Universiteit Leiden
The Netherlands

Bachelor Data Science & Artificial Intelligence

Exploring Reinforcement Learning in the Financial World

Ferid Sophian Kendić

Supervisors:
Aske Plaat
Koen Ponse

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

01/07/2025

Abstract

This thesis explores the use of deep reinforcement learning (DRL) for automated stock trading. Two DRL algorithms, Proximal Policy Optimization (PPO) and Advantage Actor-Critic (A2C), were implemented and evaluated using historical daily data from the Dow Jones 30 stocks ranging from 2018 to 2025. The agents were trained on data from 2018 to 2023 using a custom trading environment and reward structure to optimize trading behavior under realistic market conditions. Their performance was then evaluated on a separate test set covering 2023 to 2025. Results show that both PPO and A2C consistently outperformed the traditional Buy & Hold strategy in terms of average return on investment (ROI), with PPO achieving the best performance across varying market conditions. These findings suggest that DRL is a promising approach for building adaptive trading systems that respond effectively to market dynamics.

Contents

1 Introduction **1**

2 Related Work and Background **3**
2.1 Traditional Financial Theories and the Need for Automation . . . 3
2.2 Reinforcement Learning in Finance . . . 4
2.3 Key Deep Reinforcement Learning Algorithms . . . 4
2.3.1 Proximal Policy Optimization: . . . 4
2.3.2 Advantage Actor-Critic . . . 5
2.4 Prior Applications of DRL in Stock Trading . . . 6
2.5 Research Gap and Contribution . . . 6

3 Experimental Setup **7**
3.1 Data Collection and Preprocessing . . . 7
3.2 Custom Trading Environment . . . 7
3.3 Agents and Training Procedure . . . 8
3.4 Evaluation Strategy . . . 9
3.5 Metrics and Comparison . . . 9

4 Results **10**
4.1 Overview of Experimental Setup . . . 10
4.2 Performance Summary . . . 11
4.3 Detailed Analysis . . . 12
4.4 Notable Examples . . . 13
4.5 Statistical Observations . . . 13
4.6 Possible Explanations . . . 14
4.7 Visual Comparison . . . 14
4.8 Interpretation . . . 15
4.9 Learning Dynamics Over Time . . . 16

5 Conclusion **18**
5.1 Summary of Findings . . . 18
5.2 Answering the Research Questions . . . 19
5.3 Limitations and Future Work . . . 19
5.4 Conclusion . . . 20
Appendix . . . 21

References 24

Chapter 1

Introduction

Due to the rapid and ongoing development within artificial intelligence and reinforcement learning (RL), automated stock trading has gained a lot of attention both within academic research areas (such as machine learning and quantitative finance) as within the financial industry [YLZW20]. Where traditional trading strategies mainly rely on predefined rules of heuristics, automated stock trading relies on machine learning, and more specifically RL.

Reinforcement learning has given us the opportunity to develop intelligent and adaptive trading agents. These agents learn optimal actions from direct interaction with their environment (observing market prices, making trades, and receiving feedback in the form of profit or loss [SB18]). They make sequential decisions by exploring different strategies and receive feedback in the form of rewards Mnih et al. [SB18]. This makes them capable of navigating through complex and volatile market environments. RL is a branch of machine learning and focuses on how agents take actions in an environment to maximize cumulative rewards. This makes RL a suitable candidate for stock trading, as it handles rewards, balances between risk and reward, and does continuous decision making which are all important factors in trading scenarios.

According to studies such as [SB18] and [YLZW20], RL based agents can outperform traditional rule-based systems by learning policies directly from raw price data, adjusting to market volatility and discovering profitable patterns. Therefore, it shows great potential in sequential decision-making tasks such as stock trading.

Deep reinforcement learning (DRL) combines deep neural networks with RL, improving the agent's ability to maximize (cumulative) rewards. This allows trading agents to learn effective policies directly from raw market data without relying on manually engineered features [MKS+15].

This thesis will explore the suitability for RL for financial market applications, with the focus being on stock trading. The main research question is: "Is reinforcement learning a suitable approach for financial market applications, particularly stock trading?" To investigate this, we will be researching two subquestions: "Can reinforcement learning outperform traditional strategies such as Buy & Hold?" and "Can a reinforcement learning trading agent be effectively deployed using real-world market data?"

To research these questions, we will explore the application of two different DRL algorithms such as Proximal Policy Optimization (PPO) and Advantage Actor-Critic (A2C) to create and develop 2 different automated stock trading agents. These algorithms were selected due to their different approaches and their balance between efficiency, stability, and performance [SWD+17, LHP+15].

This research will implement, evaluate, and compare these DRL algorithms within a custom-built trading environment to simulate realistic market conditions. The performance of the agents is measured against benchmarks such as the Buy & Hold strategy using metrics like profit, return on investment (ROI), and cumulative reward. We analyze our performance over a period ranging from 2018 to 2025, to compare the suitability and adaptability of the agents.

The thesis will include the following chapters: Chapter 2 defines key concepts and technical background related to RL and financial trading; Chapter 3 reviews related literature on automated trading with machine learning; Chapter 4 describes the experimental setup, datasets, and results; and Chapter 5 concludes with a summary of findings and directions for future research. At the end of the thesis, an appendix is added, summarizing the hyperparameters and training configurations used for the DRL agents.

Chapter 2

Related Work and Background

2.1 Traditional Financial Theories and the Need for Automation

Classical/traditional financial theories, such as Markowitz's Modern Portfolio Theory (MPT), assume that investors make rational decisions to maximize their expected returns for a given level of risk [Mar52]. MPT represents assets based on their expected returns and the covariances between them (for example, how asset prices move relative to one another). This allows for the construction of portfolios that lie on the efficient frontier (the set of portfolios offering the highest expected return for each level of risk, or the lowest risk for each level of return) [Mar52, LdPSFF24].

However, one of the issues that these models face is that they rely on strong assumptions, such as ideal market conditions and fully rational actors (investors who are perfectly informed and free from behavioral biases). Real financial markets are often volatile, unpredictable, and influenced by irrational human behavior such as emotion [Lo05]. As a result, the assumptions underlying these theories frequently break down, motivating the need for adaptive, data-driven approaches such as automated trading systems.

