the summary statistics from this step.

Exploration rate specification ID 6 is selected from both the constrained and unconstrained approaches for T-test purposes. This decision was made as it is a logical notion to hypothesise on differences between the most efficient exploration rate specifications. The null hypothesis states that the mean Agent closing balance from specification 6 in the constrained approach is not greater than the mean Agent closing balance from specification 6 in the unconstrained approach. The alternative hypothesis states that the mean Agent closing balance from specification 6 in the constrained approach is greater than the mean Agent closing balance from specification 6 in the unconstrained approach. The hypothesis was formulated as such as the mean value from the constrained approach was greater that that of the unconstrained approach.

A one-sided T-test was carried out using sample means from both specifications. Results returned a p-value of 1.3e-4, which is much less than the accepted value of 0.05. The null hypothesis can therefore be rejected and it can be concluded that the removal of constraints caused a statically significant reduction in mean Agent closing balance for exploration rate specification 6.

6.2.2.3 Interpretation

6.2.2.3.1 RQ2 - Constrained Reinforcement Learning (CRL) Effect on Performance

The removal of risk-mitigating constraints from the state-action space resulted in a statistically-significant reduction in mean Agent performance for one of the exploration rate specifications. This does not prove that a reduction in performance, on average, is prevalent in all markets. However, in reference to Figures 6.1 and 6.2, it reasonable to suggest that a systematic reduction in performance is a reality when guardrails are removed. At least, it is difficult to make a case for the contrary, if credence if given to the unconstrained training plots. These plots clearly show that Agents lose money rapidly when there are no safety features in place to protect their portfolios.

6.2.3 Performance Against Benchmark

6.2.3.1 Results

This section contains benchmark performance results. Specifically, trading results from the overall best performing Agent from both the constrained and unconstrained approaches are compared against a buy and hold strategy. The overall best performing Agent is the Agent, regardless of which specification it belongs to, that generalised the best on unseen data for all three markets. An overall best-performing Agent was selected from both the constrained and unconstrained testing results. It follows that two plots are provided for each market.

Figures 6.12 and 6.16 show the comparison between the best Agent performance versus a buy and hold strategy for the BTC-USD market, under a constrained and unconstrained environment, respectively. Figures 6.17 and 6.18 provide the same type of comparison for the EOS-USD market, while Figures 6.19 and 6.20 show the comparison results for the NASDAQ-USD market.

