ECE1508 Final Report

Mahdi Alaeikerahrudi
Department of Mechanical & Industrial Engineering
University of Toronto
matt.alaeikerahrudi@mail.utoronto.ca

Anuj Chavan
Department of Electrical & Computer Engineering
University of Toronto
anuj.chavan@mail.utoronto.ca

Maggie Ye
Department of Electrical & Computer Engineering
University of Toronto
maggie.ye@mail.utoronto.ca

Abstract

In recent years, with the progress and development of deep neural networks and its integration with reinforcement learning (RL), there has been an ever growing interest in leveraging DRL in the field of finance, mainly for portfolio optimization. In this project, we investigate the performance of deep reinforcement learning (DRL) in the field of portfolio management. The efficacy of different DRL models is measured by comparing the results obtained via DRL with the results obtained through simple and non-RL heuristics and models.

Assentation of Teamwork

Mahdi: Helped creating the environment; created, trained, and tested PPO; created and tested MVO; generated the graphs and tables comparing the results; helped write the progress report, presentation, and final report.

Anuj: Created, trained and tested A2C, ACER; Created the presentation, slides, helped write the progress report, and final report.

Maggie: Helped creating the environment; created, trained and tested DDPG; created and tested Equal Weighting and Buy-and-Hold; helped write the progress report, presentation, and final report.

1 Introduction

Portfolio management is the action of continuous allocation of capital into a number of financial assets. In this setting, time is divided into equal periods of length T. In the beginning of each period, the agent allocates funds to each asset. The aim of the agent is to maximize the expected return of the portfolio, while minimizing the risk.

2 Preliminaries and Problem Formulation

In this section, we will denote the mathematical definition of a portfolio management problem. Assume that our portfolio consists of m different assets. Let x_i denote the amount of fund allocated to an asset $\forall i = \{1, ..., m\}$. Then the weight of an asset i is defined as $w_i = \frac{x_i}{\sum_{j=1}^{m} x_j}$.

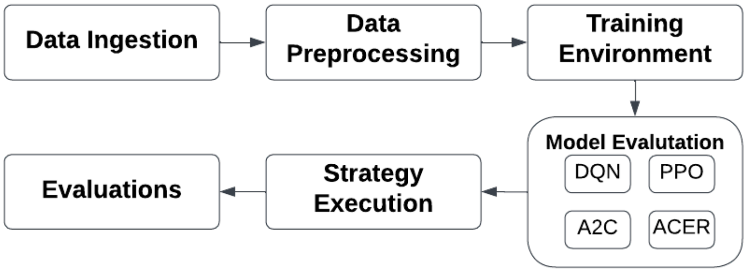
Let $\bar{r}_{t,i}$ denote the expected log-return of asset i at time t. Let $\sigma_{t,ij}$ denote the covariance between the expected log-returns of asset i and j at time t. Finally, let κ be the risk aversion factor. Then the portfolio management problem can be formulated as follows:

$$\begin{aligned} maximize\ & \textstyle\sum_{i=1}^{m} w_i \bar{r}_{t,i} - \kappa \sum_{i=1}^{m} \sum_{j=1}^{m} w_i \sigma_{t,ij} w_j \\ subject\ to\ & \textstyle\sum_{i=1}^{m} w_i = 1 \quad w_i \geq 0\ \forall i = \{1, .., m\} \end{aligned}$$

Although the above is the most widely known form of MVO, in our implementation, we will change the objective function to maximize the Sharpe Ratio [7]. The benefits of this change are two fold: Firstly, maximizing the Sharpe Ratio aligns with the reward function used in our DRL models (see section 3.3). Secondly, using Sharpe Ratio will free us from setting an arbitrary risk aversion κ.

3 Design

Our framework for stock trading follows the pipeline as shown below. First, historical market data are ingested from external sources and passed through a preprocessing stage where they are cleaned, normalized, and converted into state representations suitable for agent consumption. These processed features are then fed into a dedicated trading environment that defines the state space, available trading actions, and reward formulation. Within this environment, we train and benchmark multiple RL algorithms under identical market conditions. Once trained, each agent's learned policy is executed as a trading strategy to generate actionable signals. Finally, strategy performance is evaluated on out-of-sample data using standard financial metrics.



3.1 Dataset

Our historical equity price dataset was composed of daily OHLCV (open, high, low, close, volume) time series from 2005-2025. Data was pulled from the Yahoo! Finance APIs using the Python yfinance library [1]. We created two datasets. The first contained 5 assets from the major large-cap technology firms (MAANG: Microsoft, Apple, Amazon, Nvidia, and Google). The second dataset sampled the S&P 500 and took 3 companies from each of the 11 GICS sectors, for a total of 33 assets (See Appendix) [2]. Each dataset was partitioned into rolling windows of 7 years [7], where the first 5 years were used for training, the 6th year was used as validation and the last year was used for testing. The training split was used for agent policy learning, the validation split was used for hyperparameter selection and early stopping, and the held-out test split was reserved strictly for final performance evaluation to avoid information leakage.