There are several limitations that are exposed in these traditional approaches, as they remain limited in their ability to adapt to the complexity of real-world markets. In these dynamic and complex environments, such limitations have contributed to the growing demand for automated stock trading systems. These systems can analyze large datasets, operate continuously, and avoid common human mistakes such as emotional decision making. Other frequent behavioral biases include overconfidence, confirmation bias, and loss aversion, which can lead to poor trading decisions like holding on to losing positions or making impulsive trades [KT79, Shl00].

A common early form of automation is the classical rule-based system [Cha08, TGL13]. These systems rely on predefined rules (for example, selling after a fixed price drop). While it can be more consistent than manual trading, these systems often lack adaptability. Their inflexible rules limit their ability to respond effectively to unexpected market shifts or structural changes. While this thesis does not implement a classical rule-based system, it is included here to illustrate the motivation for more flexible approaches based on DRL.

2.2 Reinforcement Learning in Finance

Reinforcement Learning is a well-suited option/candidate for financial problems that require sequential decision-making under uncertainty, such as portfolio management and stock trading. [HXY21, BGW+24].

Stock trading environments can be modeled as Markov Decision Processes (MDP), which captures the sequence of decisions and feedback in dynamic environments [BGW+24, KSS+23]. In this setting, an agent observes a state s_t (e.g., stock prices, technical indicators, portfolio holdings), selects an action a_t (e.g., buy, sell, or hold), receives a reward r_t (e.g., profit or loss), and transitions to a new state s_{t+1}. The agent's objective is to learn a policy $\pi(a|s)$ that maximizes expected cumulative rewards over time. MDPs are particularly suitable for this context because they explicitly model state-action-reward dependencies, allowing agents to learn from sequential interactions. Other learning methods like (static) supervised learning cannot achieve this [KSS+23].

Applying RL to financial markets does introduce several challenges and issues[BGW+24, KSS+23, YLZW20]. Financial environments are non-stationary, meaning that they are continuously influenced by external variables. These variables could be macroeconomic indicators (i.e., interest rates or inflation) to geopolitical events (e.g., elections or conflicts), market sentiment (i.e., public or investor mood reflected in media or trading behavior), and institutional order flows (i.e., large-volume trades by major financial institutions) [BGW+24]. Because of this they are only partly observable [MA18].

This partial observability often leads to sparse rewards, and increases the risk of overfitting when models are only trained on historical data. However, despite these obstacles, recent advances in DRL have enabled the use of neural networks to approximate value functions and policies. This allows agents to process high-dimensional input data, adapt to complex market dynamics, and effectively learn in partially observable and non-stationary environments [MKS+15, ZLZW24, BGW+24].

2.3 Key Deep Reinforcement Learning Algorithms

2.3.1 Proximal Policy Optimization:

PPO is a policy gradient RL algorithm introduced by Schulman et al. [SWD+17]. It is designed to train agents to learn optimal behaviors in complex and high-dimensional environments, such as stock trading, where the agent must process many features and make sequential decisions.

PPO aims to balance learning performance (fast improvement) and training stability (avoiding policy divergence). PPO belongs to the class of actor-critic methods, where two neural networks are used: the actor and the critic. The actor proposes actions based on a policy, while the critic

estimates the value function, reflecting how good a given state is.

PPO improves training stability by preventing large and destabilizing updates to the policy, which can cause the agent to diverge from previously effective behavior. To achieve this, PPO uses a clipped surrogate objective that limits how much the new policy is allowed to deviate from the old one [SWD+17].:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right], \tag{2.1}$$

where $r_t(\theta)$ is the probability ratio between the new and old policies, and $\hat{A}_t$ is the estimated advantage function, which measures how much better an action is compared to the average. The clipping function ensures stable and conservative updates by preventing overly large policy shifts during training [SWD+17].

PPO often uses Generalized Advantage Estimation (GAE) [SML+15] to reduce the variance in advantage estimates, using the formula below:

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}. \tag{2.2}$$

PPO is simple to implement, sample-efficient (i.e., it can learn effectively from fewer interactions with the environment), and performs well across a wide range of tasks, such as robotic control, video games, and resource allocation [SWD+17, ABC+20]. Its stability and general performance have made it a popular choice for real-world applications, including stock trading and financial portfolio management [BGW+24].

2.3.2 Advantage Actor-Critic

A2C is a RL algorithm that combines the benefits of both policy-based and value-based methods [MBM+16]. It contains two models: an actor that updates the policy, and a critic that estimates the value function. A2C improves training efficiency by using the advantage function to reduce variance. The equation looks as follows:

$$A(s_t, a_t) = r_t + \gamma V(s_{t+1}) - V(s_t), \tag{2.3}$$

This advantage function quantifies how much better taking action a_t in state s_t is compared to the expected return of the current policy at that state [SB18]. The actor uses this signal to adjust the policy, while the critic minimizes the error in value prediction.

Unlike its predecessor A3C, A2C runs multiple parallel environments but updates the policy synchronously, which makes it more stable and easier to implement [MBM+16].

A2C is a good candidate for tasks with discrete action spaces and offers a good balance between performance and simplicity [MBM+16].

2.4 Prior Applications of DRL in Stock Trading

Yang et al. [YLZW20] introduced a DRL agent that combines multiple algorithms to improve trading performance. Their results showed that DRL agents can outperform traditional baselines, such as Buy & Hold and moving average crossover strategies, especially in trending or structured market conditions/periods where stock prices exhibit clear upward or downward momentum or follow recognizable patterns. In these environments, DRL agents are can learn and exploit dynamic patterns to adapt their policies. However, their work also highlighted key challenges, like high performance variance in and limited generalization across different market environments.

Other studies have focused on individual algorithms such as PPO and A2C for financial trading [BGW+24]. PPO is known for its training stability and conservative updates, which improve stability but may also limit rapid adaptation in highly volatile markets. A2C offers a balance between learning efficiency and performance, though its reliance on sequential signal structures can make it less effective when data is noisy or weakly structured.

2.5 Research Gap and Contribution

Despite promising results, many DRL-based trading systems still lack a comparative analysis across different algorithms and under different market conditions. Most existing studies either focus on a single strategy or evaluate performance under limited scenarios. Furthermore, challenges such as overfitting to historical data and poor generalization to new market conditions remain largely unresolved.