6.2.3.1.1 Bitcoin

Constrained

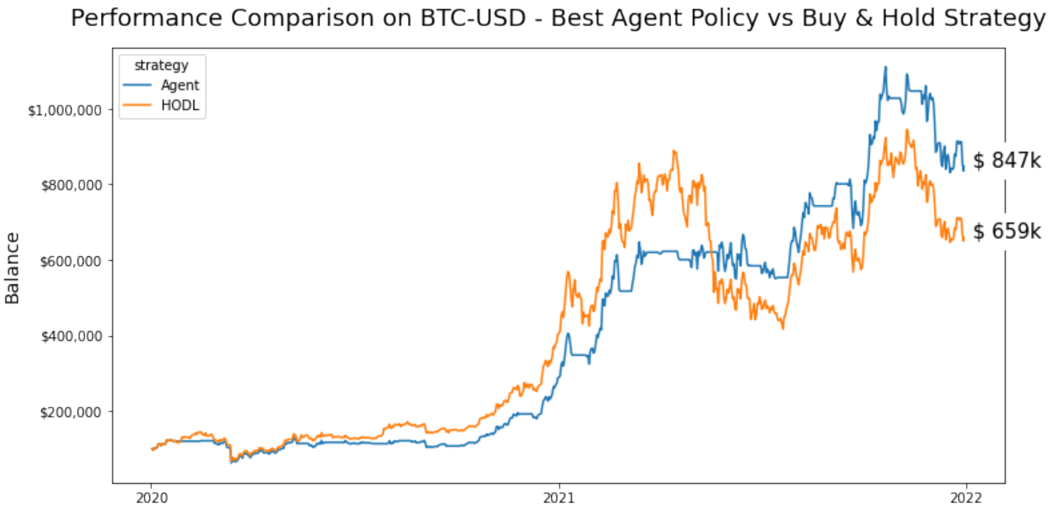

Figure 6.12: Benchmark Test - Agent vs Buy & Hold on BTC-USD Market

Unconstrained

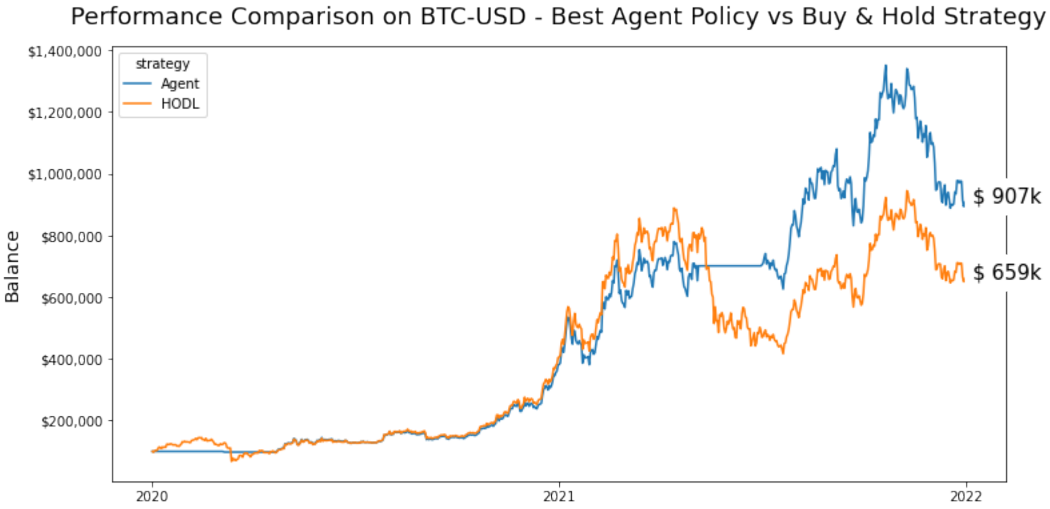

Figure 6.13: Benchmark Test - Agent vs Buy & Hold on BTC-USD Market

6.2.3.1.2 EOS

Constrained

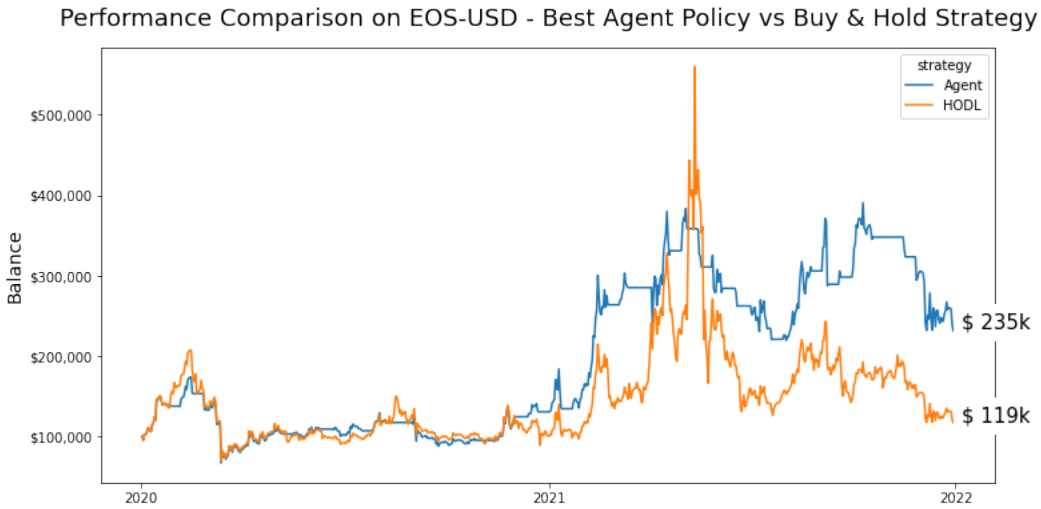

Figure 6.14: Benchmark Test - Agent vs Buy & Hold on EOS-USD Market

Unconstrained

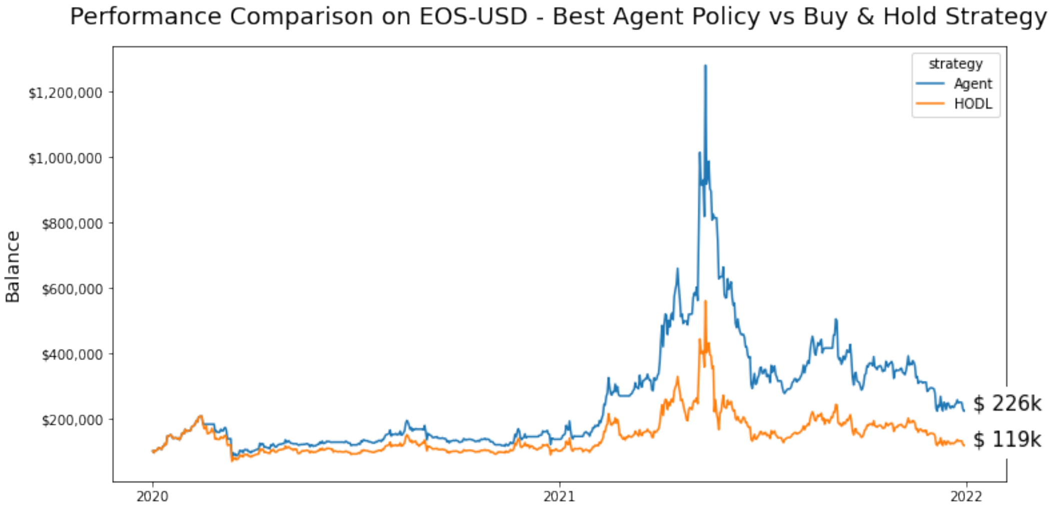

Figure 6.15: Benchmark Test - Agent vs Buy & Hold on EOS-USD Market

6.2.3.1.3 NASDAQ

Constrained

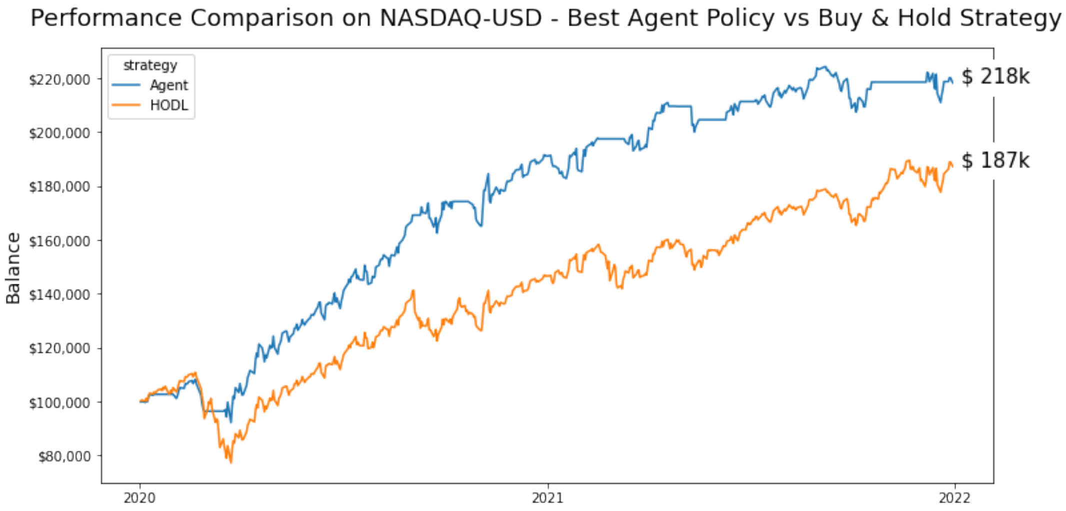

Figure 6.16: Benchmark Test - Agent vs Buy & Hold on NASDAQ-USD Market

Unconstrained

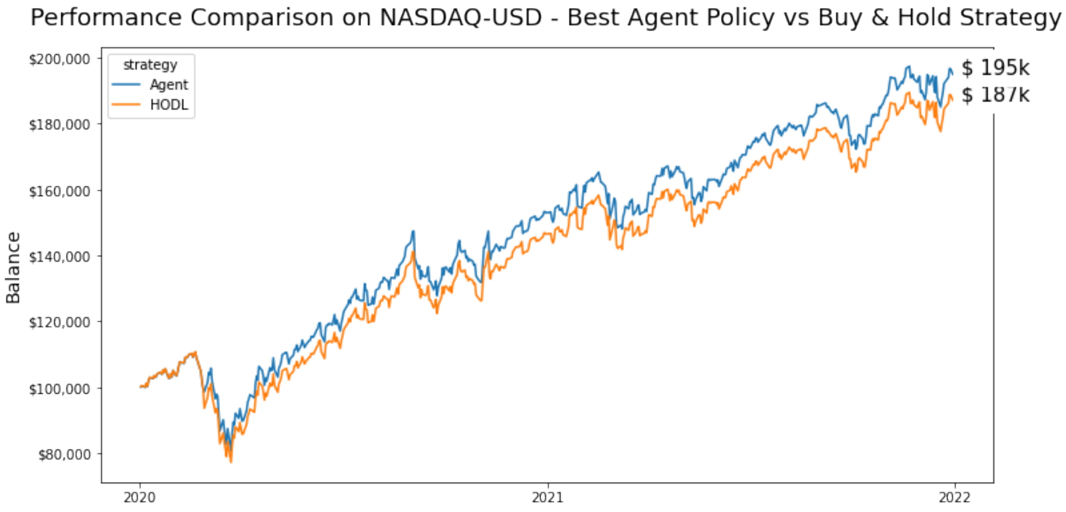

Figure 6.17: Benchmark Test - Agent vs Buy & Hold on NASDAQ-USD Market

6.2.3.2 Analysis

Each of the policies selected for benchmark testing outperformed the buy and hold strategy. On the BTC-USD market, the constrained and unconstrained policies achieved closing