3.2 Choice of Algorithms and Motivation

We employed model-free DRL algorithms to develop the trading agent capable of learning optimal trading strategies from historical values of stock market data. We explored four DRL algorithms:

Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimisation (PPO), Advantage Actor-Critic (A2C), and Actor-Critic with Experience Replay (ACER) [6].

3.3 Environment

For each time step, the agent chose n weights for each asset in the portfolio $[w_1, w_2, ...w_n]$, where $-10 \leq w_i \leq 10$. These values were then converted to a distribution between 0 and 1 using softmax, such that $\sum_{i=1}^{n} w_i = 1$. A weight of 0 meant that the agent held zero shares of that asset, whereas a weight of 1 meant that the agent's entire portfolio consisted of that asset. As the weights are continuous values, they were rebalanced slightly after the agent's decision so only whole shares of each asset were purchased while still having the weights sum to 1.

We used a simplified observation matrix from [7], which has dimensions $[n \times T]$, where T is the length of the lookback period (60 days in our case). The matrix was formed as such:

$$S_t = \begin{bmatrix} w_1 & r_{1,t-1} & \cdots & r_{1,t-T+1} \\ w_2 & r_{2,t-1} & \cdots & r_{2,t-T+1} \\ \vdots & & \ddots & \vdots \\ w_n & r_{n,t-1} & \cdots & r_{n,t-T+1} \end{bmatrix}$$

The first column represents the agent's weighted portfolio allocation. The remaining values $r_{i,t-1}...r_{i,t-T+1}, 1 \leq i \leq n$ represent the log returns for each asset i during each timestep during the lookback period, and $r_t = log(\frac{p_t}{p_{t-1}})$, where p_t represents the price of an asset at time t.

We used two reward functions to conduct an ablation study. The first function was part of our environment modification and maximized risk-adjusted returns by incorporating the Differential Sharpe Ratio (see Appendix) [5]. The second function calculated the portfolio's simple return - in other words, $\frac{p_{t+1}-p_t}{p_t}$, where p_t is the value of the portfolio at time t.

We used three heuristic models to compare against our algorithms: Equal Weighting, Buy-and-Hold, and Mean-Variance Portfolio Optimization [7]. Finally, we evaluated our algorithms' performance against heuristic baselines by calculating Cumulative Annual Return, Sharpe Ratio, and Maximum Drawdown.

4 Methodology

Each DRL model was trained on the training data, fine tuned on the validation data and finally tested on the test set. We used StableBaseline3 implementation of PPO and DDPG [8], and pytorch for A2C and ACER.

For training each DRL model, on each training window, multiple different seeds were trained to reduce the stochasticity of our models. Each seed was trained on multiple parallel environments, so that the model could explore more and gather more experience. For this purposes, SubProcVecEnv was used (also provided by StableBaseline3). After training, each seed was tested on the validation data, and the best performing seed was saved and tested on the test set of the respective training window later on. Additionally, this seed was chosen as a seed policy for the next training window [7]. Next, we discuss the training process.

At each time step (day t), the model produced the weights of each asset and the number of whole shares for each asset was calculated given the weights. Afterwards, the time step was advanced (day $t + 1$) and the new prices of the assets were obtained, the new portfolio value was computed given the new prices, and finally the portfolio return and subsequently the reward of the model were calculated.

For testing purposes, each model was given a budget of $100,000 at the start of each test year (the test starts on the 60th day of the test year, since the first 59 days are passed as observation to the agent). The same process as training was repeated, and the relevant metrics from section 3.3, were calculated from the obtained daily portfolio values through the test year.

5 Numerical Experiments

The hyper-parameters used for each DRL model are given in the Appendix. The performance of each model is given in Table 1 (see also Figures 1, 4, and 5 in the Appendix). The results will be discussed in section 6.

Table 1: Results on MAANG using Differential Sharpe Ratio

Metric	PPO	DDPG	A2C	ACER	EW	BH	MVO
Annual Return	**0.350**	0.278	0.249	0.250	0.280	0.249	0.231
Annual Volatility	0.324	0.292	0.276	0.275	0.319	0.272	**0.259**
Max Drawdown	-0.383	**-0.356**	-0.408	-0.393	-0.379	-0.397	-0.365
Sharpe Ratio	**1.720**	1.403	1.321	1.329	1.306	1.325	1.177

5.1 Ablation studies

After choosing the best-performing DRL model (PPO) from our first experiment, we investigated the impact of our modified reward function by replacing it with a simple reward that maximizes only the return. The results are given in Table 2 (see also Figures 2 and 6 in the Appendix), which will be discussed in section 6.