This thesis addresses these gaps by developing and evaluating two DRL algorithms PPO and A2C within a custom-built trading environment, `SimpleTradingEnv`. This environment simulates realistic trading using historical data from the Dow Jones 30 stocks (ranging from 2018 to 2025), and applies consistent reward structures and trading penalties. The performance of both agents are compared against a traditional Buy & Hold baseline.

To simulate market diversity, we evaluate the agents performance across different market regimes, such as bullish (upward trending), bearish (downward trending), and sideways (range-bound, stays mostly the same or slight change) periods. This evaluation helps evaluate not only which algorithm performs best overall, but also which works best under each specific market condition. Additionally, this thesis contributes a reusable evaluation framework including `SimpleTradingEnv`, PPO and A2C implementations, and standardized Dow 30 test data to support future research on DRL-based trading systems.

Chapter 3

Experimental Setup

3.1 Data Collection and Preprocessing

We selected the top 30 stocks of the Dow Jones Industrial Average (Dow 30) as our dataset, representing a diverse set of large-capital U.S. stocks across various sectors, ranging from technology and healthcare to finance and consumer goods. The historical daily price data was downloaded from Yahoo Finance for the period from January 1, 2018 to January 1, 2025.

For each stock, we retained four key price features for each day: the opening price, the highest and lowest prices of the day, and the closing price. These will be referred to as OHLC values. Additionally, we computed two technical indicators:

- **SMA5**: 5-day Simple Moving Average of closing prices.
- **SMA20**: 20-day Simple Moving Average of closing prices.

Since the calculation of moving averages requires several prior data points, the first 19 entries of each stock's time series were missing valid SMA values. These rows were removed to ensure clean and complete input data for the trading environment. Finally, the data was split chronologically into a training set (80%) and a test set (20%) to preserve the temporal structure.

3.2 Custom Trading Environment

We implemented a custom OpenAI Gym environment called `SimpleTradingEnv`, simulating a single-agent stock trading scenario. This environment provides a reproducible setting to train and test RL agents using historical stock data.

The agent begins each episode with an initial cash balance of \$10,000 and no stock holdings. At each timestep, the agent selects a continuous action $a \in [-1, 1]$, where:

- $a > 0$: Buy action (scaled by the maximum number of affordable shares),
- $a < 0$: Sell action (scaled by the current number of shares held),

- $a = 0$: Hold action.

The observation space consists of:

- OHLC values (Open, High, Low, Close) for the past 5 timesteps,
- SMA5 and SMA20 values over the same 5-day window,
- Current cash balance,
- Number of shares held.

This results in a 32-dimensional observation vector:

$$(4\ (\text{OHLC}) + 2\ (\text{SMA})) \times 5\ (\text{days}) + 2\ (\text{cash} + \text{shares}) = 32$$

The reward function at each timestep t is defined as:

$$\text{Reward}_t = \Delta\text{Portfolio Value} - \lambda \cdot \text{Shares Traded}$$

where $\lambda = 0.01$ is a penalty factor that discourages excessive trading. This reflects transaction costs and promotes more realistic trading behavior by preventing overfitting to small fluctuations in price.

While OpenAI Gym is no longer actively maintained and has been replaced by Gymnasium [Fou23], Gym (v0.26) was selected for this project because of its compatibility with `stable-baselines3` and widespread support in RL workflows.

Code and reproducibility. The full implementation of `SimpleTradingEnv`, including training scripts and evaluation tools, is available on GitHub: https://github.com/MacakFTW/Exploring-Reinforcement-Learning-in-the-Financial-World.

3.3 Agents and Training Procedure

We implemented two DRL agents:

- Proximal Policy Optimization (PPO) [SWD+17],
- Advantage Actor-Critic (A2C) [MBM+16].

Both agents were trained using the `stable-baselines3` library Raffin et al. [RHG+21], an open-source Python library built on PyTorch that provides implementations of RL algorithms such as PPO and A2C. We used the default multilayer perceptron (MLP) policy. Each agent was trained for 50,000 timesteps per stock within a `DummyVecEnv`-wrapped instance of `SimpleTradingEnv`.

3.4 Evaluation Strategy

After training, each agent was evaluated on the test portion of the data (20% of each stock's time series) by running a single full episode. The final portfolio value at the end of the test period was recorded. As a baseline, we used a simple Buy & Hold strategy: purchasing 10 shares at the start of the test period (if affordable) and holding them without trading until the end.

For each strategy, we report the following metrics:

- Final Portfolio Value,
- Profit (absolute gain over initial capital),
- Return on Investment (ROI):

$$\text{ROI} = \frac{\text{Final Portfolio Value} - 10{,}000}{10{,}000} \times 100\%$$

We also recorded the raw price change over the test period, this is the difference between the closing price on the first and last day of the test set. This served as a basic indicator of the overall market trend (e.g., bullish or bearish) for that period.

3.5 Metrics and Comparison

All experiments were conducted per stock to evaluate how each agent performs on individual stocks. After evaluation, the final results were stored in a Python dictionary and then exported to a CSV file named trading_results.csv. The CSV file includes:

- Final portfolio values and derived metrics for PPO,
- Final portfolio values and derived metrics for A2C,
- Final portfolio value for Buy & Hold,
- Raw price change over the test period.

These metrics allow for a direct comparison between DRL agents and the Buy & Hold strategy.

Chapter 4

Results

4.1 Overview of Experimental Setup

Two DRL algorithms, PPO and A2C, were trained and tested on historical daily data from the Dow Jones Industrial Average (Dow 30) stocks. Each model was trained individually per stock. For each stock, data from 2018 to 2023 was used as the training set, while data from 2023 to 2025 was used as the test set. This split in our dataset ensured that the models were evaluated on unseen realistic data from the same source.

The custom environment used for training included a 5-day lookback window and incorporated OHLCV features along with technical indicators such as the 5-day and 20-day Simple Moving Averages (SMA5 and SMA20). Each model began with an initial capital of $10,000 and was trained for 50,000 timesteps per stock.

As a benchmark, a Buy & Hold strategy was implemented for each stock over the same period. The way this benchmark works is that at the beginning, if the portfolio has enough funds, 10 stocks will be bought and hold until the end period.

4.2 Performance Summary

Table 4.1 shows the results that the algorithms had compared to the Buy & Hold strategy. The results are shown as the absolute portfolio value, the profit, the ROI and the price change per stock. PPO and A2C both outperform Buy & Hold on average, with PPO achieving the highest returns overall.