balances of \$847,000 and \$907,000, respectively. Compared with the buy and hold closing balance of \$659,000, the policies outperformed buy and hold by a magnitude of \$188,000 (29%) and \$248,000 (38%), respectively.

Testing against the EOS-USD market yieled the same outcome; both the constrained and unconstrained policies outperformed the buy and hold strategy. Results show performance gains of \$116,000 (97%) and \$107,000 (90%) respectively, versus the buy and hold strategy.

On the NASDAQ-USD benchmark test, both policies once again outperformed the buy and hold strategy. While the buy and hold strategy achieved a closing balance of \$187,000, the constrained and unconstrained approaches achieved closing balances of \$218,000 (+17%) and \$195,000 (+4%) respectively.

6.2.3.3 Interpretation

6.2.3.3.1 RQ3 - Reinforcement Learning vs Buy and Hold

Further tests are required to reason on significance of benchmark testing results. However, since the learned policies outperformed the buy and hold strategies in every benchmark test, it is reasonable to assume that yes, RL can indeed generate trading strategies to outperform the classic buy and hold strategy.

Reference is made to horizontal lines in each of the Agent's trends during the benchmark testing. The lines highlight periods where the Agents did not hold a position. In other words the policies prompted the Agents to get out of a position due to potentially adverse conditions. Figure 6.13 is a perfect example of this. Circa April 2021, when the buy and hold strategy's portfolio value plummeted do a huge decrease in Bitcoin's value, the Agent's learned policy 'knew' it was time to exit the position. Note how once Bitcoin's price started to rise again, the Agent re-entered into a position.

The positive takeaways here are two-fold; the Agents outperformed the market, and did so in a way that seems so intuitive to humans. This is quite impressive as financial trading is a notoriously difficult problem to solve.

