Extended Abstract

Motivation Understanding and trading stock market trends poses major challenges to both professional traders and automated agents alike. Different strategies of predicting trends and movement have been suggested including trading based on historical performance, gap trading or trading liquidity. Reinforcement learning agents have been trained on these strategies, but they treat each asset as its own individual entity and ignore the correlation between different assets that may arise due to related sectors. Graph Neural Networks have the power to encode such relationships, but have been an understudied area within this domain. Our goal in this work is to explore whether encoding inter-stock correlations through a GNN and passing it into an RL policy can yield better returns as compared to the market or an equiweight buy-and-hold strategy.

Method We first begin by constructing a graph where each of the nodes represents eight highly liquid tickers in the US Stock exchange (NASDAQ). These stocks are, namely, AAPL, MSFT, NVDA, JPM, XOM, TSLA, META, and AMZN. Edge weights between nodes are calculated as the Top-K (k=4) absolute value of a 60-day rolling average Pearson correlation matrix with a one-day stride. The nodes contain information on the short and medium-horizon returns, simple-moving-average ratios, RSI-14 ratios, market cap, and rolling volume. Each day is recorded as a snapshot and passed through a Graph Attention Network that produces a 64-dim embedding for each node. All embeddings from all 8 nodes are then concatenated to produce a 512 embedding vector which is passed into the PPO agent. The actor outputs continuous logits which are mapped via sigmoid and normalization long-only portfolio weight actions, and the critic supplies the value function estimate for computing advantages.

Implementation Historical data from 2015-present are downloaded from Yahoo Finance. The OHLCV (Open, High, Low, Close, Volume) are extracted for each stock each day. The graph snapshots are stored with features, edge indices, weights, and the calendar date of the snapshot. Training was done using stable-baselines 3's PPO agent and the rewards were clipped to 0.5. The model is trained on a dataset from 2015-2022 and validated on the data from 2023-2025.

Results During the approximately 600-day held-out period from 2023 to 2025, the agents grew the portfolio from $1,000,000 to approximately $3,500,000 which is about a 250% return. An equal-weight buy-and-hold strategy has a return of around 200% whereas the markets saw returns of around 50%. Furthermore, we see that the Sharpe Ratio was higher in the RL model as compared to buy-and-hold strategies as well as being less volatile. The 1.91 Sharpe Ratio of the PPO agent outperformed the buy-and-hold strategy's Sharpe Ratio of 1.84. The -28.46% max drawdown of the PPO agent also performs slightly better than the -28.91% max drawdown of the buy-and-hold strategy.

Discussion There are a couple of factors that we believe led to this result. First off, the results show that creating graphs and embeddings that encapsulate the relationships between tickers creates a model that performs at least as well as buy-and-hold strategies. We see in the results, that the GAT was able to highlight decouplings between stocks especially within similar sectors. For example, we see the agent act in a particular way to NVDA that was separate from other similar tech-related assets. We also clipped log-return which prevented explosion of gradients while successfully promoting the actions of compounding interest. Some limitations are the use of a relatively small stock universe of n=8 stocks and the absence of real-time data on how others react when large buy/sell orders are placed by this agent.

Conclusion Encoding inter-asset relationships as a dynamic graph and training a graph-aware PPO agent produces a model that has higher returns, lower max draw down, and higher Sharpe value. This experiment demonstrated that relations between stocks can be embedded as graphs, and then harnessed by modern RL architectures are able to create adaptive portfolio strategies that result in better actions.

Graph-based Stock Market RL Agent

Nevin Aresh
Department of Biomedical Data Science
Stanford University
naresh6@stanford.edu

Abstract

We present a graph-aware Proximal Policy Optimization (PPO) network that is able to turn inter-stock correlations into actionable portfolio weights and outperforms classical trading baselines. Data is collected eight high liquidity US equities from 2015-2025 and each trading day is encoded as a graph where the nodes contain eight engineered indicators and the edge weights are calculated using Pearson correlation of 60-day log returns. A two later Graph Attention Network (GAT) is then used to produce an embedding that is fed to a PPO actor-critic where only long-actions can be taken and the rewards are calculated as a clipped daily log-return. The model was trained on 2015-2022 and then the GNN-PPO agent was able to grow $1,000,000 to about $3,500,000 with a high Sharpe ratio and a minimized maximum draw down. We saw that the GNN-PPO model was able to outperform the baselines and results showed that a shorter training window (2019-2022) was able to perform better than a 2015-2022 training window. These results confirm that modeling cross-asset relationships using GANs have the capability to enhance returns, and can be expanded to a universe with more stocks, more adaptive edge definitions and more complex reward designs.