Table 4.1: Comparison of PPO, A2C, and Buy & Hold strategies across Dow 30 stocks

Ticker	PPO_Abs	PPO_Profit	PPO_ROI	A2C_Abs	A2C_Profit	A2C_ROI	B&H_Abs	B&H_Profit	B&H_ROI	Price_Change
AAPL	11174.69	1174.69	11.75%	14892.82	4892.82	48.93%	10718.96	718.96	7.19%	71.90%
AMGN	9490.78	-509.22	-5.09%	10177.31	177.31	1.77%	10111.31	111.31	1.11%	11.13%
AXP	18880.43	8880.43	88.80%	18023.73	8023.73	80.24%	11325.94	1325.94	13.26%	132.59%
BA	10911.24	911.24	9.11%	7769.95	-2230.05	-22.30%	9402.90	-597.10	-5.97%	-59.71%
CAT	13674.86	3674.86	36.75%	13827.32	3827.32	38.27%	10834.42	834.42	8.34%	83.44%
CSCO	9067.73	-932.27	-9.32%	11102.79	1102.79	11.03%	10072.94	72.94	0.73%	7.29%
CVX	8770.60	-1229.40	-12.29%	9925.86	-74.14	-0.74%	9909.10	-90.90	-0.91%	-9.09%
DIS	9408.56	-591.44	-5.91%	10000.00	0.00	0.00%	10236.24	236.24	2.36%	23.62%
DOW	9959.43	-40.57	-0.41%	8192.53	-1807.47	-18.07%	9951.86	-48.14	-0.48%	-4.81%
GS	15505.42	5505.42	55.05%	12434.24	2434.24	24.34%	12458.08	2458.08	24.58%	245.81%
HD	11809.96	1809.96	18.10%	12440.33	2440.33	24.40%	10702.22	702.22	7.02%	70.22%
HON	11874.40	1874.40	18.74%	12429.14	2429.14	24.29%	10409.98	409.98	4.10%	41.00%
IBM	12372.70	2372.70	23.73%	15413.32	5413.32	54.13%	10832.94	832.94	8.33%	83.29%
INTC	9686.34	-313.66	-3.14%	5472.17	-4527.83	-45.28%	9849.57	-150.43	-1.50%	-15.04%
JNJ	9924.68	-75.32	-0.75%	8982.12	-1017.88	-10.18%	9793.83	-206.17	-2.06%	-20.62%
JPM	12648.84	2648.84	26.49%	16339.68	6339.68	63.40%	10886.50	886.50	8.87%	88.65%
KO	11321.77	1321.77	13.22%	10743.80	743.80	7.44%	10040.53	40.53	0.41%	4.05%
MCD	10424.61	424.61	4.25%	10704.81	704.81	7.05%	10109.66	109.66	1.10%	10.97%
MMM	12697.88	2697.88	26.98%	16039.19	6039.19	60.39%	10475.54	475.54	4.76%	47.55%
MRK	9234.69	-765.31	-7.65%	9352.40	-647.60	-6.48%	9950.32	-49.68	-0.50%	-4.97%
MSFT	11312.64	1312.64	13.13%	13271.27	3271.27	32.71%	11008.07	1008.07	10.08%	100.81%
NKE	8155.46	-1844.54	-18.45%	7452.29	-2547.71	-25.48%	9708.95	-291.05	-2.91%	-29.11%
PG	10233.57	233.57	2.34%	11339.04	1339.04	13.39%	10163.95	163.95	1.64%	16.39%
CRM	12742.74	2742.74	27.43%	13963.66	3963.66	39.64%	11230.59	1230.59	12.31%	123.06%
TRV	11385.82	1385.82	13.86%	15428.28	5428.28	54.28%	10788.44	788.44	7.88%	78.84%
UNH	11357.78	1357.78	13.58%	10422.22	422.22	4.22%	10063.88	63.88	0.64%	6.39%
V	12482.54	2482.54	24.83%	13264.44	3264.44	32.64%	10752.43	752.43	7.52%	75.24%
VZ	11444.69	1444.69	14.45%	13054.43	3054.43	30.54%	10088.82	88.82	0.89%	8.88%
WBA	8104.65	-1895.35	-18.95%	3852.55	-6147.45	-61.47%	9835.43	-164.57	-1.65%	-16.46%
WMT	15739.15	5739.15	57.39%	17504.02	7504.02	75.04%	10376.49	376.49	3.76%	37.65%

Table 4.2 summarizes the performance of PPO, A2C, and the Buy & Hold strategy across the Dow 30 stocks. For each stock, we report the final portfolio value (at the end of the test set, 01/01/2025), absolute profit, ROI, and the raw price change over the test period.

Table 4.2: Average Performance per Strategy

Strategy	**Avg. Final Value**	**Avg. Profit**	**Avg. ROI**	**Profitable Trades**
PPO	\$11,506.23	\$1,506.23	**15.06%**	20 / 30
A2C	\$11,411.01	\$1,411.01	**14.11%**	21 / 30
Buy & Hold	\$10,462.43	\$462.43	**4.62%**	22 / 30

Both PPO and A2C significantly outperformed Buy & Hold on average, as shown in Table 4.2. Although Buy & Hold achieved a higher number of profitable trades (22 out of 30), its average profit

per trade was considerably lower ($462.43) compared to PPO ($1,506.23) and A2C ($1,411.01). This indicates that while Buy & Hold was more frequently profitable, its gains were typically modest. In contrast, PPO and A2C achieved larger profits when they were successful, indicating that the DRL agents were better at adapting effectively to varying market conditions. PPO in particular showed the highest average profit, implying more consistent and robust performance across stocks.

4.3 Detailed Analysis

PPO vs. Buy & Hold

PPO outperformed the Buy & Hold strategy in 23 out of 30 stocks, achieving a maximum ROI of 88.80% on American Express (AXP). This suggests that PPO was able to effectively adapt to dynamic price movements and take advantage of market opportunities that Buy & Hold could not exploit.

While Buy & Hold mainly benefits from long term upward trends, PPO continuously adjusted its positions in response to recent market conditions, allowing it to navigate through both trending and volatile environments more effectively. Its continuous evaluation of state-action values helped the agent make better decisions such as holding onto losing positions for too long (which Buy & Hold does per default). This resulted into better cumulative rewards across the wide range of stocks.