Chapter 7

Conclusion

In this study, three research questions were proposed. Namely, does Q-Learning exploration rate variation affect learned policy performance on unseen data? How does constrained Reinforcement Learning affect profitability, and can Reinforcement Learning-generated trading strategies outperform the buy and hold strategy?

Addressing the first question, yes, variation in the Q-Learning exploration rate parameter does affect learned policy performance. Results show that as the exploration rate value is reduced during training, the resultant policies tend to perform better on unseen data. In fact, this claim is reinforced by a statistical test which proved that a statistical difference exists between the top two best performing exploration rate specifications. BTC-USD was the only market the test was carried out on, and therefore an open question remains as to whether this assertion holds up for other markets. Despite this open question, each market exhibited the same behaviour in terms of the mean Agent performance. Specifically, the mean Agent performance got better in each market as the exploration rate was reduced. Therefore is not unreasonable to assume that in general, variation in the exploration rate affects profitability of subsequently-created Agents.

Constrained RL has shown great promise in various industries. This study explored the effect of adding constraints to state-action spaces of Q-Learning, algorithmic-trading Agents. Initial results showed, with little doubt, that the removal of constraints caused a degradation in mean-Agent closing balance. The performance balance was not entirely in favour of the constrained approach and thus a significance test was carried out with a p-value threshold of 0.05. Results from the test led to the rejection of the null hypothesis that no difference between the two sets of results exist. In other words, constrained affects profitability in the positive sense.

Benchmark testing showed that RL-generated trading strategies can outperform the buy and hold strategy. In each market in this study, the most profitable Agents made a series of trades that resulted in a higher closing balance than the buy and hold approach. Furthermore, not just the constrained RL-generated strategies outperformed the buy and hold strategy, but also those generated from the unconstrained approach. In fact, the policy pertaining to the unconstrained approach performed better than than the constrained policy in the BTC-USD benchmark test. This is not to dispute outcomes from RQ2. It simply means that there exists a single policy generated from unconstrained RL that was capable of outperforming the best constrained policy. The significance results of RQ2 still stand to reason; that the removal of constraints leads to performance degradation, on average.

7.1 Future Work

7.1.1 Application to High Frequency Trading

This study dealt with financial market data at hourly and data intervals. While the outcomes are promising, they do not say anything about generalisation or applicability to other trading timeframes, such as those related to High-Frequency Trading; ¡ 1 second intervals. Indeed, HFT is a notoriously difficult environment to be profitable in, not least because of the transaction fees which are deducted from the portfolio balance at high speed. One area of future work could be to apply the methodology outlined in this study and change the experimental settings to overcome this challenge.

7.1.2 Application to Other Financial Instruments

Only cryptocurrencies and a composite index were considered in this study. These are but a rather small subset of the types of financial instruments that can be traded. Another area of further research on this topic would be to experiment with other instruments such as bonds, stocks, and FOREX markets. It would be interesting to see, initially, if the generalisation capabilities of the Agents created in this study, hold up to new types of instruments.

7.1.3 Further Technical Indicator Experimentation

There are hundreds of indicators used for technical analysis. In this study, only some of the most were used to encode the environment state. Further work could expand on this list and experiment with different combinations of indicators to see if performance can be improved. In particular, only one volume-based indicator, OBV, was used in this study. Could the encoding of volume in more elaborate ways, combined with price encoding, lead to better results?

References

[1] G. Lucarelli and M. Borrotti, "A deep q-learning portfolio management framework for the cryptocurrency market," *Neural Computing and Applications*, vol. 32, no. 23, pp. 17 229–17 244, 2020.

[2] S.-J. Bu and S.-B. Cho, "Learning optimal q-function using deep boltzmann machine for reliable trading of cryptocurrency," in *International Conference on Intelligent Data Engineering and Automated Learning*, Springer, 2018, pp. 468–480.

[3] S. Colianni, S. Rosales, and M. Signorotti, "Algorithmic trading of cryptocurrency based on twitter sentiment analysis," *CS229 Project*, pp. 1–5, 2015.