Table 2: Results on MAANG using Simple Return

Metric	PPO	EW	BH	MVO
Annual Return	**0.304**	0.280	0.249	0.231
Annual Volatility	0.317	0.319	0.272	**0.259**
Max Drawdown	-0.393	-0.379	-0.397	**-0.366**
Sharpe Ratio	**1.519**	1.306	1.325	1.177

6 Discussion

Given the results, we make the following observations: Firstly, using DSR has a noticeable impact on the performance of the DRL model, as the Sharpe Ratio and annual return of PPO dropped by 11% and 13% respectively, when using simple return as the reward function. Secondly, even though PPO outperformed every model, the models performed very similarly. Lastly, MVO performed the worst out of the baseline heuristics, which aligns with the literature, as it has been shown that MVO in this format is not an effective strategy [4].

Table 3: Pearson correlation coefficient between MAANG

Ticker	AAPL	AMZN	GOOG	MSFT	NVDA
AAPL	1	0.933	0.979	0.988	0.811
AMZN	0.933	1	0.959	0.945	0.713
GOOG	0.979	0.959	1	0.984	0.804
MSFT	0.988	0.945	0.984	1	0.833
NVDA	0.811	0.713	0.804	0.833	1

We suspect the main reason behind the second observation is that MAANG assets are highly correlated (see Table 3), meaning that there is little to be learned from and a strategy as simple as equal weighing will do surprisingly well. Additionally, high correlation leads to poor diversification of portfolio. This means that when the price of one asset plummets, the prices of all other assets will also go down, and consequently the value of the portfolio will decrease (because there is no other option than buying these plummeting assets for our agents). However, we expect that the latter could have been prevented had shorting been allowed, since by allowing shorting, the agent would be able to leverage the plummeting assets and still increase the value of the portfolio.

To test the above hypothesis, we trained PPO on our second dataset (see the Appendix and Section 3.1) and tested it against the baseline models. The results are given in Table 4 (also see Figures 3 and 7 in the Appendix).

The results affirm our belief, as we both see a significant jump in PPO's performance compared to when using only MAANG assets (the annual return has almost tripled and Sharpe Ratio has more than doubled), and the performance of PPO is much better than our baseline models. This goes to show that to best leverage a DRL model, we must ensure that the set of assets is diverse.

Table 4: Results on S&P using Sharpe Ratio

Metric	PPO	EW	BH	MVO
Annual Return	**1.128**	0.105	0.105	0.066
Annual Volatility	1.522	0.159	**0.152**	0.202
Max Drawdown	-0.297	-0.235	**-0.231**	-0.270
Sharpe Ratio	**3.916**	0.891	0.936	0.457

Across all three experiments, there was not much variation in Maximum Drawdown values for any of the models. Since Maximum Drawdown measures the largest decline in an investment over a given time period, we suspect the reason for this is because shorting was not allowed in our environment. This meant that an agent would be forced to hold on to an asset even as its value decreased.

7 Conclusions

Although our initial results were underwhelming, we have shown the very promising potential of utilizing DRL for portfolio optimization in the more diverse set of assets. Given this potential, we discuss some of the improvements that can be investigated in future works.

In our work our state representation was very simple. While this state representation is identical to the data that is assumed to be available in classical methods (such as MVO), we are not leveraging the full power of deep neural networks. We suspect that incorporating an improved and more comprehensive state representation, for example by adding volatility metrics or the relative strength index, could dramatically improve the performance of DRL models.

In our setting, we assumed that there were no transaction costs, which is highly idealized and can lead to highly inaccurate performance simulations. Hence, including transaction costs can allow us to evaluate the performances of DRL models more realistically.

Finally, we suspect that allowing shorting would lead the model to perform and make more intelligent trading strategies even if the portfolio consists only of highly correlated assets.

References

[1] Ran Aroussi. *yfinance*. URL: `https://github.com/ranaroussi/yfinance`.

[2] MSCI Inc. *The Global Industry Classification Standard (GICS®)*. URL: `https://www.msci.com/indexes/index-resources/gics`.

[3] Peng X Jia Z Gao Q. “STM-DDPG for Trading with Variable Positions”. In: *Sensors* (2021). DOI: `10.3390/s21196571`.

[4] Richard O. Michaud. “The Markowitz Optimization Enigma: Is ’Optimized’ Optimal?” In: *Financial Analysts Journal* 45.1 (1989), pp. 31–42. ISSN: 0015198X. URL: `http://www.jstor.org/stable/4479185`.

[5] J. Moody and M. Saffell. “Learning to trade via direct reinforcement”. In: *IEEE Transactions on Neural Networks* 12.4 (2001), pp. 875–889. DOI: `10.1109/72.935097`.

[6] Prasanna Kumar R et al. “Analyzing Deep Reinforcement Learning Strategies for Enhanced Profit Generation and Risk Mitigation in Algorithm Stock Trading”. In: *2023 6th International Conference on Recent Trends in Advance Computing (ICRTAC)*. 2023, pp. 766–771. DOI: `10.1109/ICRTAC59277.2023.10480823`.

[7] Srijan Sood et al. “Deep Reinforcement Learning for Optimal Portfolio Allocation: A Comparative Study with Mean-Variance Optimization”. In: URL: `https://api.semanticscholar.org/CorpusID:264432026`.