1 Introduction

The stock market is a complex system of moving parts that many have tried to model and predict. Strategies such as buy low, sell high have been taught to many since a young age, but without knowledge of future prices, it is nearly impossible to predict exactly when the lows and highs will occur. Strategies such as using historical trends and trading between liquidity zones are common strategies used nowadays to predict the movement of prices. Traditional quantitative strategies, such as these, typically ignore, simplify, or treat the relationships between stocks as stationary. Oftentimes, these strategies focus on indicators that are applied to each individual stock rather than trying to model relationships between different tickers as financial markets are adaptive systems where the price of one security rarely moves in isolation; usually, an entire sector moves together. Thus, many of today's agents struggle when movements and shifts in the market are driven by cross-asset dependencies.

Graph Neural Networks (GNNs) which were introduced in 2005 (Khemani et al. (2024)) provides a competitive way to model these dependencies. By representing the tickers and associated properties as nodes and calculating the rolling correlation between nodes as weighted edges, we are able to reveal both intra- and inter-sectoral structures as well as couplings that might occur between stocks due to momentum shifts in the economy. However, recent literature still shows that GNNs excel at extraction information in pair-wise interactions and struggles with capturing higher-order, multi-node interactions (Sinha et al. (2025)). Thus, techniques that use fully supervised price prediction or reinforcement learning are still preferred. Thus, we propose that combining a GNN feature extractor to calculate graph embeddings at each snapshot with a modern policy-gradient method could result in an agent that can reason over both individual factors and network-level signals which results in better trading decisions.

In this study, we bring such an agent to fruition. We construct a daily, edge-weighted graph for a small universe of eight highly liquid US stocks and ETFs. Edge weights are calculations using rolling correlations, and nodes contain both basic and more complex hand-engineered features. The daily snapshots are processed by Graph Attention Network that outputs a compact, fixed-length embedding of the entire market state on that day. We then feed this into a Proximal Policy Optimization actor-critic that outputs continuous portfolio weights under the constraint that long-only actions are permitted. The environment executes trades with realistic transaction fees and rewards for the agent are a clipped daily log-return which allows for end-to-end training.

2 Related Work

Previous works have attempted to create a RL agent capable of beating the stock market, but none have attempted to combine the power of GNNs with RL in such models despite it being a promising avenue of research (Almasan et al. (2022)). Early portfolio-RL studies often treated each asset as its own entity such as the work on FinRL (Liu et al. (2022)) which provides a suite that can compute tabular features for each stock and then couple them with agents such as A2C, DDPG, and SAC to predict investment strategies. Recently works by Kabbani and Duman (2022) showed that leveraging self-attention to pool signals across SP-500 constituents can also increase Sharpe value gains. However, such works only have information flowing implicitly through softmax pooling and fail to capture inter-asset relationships.

Almasan et al. (2022) proposed an RL agent using a static dependency graph, optimized using evolution strategies, but his methods failed to take into account changing relationships between tickers as well as the fact that his implementation was limited to cryptocurrency data. Another such work by Sarlakifar et al. (2025) used LSTMs to understand and capture long-term dependencies in stocks and feed this information into a PPO agent to handle time series data in a dynamic market, but similarly fails to encode relationships between tickers. Our work on this project differs significantly from previous works in numerous ways. First, we build daily edge-weighted graphs using the rolling Pearson correlation which allows for adaption to new regimes and changes between the correlation of two stocks. We also have a GAT encoder used to extract features from each graph snapshot which can then me used by the PPO actor-critic. Finally, we benchmark against multiple strategies to compare performance including market-trading, buy-and-hold and moving-average crossover. This is the first such project to demonstrate that a dynamic GNN-PPO system is able to outperform the above baselines in a multi-year validation set.

3 Method

The methods of this work can be described in four major stages: data preparation, graph construction, encoding of snapshots with Graph Attention Networks (GAT), and reinforcement-learning. We begin by collecting daily open-high-low-close-volume (OHLCV) data for eight US tickers from 01-01-2015 through 05-18-2025. For every trading day t and ticker S_i, we compute an eight-dimensional node-feature vector $x_i(t)$. The 8 features included consist of the one-day return:

$$r_i^{(1)}(t) = ln(\frac{P_i(t)}{P_i(t-1)})$$

where $P_i(t)$ denotes the closing price. The second is the five-day log-return:

$$r_i^{(5)}(t) = ln(\frac{P_i(t)}{P_i(t-5)})$$

, Both of these features are used to capture short-term momentum in a particular ticker's price with the one-day log-return capturing overnight and daily changes and the five-day log-return capturing weekly trends. The third features is the raw log-price:

$$ln(P_i(t))$$

which stabilizes variance between price levels and lets the network have a sense of the absolute price levels of each stock. The fourth and fifth features quantify trend strength via a ratio of the simple-moving-averages (SMA) by calculating SMA10 and SMA50:

$$SMA10_i(t) = \frac{P_i(t)}{\frac{1}{10}\sum_{k=0}^{9} P_i(t-k)}$$

$$SMA50_i(t) = \frac{P_i(t)}{\frac{1}{50}\sum_{k=0}^{49} P_i(t-k)}$$

The sixth feature is the fourteen-period relative-strength index:

$$RSI14_i(t) = 100 - \frac{100}{1 + \frac{U_i(t)}{D_i(t}}$$

where ($U_i(t)$ is the mean of the positive price changes and $D_i(t)$ is the mean of absolute negative changes over the past fourteen sessions. It gives information on whether stocks are considered over-bought or over-sold. The seventh feature is volume which is incorporated using a twenty-day z-score,

$$z - vol_i(t) = \frac{V_i(t) - \mu_{20}(V_i)}{\sigma_{20}(V_i)}$$

where $V_i(t)$ refers to the share volume and μ_{20} and σ_{20} denote the twenty day mean and standard deviation. Spikes in this metric have been shown to be good indicators of breakouts or reversals. The eighth and final feature is log-capitalization:

$$ln(P_i(t) \times N_i^{out})$$

where N_i^{out} is the free-float share count.

To encode cross-asset relations, we constructed a graph with daily edge weights. Thus the graph $G_t = (V, E_t, W_t)$ where V is the nodes and takes $V = \{S_1, \ ... \ , S_8\}$ and E_t are the edges at a specific time, t, and W_t is the weight of the edges. For each ordered pair $(i.j)$, we compute the empirical Pearson correlation (Ahmed and Kumar (2018)) of the preceding sixty log-returns:

$$\rho_{ij}(t) = \frac{\sum_{k=1}^{60} \left(r_i^{(1)}(t-k) - \bar{r}_i\right)\left(r_j^{(1)}(t-k) - \bar{r}_j\right)}{\sqrt{\sum_{k=1}^{60} \left(r_i^{(1)}(t-k) - \bar{r}_i\right)^2} \sqrt{\sum_{k=1}^{60} \left(r_j^{(1)}(t-k) - \bar{r}_j\right)^2}}$$

where $\bar{r}_i$ is the sixty day mean and $r^{(1)}$ is the one-day return as defined earlier. Pearson correlation was chosen because it is symmetric and the direction of nodes does not change the correlation, it is bounded in the range [-1, 1], and is easy to update incrementally. Other alternatives including using correlations that capture non-linear dependence were considered, but have not been implemented in this version of the model.

To keep a relative level of sparsity within the graph while informative, only the top $K = 4$ edges for each node with the largest value of $|\rho_{ij}|$ are selected as the weights to form W_t. For each of these 4, an undirected edge with weight ρ_{ij} is stored.

Each snapshot G_t is fed to a two-layer Graph Attention Network. Let $H^{(0)} = X_t \in \mathbb{R}^{8\times 8}$. The first GATv2 layer computes:

$$H^{(1)} = \sigma(\alpha^{(1)} \cdot softmax_j(f(H_i^{(0)}, H_j^{(0)}, \rho_{ij}))W^{(1)}H_j^{(0)})$$

where the attention coefficient, f, depends on the node embeddings and edge weight. A second GATv2 layer produces $H^{(2)} \in \mathbb{R}^{8\times 64}$. Flattening this yields a 512-dimension vector s_t that effectively summarizes the attributes in each node and the snapshot of the correlation graph.

The reinforcement-learning agent uses Proximal Policy Optimization. PPO maximises the clipped surrogate objective

$$L_{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right]$$

where A_t is the advantage estimate and the importance ratio $r_t(\theta)$ can be calculated as follows:

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

PPO was selected for this project because it allows for continuous actions, the clipped update implementation prevents explosive gradients, which are common in projects such as these, and the

relative ease with which we can implement PPO. Policies such as SAC were considered, but the replay buffer breaks would break the temporal ordering that is important within stock market prices and trading. This could introduce hindsight bias during periods of market shifts due to external events such as the 2020 COVID pandemic.

The actor network takes the s_t vector from the GAT and outputs Gaussian means $\mu_t \in \mathbb{R}^8$. These then get passed through a sigmoid to standardize between [0,1] and then normalized to a weight vector, w_t such that $\sum_i w_{i,t} = 1$. With the known current portfolio value, V_t and the prices P_t, the target dollar value of each position is calculated as $w_t V_t$ and it follows that the target holdings of each share are then $q_t = (w_t P_t)/P_t$. The trades are executed such that $\Delta q_t = q_t - q_{t-1}$ shares are either bought or sold and a transaction fee of $\gamma||\Delta q_t \cdot P_t||$ is deducted. This gives us a new net-worth of of V_{t+1} and then the clipped log-return is calculated as $r_t = clip(ln(V_{t+1}/V_t), -.05, .05)$.