[4] F. Fang, W. Chung, C. Ventre, *et al.*, "Ascertaining price formation in cryptocurrency markets with deeplearning," *arXiv preprint arXiv:2003.00803*, 2020.

[5] L. Alessandretti, A. ElBahrawy, L. M. Aiello, and A. Baronchelli, "Anticipating cryptocurrency prices using machine learning," *Complexity*, vol. 2018, 2018.

[6] J. D. Piotroski, "Value investing: The use of historical financial statement information to separate winners from losers," *Journal of Accounting Research*, pp. 1–41, 2000.

[7] M. P. Taylor and H. Allen, "The use of technical analysis in the foreign exchange market," *Journal of international Money and Finance*, vol. 11, no. 3, pp. 304–314, 1992.

[8] W. Brock, J. Lakonishok, and B. LeBaron, "Simple technical trading rules and the stochastic properties of stock returns," *The Journal of finance*, vol. 47, no. 5, pp. 1731–1764, 1992.

[9] Z. Xiong, X.-Y. Liu, S. Zhong, H. Yang, and A. Walid, "Practical deep reinforcement learning approach for stock trading," *arXiv preprint arXiv:1811.07522*, 2018.

[10] Y. Niv, "Reinforcement learning in the brain," *Journal of Mathematical Psychology*, vol. 53, no. 3, pp. 139–154, 2009.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[12] S. Franklin and A. Graesser, "Is it an agent, or just a program?: A taxonomy for autonomous agents," in *International workshop on agent theories, architectures, and languages*, Springer, 1996, pp. 21–35.

[13] P. Y. Glorennec, "Reinforcement learning: An overview," in *Proceedings European Symposium on Intelligent Techniques (ESIT-00), Aachen, Germany*, Citeseer, 2000, pp. 14–15.

[14] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[15] M. Wunder, M. L. Littman, and M. Babes, "Classes of multiagent q-learning dynamics with epsilon-greedy exploration," in *ICML*, 2010.

[16] S. Ishii, W. Yoshida, and J. Yoshimoto, "Control of exploitation–exploration meta-parameter in reinforcement learning," *Neural networks*, vol. 15, no. 4-6, pp. 665–687, 2002.

[17] L. Busoniu, R. Babuska, and B. De Schutter, “A comprehensive survey of multiagent reinforcement learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[18] F. Fernández and M. Veloso, “Probabilistic policy reuse in a reinforcement learning agent,” in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 2006, pp. 720–727.

[19] M. Tokic, “Adaptive ε-greedy exploration in reinforcement learning based on value differences,” in *Annual Conference on Artificial Intelligence*, Springer, 2010, pp. 203–210.

[20] M. Tan, “Multi-agent reinforcement learning: Independent vs. cooperative agents,” in *Proceedings of the tenth international conference on machine learning*, 1993, pp. 330–337.

[21] K. Lei, B. Zhang, Y. Li, M. Yang, and Y. Shen, “Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading,” *Expert Systems with Applications*, vol. 140, p. 112 872, 2020.

[22] Z. Zhang, S. Zohren, and S. Roberts, “Deep reinforcement learning for trading,” *The Journal of Financial Data Science*, vol. 2, no. 2, pp. 25–40, 2020.

[23] J. M. Karpoff, “The relation between price changes and trading volume: A survey,” *Journal of Financial and quantitative Analysis*, vol. 22, no. 1, pp. 109–126, 1987.

[24] C. J. Neely, D. E. Rapach, J. Tu, and G. Zhou, “Forecasting the equity risk premium: The role of technical indicators,” *Management science*, vol. 60, no. 7, pp. 1772–1791, 2014.

[25] M. A. Dempster, T. W. Payne, Y. Romahi, and G. W. Thompson, "Computational learning techniques for intraday fx trading using popular technical indicators," *IEEE Transactions on neural networks*, vol. 12, no. 4, pp. 744–754, 2001.