[8] *Stable-Baselines3 Docs - Reliable Reinforcement Learning Implementations*. URL: `https://stable-baselines3.readthedocs.io/en/master/index.html`.

Appendix

S&P Dataset Companies

- **Industrials:** Caterpillar Inc. (CAT), CSX Corporation (CSX), Rockwell Automation (ROK)
- **Health Care:** Boston Scientific (BSX), Cigna (CI), Danaher Corporation (DHR)
- **Information Technology:** Applied Materials (AMAT), Micron Technology (MU), Qualcomm (QCOM)
- **Utilities:** Edison International (EIX), PPL Corporation (PPL), Xcel Energy (XEL)
- **Financials:** BNY Mellon (BK), KeyCorp (KEY), Travelers Companies (The) (TRV)
- **Materials:** Nucor (NUE), PPG Industries (PPG), Vulcan Materials Company (VMC)
- **Consumer Staples:** Campbell's Company (The) (CPB), Kroger (KR), Target Corporation (TGT)
- **Energy:** Devon Energy (DVN), Schlumberger (SLB), Valero Energy (VLO)
- **Communication Services:** Electronic Arts (EA), AT&T (T), Verizon (VZ)
- **Consumer Discretionary:** Marriott International (MAR), TJX Companies (TJX), Yum! Brands (YUM)
- **Real Estate :** Equity Residential (EQR), Prologis (PLD), Weyerhaeuser (WY)

Differential Sharpe Ratio

The Differential Sharpe Ratio (DSR) [5] is defined as follows:

$$D_t = \frac{B_{t-1}\Delta At - \frac{1}{2}A_{t-1}\Delta B_t}{(B_{t-1} - A_{t-1}^2)^{3/2}}$$

With

$$A_t = \frac{1}{t}\sum_{i=1}^{t} R_i,\ B_t = \frac{1}{t}\sum_{i=1}^{t} R_i^2,$$
$$A_t = A_{t-1} + \Delta A_t,$$
$$B_t = B_{t-1} + \Delta B_t,$$
$$\Delta A_t = R_t - A_{t-1},$$
$$\Delta B_t = R_t - B_{t-1}$$

Here R_t is the return of our portfolio at time t. In addition, we have $A_0 = B_0$ and $\eta = \frac{1}{252}$ (a year has almost 252 work days) [7].

Hyper-Parameters

Table 5 lists the hyper-parameters all DRL models have in common. These were picked based on empirical evidence [7],[3].

Table 5: Model Hyper-Parameters

Model	Gamma	Learning Rate	Batch Size	n_env	n_seeds	n_episodes	n_steps
PPO	0.9	3e-4 annealed to 1e-5	252	5	3	50	1260
DDPG	0.8	3e-4 annealed to 1e-5	128	10	2	100	2520
A2C	0.9	3e-4 (const)	on-policy (full-batch)	5	2	50	1260
ACER	0.9	3e-4 (const)	on-policy (full-batch)	5	2	50	1260

In Table 5, n_env is the number of environments that were used to train the models, n_seeds is the number of different (seed) agents trained, and n_steps is the rollout buffer size, computed as $252 \times n\_env$ for each model.

The other hyper-parameters for PPO were: clip_range=0.25, gae_lambda=0.9, tanh as the activation function, and 3 hidden layers of size [32, 64, 32].

The other hyper-parameters for DDPG were: tau=0.01, the number of critics set to 1, tanh as the activation function, and 2 hidden layers of size [64, 32].

Performance Graphs

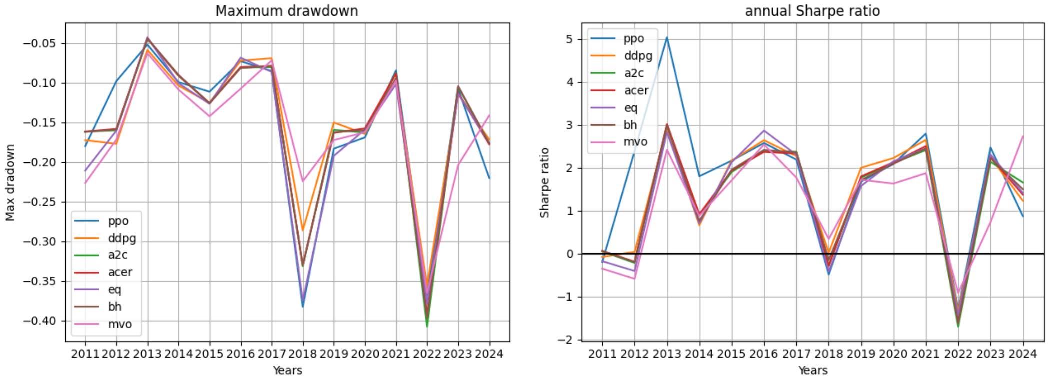


Figure 1: Comparison of Max Drawdown and Sharpe Ratio (on MAANG assets)