A2C vs. Buy & Hold

A2C outperformed the Buy & Hold strategy in 22 out of 30 stocks, also achieving its strongest results on AXP with a ROI of 80.24%. However, A2C also suffered significant losses, particularly on bearish stocks like Walgreens Boots Alliance (WBA), where it had a negative ROI of -61.47%.

This could suggest that while A2C can effectively exploit consistent trends, it may be more vulnerable in volatile or irregular market conditions. Since A2C is an on-policy algorithm that updates its policy based on the current trajectory, it can be more sensitive to short-term noise. In stocks like WBA, which may have shown inconsistent signals during the test period, A2C may have overfitted to recent trends and failed to adapt when the market direction changed.

Furthermore, the critic in A2C estimates the value function, which can introduce additional variance if the price signals are unpredictable with sudden price drops. This makes A2C more prone to suboptimal decision-making in environments where the data is not sequentially coherent, leading to poorer generalization.

In summary, while A2C demonstrated strong average performance, its performance instability in unstructured markets may explain its hard losses in certain stocks.

Buy & Hold

Buy & hold generated profitable trades on 22 out of 30 stocks but generally underperformed compared to the RL agents in terms of average profit and ROI (4.62% vs. 15.06% for PPO and

14.11% for A2C). It outperformed PPO in 7 cases and A2C in 8, particularly in stocks that experienced bearish market conditions.

For example, in WBA, a consistently declining stock during the test period (2023–2025), Buy & Hold achieved a relatively small loss of -1.65%, whereas A2C suffered a much larger loss of -61.47%, and PPO also underperformed with -18.95%. Similar patterns can be seen in Intel (INTC), where Buy & Hold yielded a modest loss of -1.50%, while PPO had a loss of -3.14% and A2C had a heavy loss of -45.28%. In Chevron (CVX), Buy & Hold also outperformed PPO and A2C with a return of -0.91% compared to PPO's -12.29% and A2C's -0.74%.

These examples showcase Buy & Hold's relative advantage in minimizing losses when prices decline gradually. Its passive approach avoids unnecessary trades that can negatively affect the agent's performance when markets are either not moving much or are steadily declining. However, its lack of adaptability limits its ability to capitalize on favorable trends, making it less effective overall than PPO and A2C.

4.4 Notable Examples

- **AXP (American Express):** PPO and A2C both achieved their highest returns here (88.80% and 80.24%), likely due to the stock's strong bullish trend and price momentum, which allowed the agents to build gains by effectively identifying when to buy, hold and sell.
- **WBA (Walgreens Boots Alliance):** A2C suffered a large loss of -61.47%, likely from poor policy updates in a volatile/declining market, whereas PPO and Buy & Hold limited losses to around -19% and -1.65% respectively. This suggests A2C may have overreacted to noise or lacked generalization.
- **INTC (Intel):** All strategies performed poorly, with A2C returning -45.28%, PPO -3.14%, and Buy & Hold -1.50%. This points to an unstable market with no consistent direction, preventing the agents from learning effective trading signals.

4.5 Statistical Observations

- PPO had the highest average ROI (15.06%) and performed best overall.
- A2C had a slightly lower average ROI (14.11%) but showed greater volatility.
- Buy & Hold produced stable but lower returns (4.62% on average).
- PPO outperformed Buy & Hold in 77% of the stocks; A2C in 73%.

4.6 Possible Explanations

The performance differences between the different strategies can be explained by how each one responds to different market conditions.

- PPO's superior performance (highest average ROI and most consistent results) likely comes from its clipped objective function, which stabilizes learning and avoids large, unstable policy updates. This may have helped it adapt steadily to trending price patterns without overfitting to noise.
- A2C's volatility could be due to its synchronous update mechanism and higher sensitivity to short-term changes. In unstable or noisy markets (e.g., WBA, INTC), A2C may have overreacted to recent price movements, leading to suboptimal decisions and large losses.
- Buy & Hold's stable but lower returns reflect its passive nature: while it benefits from strong upward trends (e.g., AXP, CRM), it lacks the flexibility to respond to downturns or sideways markets. Its occasional outperformance in bearish or stagnant conditions suggests that avoiding overtrading can be advantageous when no clear signal exists.

Overall, these results highlight the trade-offs between DRL based strategies and passive investing (buying & holding) under different market environments.

4.7 Visual Comparison

The performance of each strategy across all 30 stocks is illustrated in Figure 4.1, which shows the ROI for PPO, A2C, and Buy & Hold side-by-side per ticker.

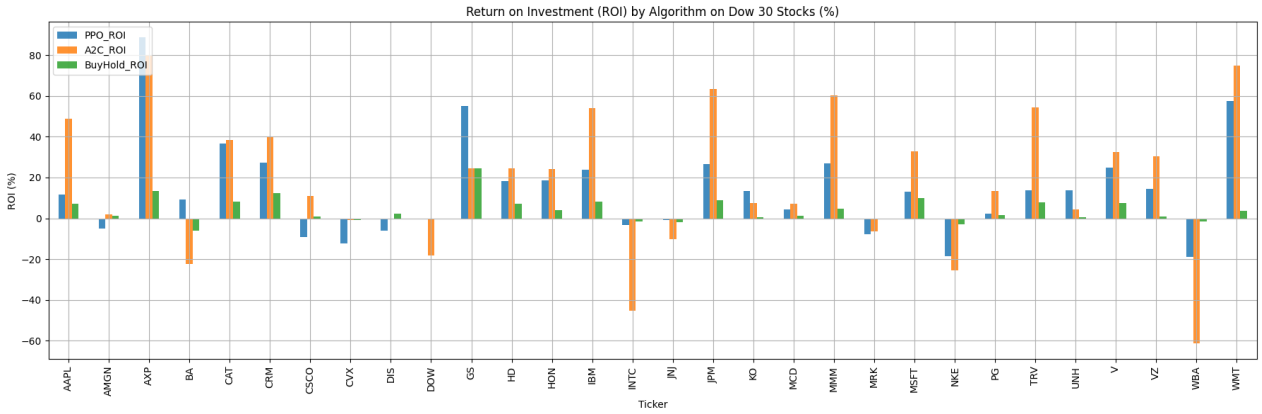

Figure 4.1: Return on Investment (ROI) Comparison per Strategy per Stock where A2C makes big profits but also some big losses and both PPO and A2C mostly outperform Buy & Hold.