[26] S. Gumparthi, "Relative strength index for developing effective trading strategies in constructing optimal portfolio," *International Journal of Applied Engineering Research*, vol. 12, no. 19, pp. 8926–8936, 2017.

[27] R. Rosillo, D. De la Fuente, and J. A. L. Brugos, "Technical analysis and the spanish stock exchange: Testing the rsi, macd, momentum and stochastic rules using spanish market companies," *Applied Economics*, vol. 45, no. 12, pp. 1541–1550, 2013.

[28] J. B. Chakole, M. S. Kolhe, G. D. Mahapurush, A. Yadav, and M. P. Kurhekar, "A q-learning agent for automated trading in equity stock markets," *Expert Systems with Applications*, vol. 163, p. 113 761, 2021.

[29] J. Achiam, A. Ray, and D. Amodei. "Safety gym." (2019), [Online]. Available: `https://openai.com/blog/safety-gym/` (visited on 09/07/2022).

[30] G.-H. Chen, M.-Y. Kao, Y.-D. Lyuu, and H.-K. Wong, "Optimal buy-and-hold strategies for financial markets with bounded daily returns," in *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, 1999, pp. 119–128.

[31] S. Yam, S. Yung, and W. Zhou, "Two rationales behind the 'buy-and-hold or sell-at-once'strategy," *Journal of Applied Probability*, vol. 46, no. 3, pp. 651–668, 2009.

[32] W. Tilson and J. Heins. "Buy and hold is risky." (2010), [Online]. Available: `https://www.kiplinger.com/article/investing/t052-c017-s001-buy-and-hold-is-risky.html` (visited on 09/07/2022).

[33] S. Ross. "Proof that buy-and-hold investing works." (2022), [Online]. Available: `https://www.investopedia.com/articles/investing/100215/statistical-proof-buyandhold-investing-pays.asp` (visited on 09/07/2022).

[34] D. Lohpetch and D. Corne, "Discovering effective technical trading rules with genetic programming: Towards robustly outperforming buy-and-hold," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, IEEE, 2009, pp. 439–444.

[35] ——, "Outperforming buy-and-hold with evolved technical trading rules: Daily, weekly and monthly trading," in *European Conference on the Applications of Evolutionary Computation*, Springer, 2010, pp. 171–181.

[36] Z. Dai, H. Zhu, and J. Kang, "New technical indicators and stock returns predictability," *International Review of Economics & Finance*, vol. 71, pp. 127–142, 2021.

[37] R. Peachavanish, "Stock selection and trading based on cluster analysis of trend and momentum indicators," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, IMECS, vol. 1, 2016, pp. 317–321.

[38] I. Gurrib, "Performance of the average directional index as a market timing tool for the most actively traded usd based currency pairs," *Banks and Bank Systems*, vol. 13, no. 3, pp. 58–70, 2018.

[39] J. W. Wilder, *New concepts in technical trading systems.* Trend Research, 1978.

[40] J. Coakley, M. Marzano, and J. Nankervis, "How profitable are fx technical trading rules?" *International Review of Financial Analysis*, vol. 45, pp. 273–282, 2016.

[41] J. Fernando. “Relative strength index (rsi) indicator explained with formula.” (2022), [Online]. Available: https://www.investopedia.com/terms/r/rsi.asp (visited on 09/07/2022).

[42] G. C. Lane, “Lane’s stochastics,” *Technical Analysis of Stocks and Commodities*, vol. 2, no. 3, p. 80, 1984.

[43] T. S. Chande and S. Kroll, *The new technical trader: boost your profit by plugging into the latest indicators*. John Wiley & Sons Incorporated, 1994, vol. 44.

[44] M. Asad Khan, “Technical analysis: Concept or reality?,” vol. 18, pp. 732–751, Oct. 2016.

[45] C. Team. “Stochastic rsi - overview, how to calculate, how to interpret.” (2021), [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/other/stochastic-rsi-stochrsi/ (visited on 09/07/2022).

[46] J. S. Vaiz and M. Ramaswami, “A study on technical indicators in stock price movement prediction using decision tree algorithms,” *American Journal of Engineering Research (AJER)*, vol. 5, no. 12, pp. 207–212, 2016.