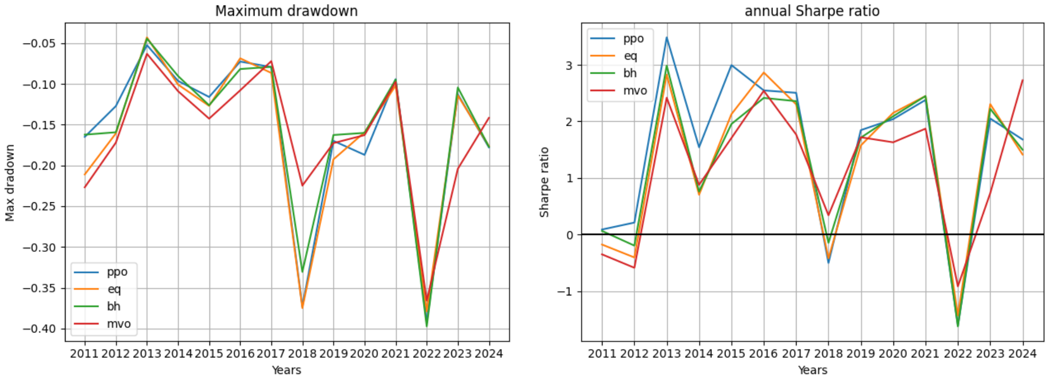


Figure 2: Comparison of Max Drawdown and Sharpe Ratio (on MAANG assets with simple return)

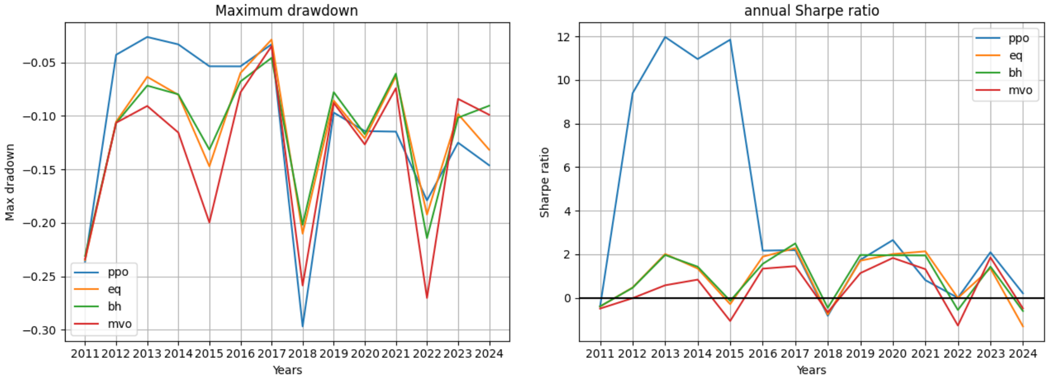


Figure 3: Comparison of Max Drawdown and Sharpe Ratio (on S&P data)