The portfolio values for each strategy across all 30 stocks are illustrated in Figure 4.2. It shows the portfolio values side by side per ticker.

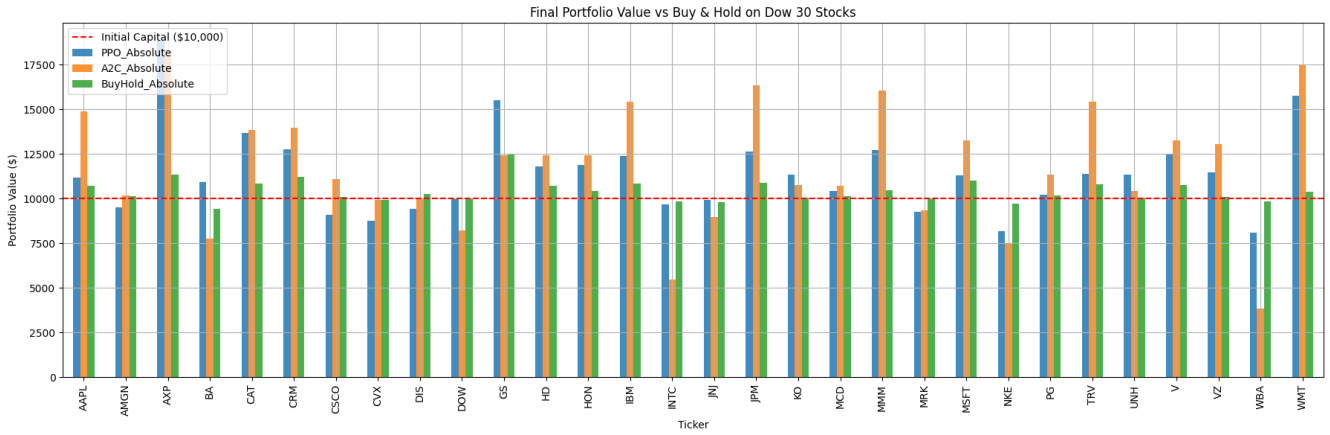

Figure 4.2: Final portfolio values per stock for PPO, A2C, and Buy & Hold, Where PPO and A2C mostly outperform Buy & Hold.

4.8 Interpretation

The results demonstrate that DRL agents, when trained individually per stock with an optimized state space and reward function, can consistently outperform the traditional Buy & Hold strategy. PPO in particular, it emerged as the most reliable performer, returning strong average rewards while keeping the risk of significant losses relatively low. Its stable policy updates and clipping mechanism likely contributed to its ability to adapt steadily to diverse market conditions without overreacting to noise or short-term fluctuations.

While A2C achieved high and competitive average rewards and outperformed Buy & Hold in 22 of 30 cases, it does show a larger variability in performance. Its vulnerability to significant losses such as -61.47% ROI on WBA can be linked to its more aggressive policy updates and lack of a stabilizing constraint like PPO's clipping. As a result, A2C was more likely to overfit and behave unpredictably in unstable or noisy markets.

These results suggest that while both DRL agents show strong potential for adaptive trading, PPO offers a better balance between return and risk, making it a better candidate for real-world trading where minimizing risk and preserving capital are essential. Furthermore, the consistently lower returns of Buy & Hold reinforce the value of dynamic, data-driven strategies in navigating varying market conditions.

4.9 Learning Dynamics Over Time

To further analyze agent behavior, we plot the average cumulative reward per timestep across all stocks during training and testing.

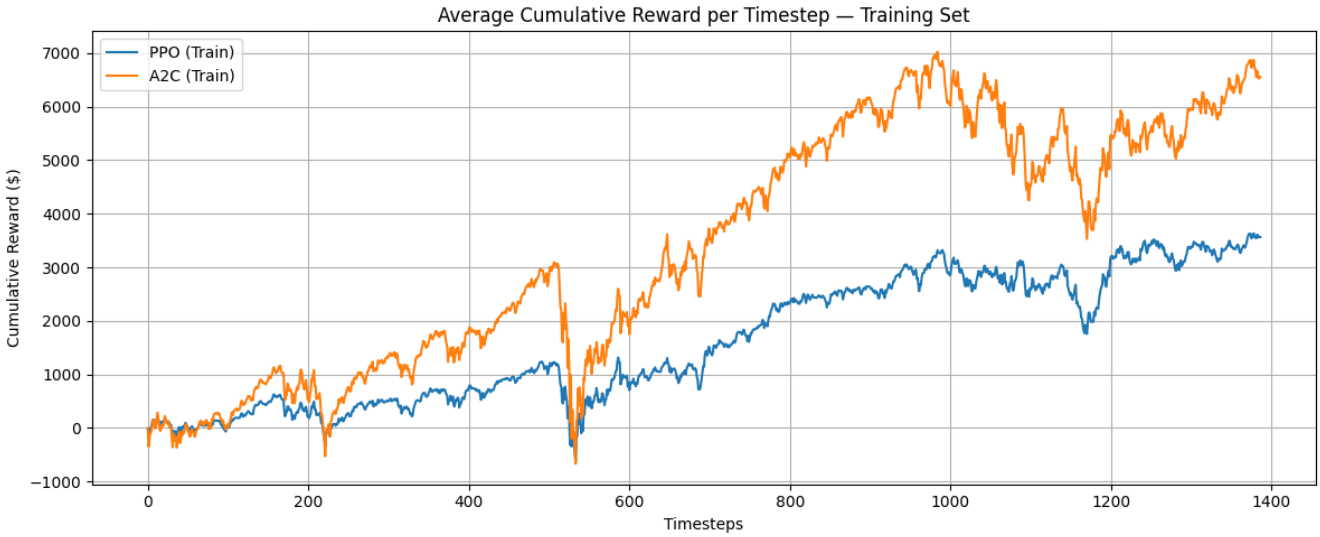

Figure 4.3: Final portfolio values per stock for PPO, A2C, and Buy & Hold. A2C accumulates rewards faster during training compared to PPO.

Figure 4.3 shows that A2C accumulates rewards faster during training compared to PPO, suggesting faster learning. However, this comes at the cost of greater variance and instability. PPO progresses more steadily, consistent with its clipped objective which stabilizes policy updates.