[47] N. Fikri, K. Moussaid, M. Rida, A. El Omri, and N. Abghour, “A channeled multilayer perceptron as multi-modal approach for two time-frames algo-trading strategy,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 2, 2022.

[48] C. Team. “Moving average - overview, types and examples, ema vs sma.” (2022), [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/other/moving-average/ (visited on 09/07/2022).

[49] A. Hayes. "Golden cross patterns explained with examples and charts." (2022), [Online]. Available: https://www.investopedia.com/terms/g/goldencross.asp (visited on 09/07/2022).

[50] J. Maverick. "Moving average convergence divergence and its calculation." (2022), [Online]. Available: https://www.investopedia.com/ask/answers/122414/what-moving-average-convergence-divergence-macd-formula-and-how-it-calculated.asp (visited on 09/07/2022).

[51] T. T.-L. Chong and W.-K. Ng, "Technical analysis and the london stock exchange: Testing the macd and rsi rules using the ft30," *Applied Economics Letters*, vol. 15, no. 14, pp. 1111–1114, 2008.

[52] D. Eric, G. Andjelic, and S. Redzepagic, "Application of macd and rvi indicators as functions of investment strategy optimization on the financial market," *Zbornik radova Ekonomskog fakulteta u Rijeci: časopis za ekonomsku teoriju i praksu*, vol. 27, no. 1, pp. 171–196, 2009.

[53] A. Hayes. "On balance volume (obv) definition." (2022), [Online]. Available: https://www.investopedia.com/terms/o/onbalancevolume.asp (visited on 09/04/2022).

[54] W. W. H. Tsang, T. T. L. Chong, *et al.*, "Profitability of the on-balance volume indicator," *Economics Bulletin*, vol. 29, no. 3, pp. 2424–2431, 2009.

[55] J. Chen and E. P. Tsang, *Detecting Regime Change in Computational Finance: Data Science, Machine Learning and Algorithmic Trading.* CRC Press, 2020.

[56] X. Liu, H. An, and L. Wang, "Performance of generated moving average strategies in natural gas futures prices at different time scales," *Journal of Natural Gas Science and Engineering*, vol. 24, pp. 337–345, 2015.

[57] E. Tsang and J. Chen, "Regime change detection using directional change indicators in the foreign exchange market to chart brexit," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 3, pp. 185–193, 2018.

[58] J. B. Glattfelder, A. Dupuis, and R. B. Olsen, "Patterns in high-frequency fx data: Discovery of 12 empirical scaling laws," *Quantitative Finance*, vol. 11, no. 4, pp. 599–614, 2011.

[59] N. Alkhamees and M. Fasli, "A directional change based trading strategy with dynamic thresholds," in *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE, 2017, pp. 283–292.

[60] J. W. Lee, J. Park, O. Jangmin, J. Lee, and E. Hong, "A multiagent approach to *q*-learning for daily stock trading," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 6, pp. 864–877, 2007.

[61] H. Park, M. K. Sim, and D. G. Choi, "An intelligent financial portfolio trading strategy using deep q-learning," *Expert Systems with Applications*, vol. 158, p. 113 573, 2020.

[62] X. Du, J. Zhai, and K. Lv, "Algorithm trading using q-learning and recurrent reinforcement learning," *positions*, vol. 1, no. 1, 2016.

[63] cryptodatadownload.com. "Bistamp data." (2022), [Online]. Available: `https://www.cryptodatadownload.com/data/bitstamp/` (visited on 08/01/2022).

[64] ——, "Bitfinex data." (2022), [Online]. Available: `https://www.cryptodatadownload.com/data/bitfinex/` (visited on 08/01/2022).

[65] backtestmarket.com. "Nasdaq 1h historial data." (2022), [Online]. Available: `https://www.backtestmarket.com/en/nasdaq-1h` (visited on 08/01/2022).

Appendix

Source Code Link: `https://github.com/johnnyirldev/RL_Trading`