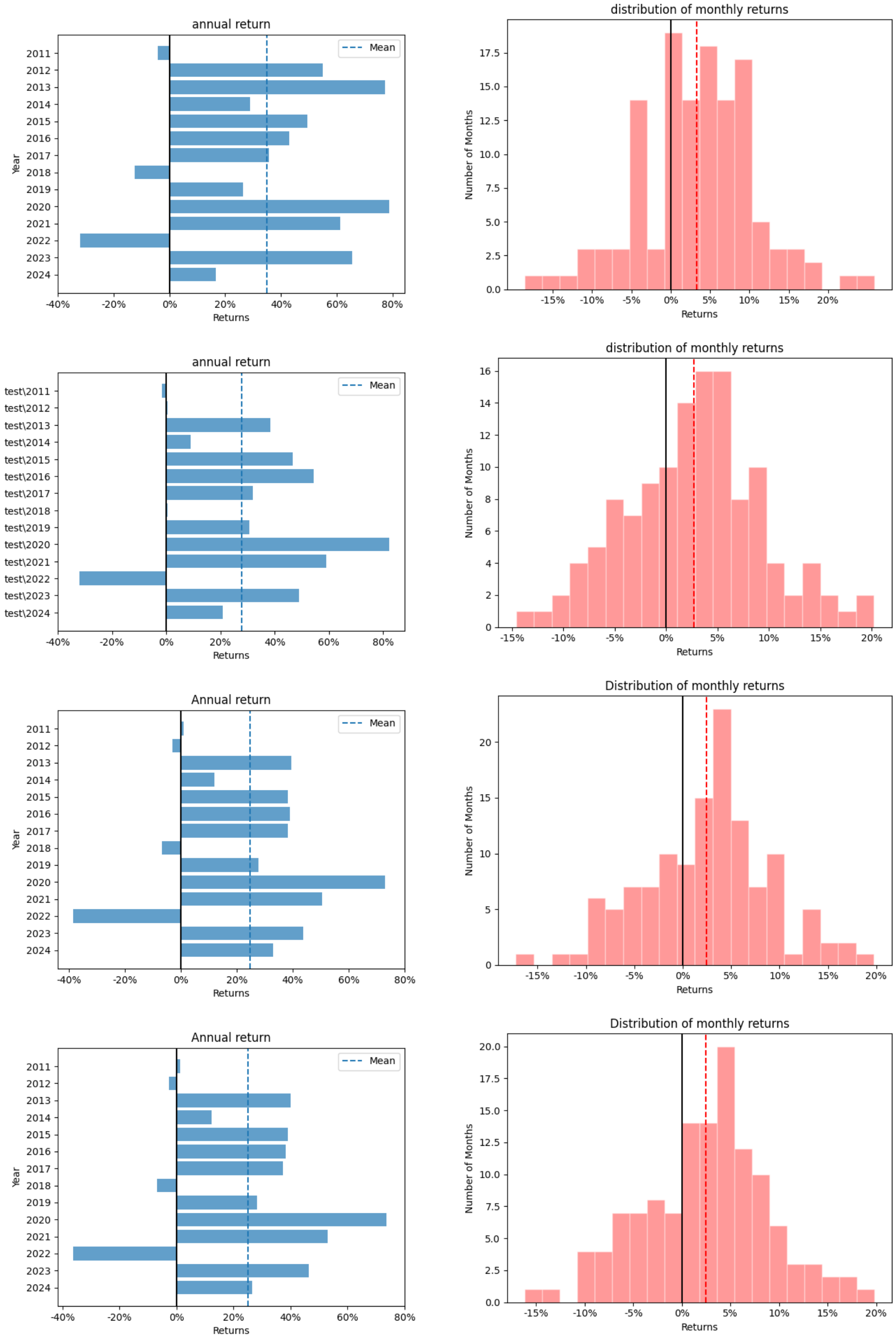


Figure 4: Performance metrics (annual returns and monthly return distributions) for all evaluated strategies on MAANG assets: PPO, DDPG, A2C, ACER (current page), Equal Weight, MVO, and Buy & Hold (next page) respectively.

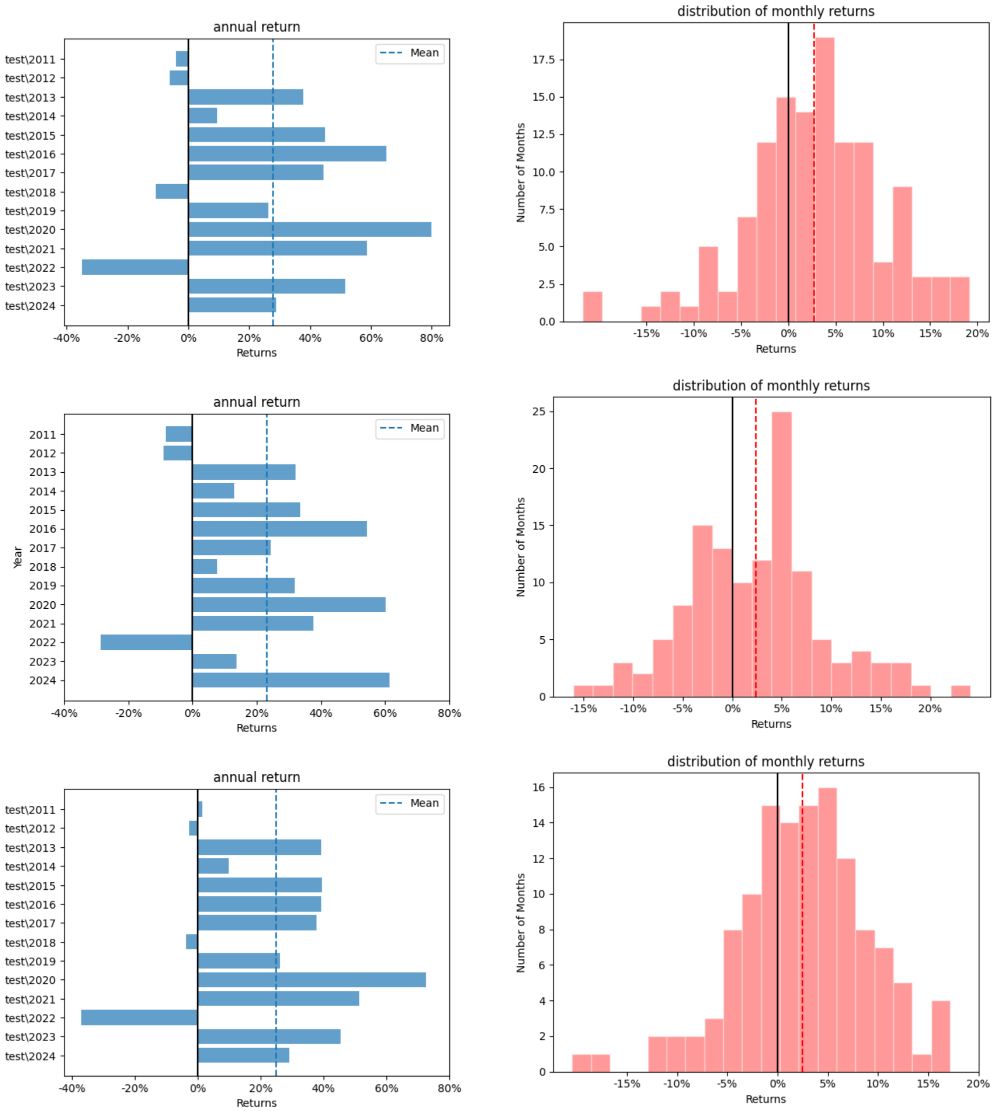


Figure 5: Performance metrics (annual returns and monthly return distributions) for all evaluated strategies on MAANG assets: PPO, DDPG, A2C, ACER (previous page), Equal Weight, MVO, and Buy & Hold (current page) respectively.