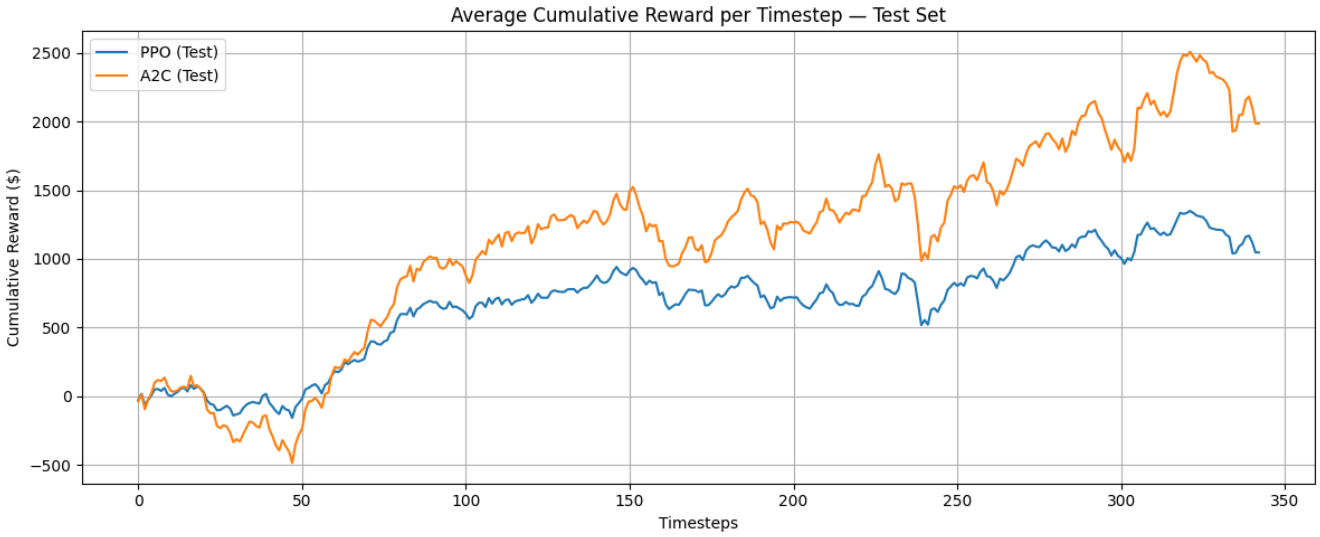

Figure 4.4: Final portfolio values per stock for PPO, A2C, and Buy & Hold. PPO generalizes more effectively across market condition while A2C may exploit certain trends more aggressively.

In Figure 4.4, we observe a continuation of this trend. A2C still accumulates more cumulative reward overall, but with increased volatility. PPO exhibits smoother and more controlled reward growth, indicating better generalization and risk control during unseen data.
These results align with our broader findings: while A2C may exploit certain trends more aggressively during training, PPO generalizes more effectively across market conditions, resulting in higher average ROI and more consistent performance in test environments.

The cumulative reward graphs were generated from a separate training run and may show slightly different results compared to the evaluation earlier, due to randomness in DRL training and different starting conditions. However, the overall patterns are consistent and support the same conclusions about the performance of PPO and A2C.

Chapter 5

Conclusion

This thesis investigated the application of DRL to stock trading, focusing on two popular policy gradient methods: PPO and A2C. The goal was to evaluate their performance in comparison to the traditional Buy & Hold strategy across the Dow 30 stocks.

5.1 Summary of Findings

Both PPO and A2C agents were trained and tested on a limited dataset of 30 stocks. Within the framework of this study, they showed promising results and seemed to demonstrate the ability to learn profitable trading behaviors from historical market data. On average, PPO achieved the highest returns, with a final portfolio value of $11,506.23 and an average ROI of 15.06%, followed closely by A2C with a final portfolio value of $11,411.01 and an average ROI of 14.11%. In contrast, Buy & Hold achieved final portfolio value of $10,462.43 and a modest average ROI of 4.62%.

In terms of consistency, PPO yielded a profit on 20 out of 30 stocks, while A2C and Buy & Hold were profitable on 20 and 22 stocks respectively. Notably, PPO outperformed Buy & Hold on 23 stocks, and A2C did so on 22 stocks. However, A2C showed higher volatility, with significant losses on some stocks (e.g., a -61.47% ROI on WBA), whereas PPO remained more stable across stocks.

It should also be noted that agents stronger performance might be highlighted due to Buy & Hold's limitations. While the DRL agents had the flexibility to adjust their positions and trade their full portfolio throughout the test period, the Buy & Hold strategy was limited to purchasing only 10 shares at the start and holding them until the end. This may have limited Buy & Hold's ability to adapt to changing market conditions, which could explain the difference in results.

5.2 Answering the Research Questions

Is reinforcement learning a suitable approach for financial market applications, particularly stock trading?

Yes, the results of this study suggest that RL , specifically algorithms like PPO and A2C, show promise as an approach for automated stock trading PPO outperformed Buy & Hold in 77% of the stocks, and A2C in 73%, both achieving significantly higher average returns (15.06% and 14.11%,) compared to Buy & Hold (4.62%). These agents were able to adapt their trading behavior based on recent market conditions, capturing short- and medium-term trends that static strategies could not exploit. The ability to make dynamic, data-driven decisions allowed the RL agents to succeed in bullish environments and (try to) avoid losses in moderately volatile markets. While the results are based on the specific dataset and setup used in this study, they provide a basis for future research in this area. Further research is highly recommended to better understand how well DRL methods perform across different financial settings and conditions.

Can reinforcement learning outperform traditional strategies such as Buy & Hold?

Yes, both PPO and A2C consistently outperformed Buy & Hold in terms of average ROI, profit, and the number of stocks where they delivered superior results. PPO achieved the highest average ROI (15.06%) and beat Buy & Hold in 77% of cases, showing strong generalization across diverse market conditions. These results suggest that RL agents can dynamically adjust to price trends and volatility, capturing gains that static strategies often miss. However, it is important to mention again that the Buy & Hold benchmark used in this study was limited to purchasing 10 shares at the start of the test period without any adaptation during this period. This may have contributed to the observed performance gap in favor of the RL agents.

Can a reinforcement learning trading agent be effectively deployed using real-world market data?

Yes, the agents were trained and evaluated using real historical price data from the Dow Jones 30 stocks and achieved strong performance. Both PPO and A2C successfully adapted to different market environments and have demonstrated that RL can effectively process real-world financial data to potentially generate profitable trading decisions. However, while these results are promising, further research is needed to evaluate how well such agents perform in other markets, time periods, and under different trading constraints.

5.3 Limitations and Future Work

Even though these results seem promising, this study has several limitations. The evaluation was limited to daily price data and a simplified trading environment without transaction costs. The Buy & Hold benchmark was also limited as it was only allowed to buy 10 shares at the start of the test period and is forced to hold them until the end. It can not choose to buy more or sell

current stock, which may have contributed to the performance gap observed in favor of the RL agents.

Future work could address these limitations and extend this research in several directions:

- Incorporating high-frequency (intra-day) or tick data,
- Introducing more realistic market features such as transaction costs,
- Using more advanced architectures (e.g., LSTMs, Transformers),
- Expanding the asset universe beyond the Dow 30 to include different sectors,
- Live testing on a paper or real-money trading account.
- Comparing RL agents to stronger baselines, such as momentum strategies or portfolio optimization methods.

5.4 Conclusion

Overall, this thesis demonstrates that RL, when carefully applied with stock-specific training, an optimized observation space and reward function that accounts for trading behavior, can outperform traditional strategies like Buy & Hold on real market data.

Both PPO and A2C agents have adapted to varying market conditions and achieved superior average ROI and profits across most Dow 30 stocks. These results highlight the potential of DRL to develop more adaptive and data-driven trading bots. For the financial industry, this opens the door for developing intelligent trading algorithms that can better respond to dynamic market conditions, particularly in environments where rule-based strategies underperform.

Appendix A: Hyperparameters and Training Configuration

This appendix outlines the training parameters, environment setup, and policy configurations used for the PPO and A2C agents in this research. All implementations relied on the `stable-baselines3`[RHG+21] library with its default settings unless stated otherwise.

General Settings

- Framework: `stable-baselines3` (PyTorch backend)
- Policy Network: `MlpPolicy` (Multilayer Perceptron)
- Network Architecture: 2 hidden layers of 64 units each (ReLU activation)
- Initial Capital: \$10,000
- Training Timesteps per Stock: 50,000

Environment Configuration

- Observation Space: 32-dimensional vector (OHLC, SMA5, SMA20, cash, shares)
- Action Space: Continuous action in $[-1, 1]$ (buy/sell/hold fraction)
- Reward Function:

$$\text{Reward}_t = \Delta\text{Portfolio Value} - \lambda \cdot \text{Shares Traded}, \quad \lambda = 0.01$$

- Lookback Window: 5 days

PPO Hyperparameters (Defaults)

- Learning Rate: 3×10^{-4}
- Discount Factor (γ): 0.99
- GAE Lambda (λ): 0.95
- Clip Range (ϵ): 0.2
- Batch Size: 64
- Number of Epochs: 10
- Value Function Coefficient: 0.5
- Gradient Clipping: 0.5

A2C Hyperparameters (Defaults)

- Learning Rate: 7×10^{-4}
- Discount Factor (γ): 0.99
- Entropy Coefficient: 0.01
- Value Function Coefficient: 0.5
- Number of Steps per Update: 5
- Gradient Clipping: 0.5

Notes

- PPO used Generalized Advantage Estimation (GAE) for variance reduction.
- A2C used synchronous updates from multiple environments.
- No custom exploration strategy (e.g., ϵ-greedy) was used. Exploration was handled by the stochastic nature of the policies.

Bibliography

[ABC+20] Marcin Andrychowicz, Bowen Baker, Maciek Chociej, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[BGW+24] Yahui Bai, Yuhe Gao, Runzhe Wan, Sheng Zhang, and Rui Song. A review of reinforcement learning in financial applications. *Annual Review of Statistics and Its Application*, 2024.

[Cha08] Ernest P. Chan. *Quantitative Trading: How to Build Your Own Algorithmic Trading Business*. John Wiley & Sons, 2008.

[Fou23] Farama Foundation. Gymnasium: A modern reinforcement learning platform. https://github.com/Farama-Foundation/Gymnasium, 2023. Accessed: 2025-06-28.

[HXY21] Ben Hambly, Renyuan Xu, and Huining Yang. Recent advances in reinforcement learning in finance. *Mathematical Finance*, 2021.

[KSS+23] Abhishek Kumar, Pawan Sharma, Ramesh Singh, Gaurav Yadav, et al. Reinforcement learning in energy finance: A comprehensive review. *Energies*, 18(11):2712, 2023.

[KT79] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979.

[LdPSFF24] Marcos López de Prado, Joseph Simonian, Francesco A. Fabozzi, and Frank J. Fabozzi. Enhancing markowitz's portfolio selection paradigm with machine learning. *Annals of Operations Research*, 346(1):319–340, 2024.

[LHP+15] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[Lo05] Andrew W. Lo. Reconciling efficient markets with behavioral finance: The adaptive markets hypothesis. *Journal of Investment Consulting*, 7(2):21–44, 2005.

[MA18] Bruno Molinari and Marco Avellaneda. A bayesian network using partially observable orders imbalance. ICMA Group Whitepaper, 2018. https://www.icmagroup.org/assets/Uploads/Bayesian.pdf.

[Mar52] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[MBM+16] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 1928–1937. PMLR, 2016.

[MKS+15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[RHG+21] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. https://github.com/DLR-RM/stable-baselines3, 2021. Accessed: 2025-06-28.

[SB18] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018. Available online at http://incompleteideas.net/book/RLbook2020.pdf.

[Shl00] Andrei Shleifer. *Inefficient Markets: An Introduction to Behavioral Finance*. Oxford University Press, 2000.

[SML+15] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[SWD+17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[TGL13] Philip Treleaven, Mark Galas, and Sanjay Lalchand. Algorithmic trading: The play-at-home version. *Communications of the ACM*, 56(11):76–85, 2013.

[YLZW20] Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. Deep reinforcement learning for automated stock trading: An ensemble strategy. *SSRN Electronic Journal*, 2020.

[ZLZW24] Xiaoyu Zhang, Mingyu Li, Yifan Zhao, and Jun Wang. Industry-grade deep reinforcement learning for portfolio optimization. *arXiv preprint arXiv:2403.07916*, 2024.