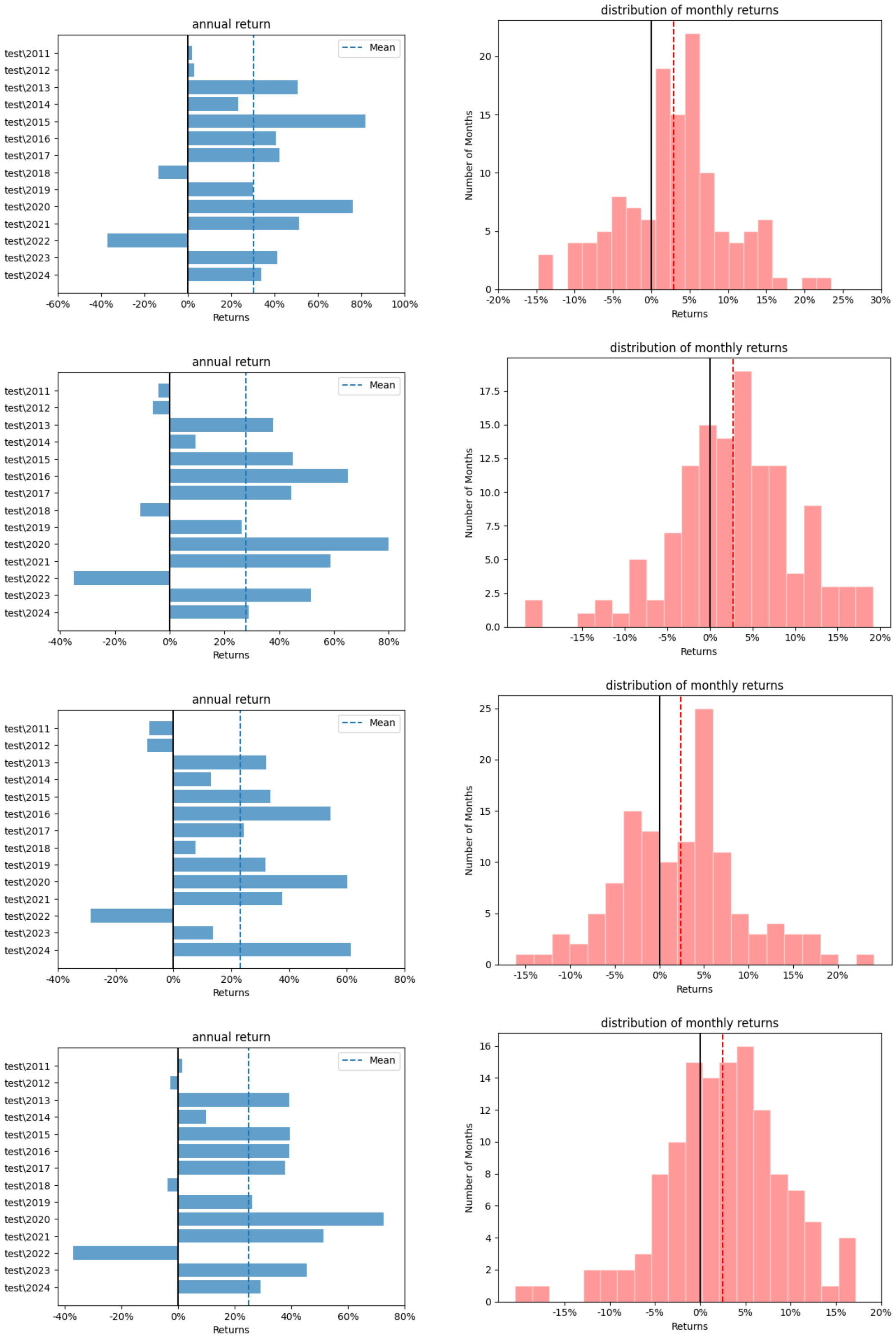


Figure 6: Performance metrics (annual returns and monthly return distributions) for all evaluated strategies on the simple return reward function and the MAANG assets: PPO, Equal Weight, MVO, and Buy & Hold respectively.

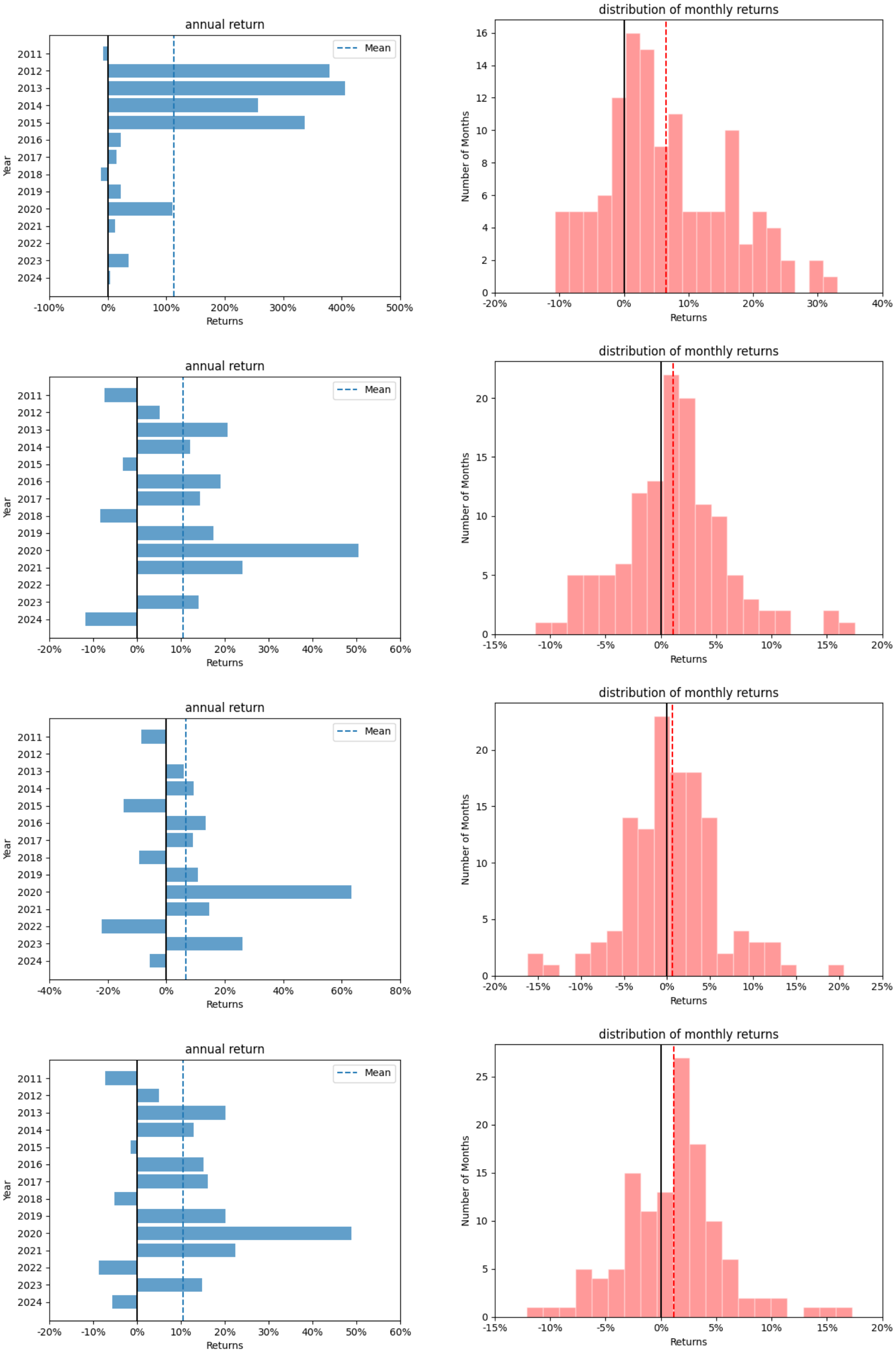


Figure 7: Performance metrics (annual returns, and monthly return distributions) for all evaluated strategies on the S&P data: PPO, Equal Weight, MVO, and Buy & Hold respectively.

Consent for Information Sharing

As part of this course, we may share selected project materials (e.g., reports, presentation slides, and presentation recordings) on the course webpage as learning resources for future students. Additionally, we may use anonymized project information for internal statistical analysis of course outcomes. Please indicate your preferences below. *Your choices will not affect your grade in any way.*

Consent for Sharing Project Materials

Please keep the item you wish and remove the other one.

- All group members consent to allow the project materials (report, slides, and presentation recording) to be shared on the course page for future students.

Consent for Use of Project Information in Statistical Analysis

Please keep the item you wish and remove the other one.

- All group members consent to allow anonymized information about the project (e.g., topic, methods, outcomes, grades) to be used by the instructor for statistical analysis and course improvement.

Optional Comments

Please list any specific conditions or comments here.

Group Identification

- Group number: 19
- Names of group members: Mahdi Alaeikerahrudi, Anuj Chavan, Maggie Ye
- Signature of of group members: Mahdi Alaeikerahrudi, Anuj Chevan, Maggie Ye
- Date: December 7, 2025