4 Experimental Setup

The experiments were performed on daily OHLCV data downloaded from Yahoo Finance for eight liquid US equities: AAPL, MSFT, NVDA, AMZN, META, TSLA, JPM, and XOM. The raw samples are collected from 01-01-2014 to 05-18-2025, but the data from 2014-2015 is used only to calculate rolling windows for the first couple of trading days in 2015. Thus, the dataset from 01-01-2015 to 12-31-2022 forms the training set of about 2,000 training days. The remaining training days (around 600) from 01-01-2023 to 05-18-2025 are used as a held-out evaluation set. Every trading day is converted to a graph snapshot containing the eight engineered features as listed in the Methods section. During the training, the agent sees snapshots in sequence to ensure that it never sees future data, which allows for a more realistic agent.

The environment starts each episode with $1,000,000 in cash, no positions, a proportional fee on trade value (.01%), and a limitation that only long trades can be executed. Rewards are clipped within ±5% to prevent exploding or vanishing gradients. The GAT was implemented using PyTorch-Geometric 2.5 and the PPO policy was trained using Stable-Baseline3's PPO implementation where n_steps=1024, batch_size=2048, learning rate = 1×10^{-4}, and the entropy coefficient = 5×10^{-3}. Training and evaluation were both run on an AWS g4dn.xlarge instance (NVIDIA T4 GPU) and required around 4 hours of training time.

The quality of the agent is assessed on the hold-out window using 3 main metrics:

- **Total Return:** Calculated as $\frac{ending\ net\ worth}{initial\ capital} - 1$
- **Annualized Sharpe ratio:** Calculated as $\frac{\sqrt{252}\mu}{\sigma}$ using the μ and σ from daily log return
- **Maximum Draw down:** Calculated as the worst observed peak to trough decline in portfolio equity.

We compare the GNN-PPO agent against three baselines: a policy that follows the follows market indicators such as SPY or VOO, an equal-weight buy-and-hold strategy of the 8 tickers, and a simple stock trading algorithm that uses the python backtrader library to trade based on momentum/the moving-average crossover. The moving-average crossover trading strategy is one that is more conservative in its actions. Trades are executed when price crosses the moving average as these indicate reversals in trends and therefore a higher likelihood for the price of these stocks to continue in that direction. These experiments allow us to confirm whether our agent beats the market, beats a buy-and-hold of the same 8 tickers, and beats a model that trades breakouts in these tickers.

5 Results

Training was conducted for 1,000,000 environment steps over the 2015-2022 window. The tensorboard plots showed that the policy's entropy decreased from -11.35 to -12.04 and the approximate=KL divergence always remained under the 0.01 clip threshold. We had seen NaN explosions and parameter blow-ups in earlier training iterations, but the implementation of the clipped log-return rewards and adding a cap on the gradient-norm seemed to correct these issues.

On the held-out period of 595 trading data (01-03-2023 to 05-23-2025), the graph aware PPO was able to see an increase in net worth from $1,000,000 to $3,246,300 when trained on the full training

set of 2015-2022 and an increase to $3,507,400 when trained on a training set from 2019-2022. For comparison, an equal-weight buy-and-hold of the same eight tickers raw a return of $2,978,800. During the same period, the market saw portfolio growth to $1,520,000, and the implemented moving-average crossover strategy ended at $1,632,200.

Risk-adjusted metrics also show a similar result. The Sharpe ratios were 1.91 for our GNN-PPO model, 1.84 for the Equal Weight Buy and Hold, 1.02 for the moving-average crossover strategy and 0.92 for the market strategy. Furthermore, looking at the maximum drawdown, we see that they were -28.46% for our GNN-PPO agent, -28.91% for the Equal Weight Buy and Hold strategy, -22% for the market, and -15.56% for the moving-average crossover strategy. These results can be seen in Table 2.

Furthermore, visual inspection of the relative portfolio weights show how the percentage of each ticker held changes over time. This is represented in Table 1. We see that the agent started with approximately equal shares of each of the 8 tickers, ranging from 8.7% to 15% and ends with a much larger spread between the tickers going as low as 5% and as high as 35%. This shows that the agent initially bought almost equal shares of each of the eight tickers to allow for a diversified portfolio. As time progressed, the agent made trades that successfully increased the percentage of well-performing stocks and sold off those that were underperforming. We see that the relative weight of stocks such as META and NVDA in the portfolio were increased over the duration whereas stocks such as AAPL, JPM, MSFT, and XOM were sold off over the 1.5 year duration. Some stocks such as AMZN and TSLA fluctuated with the agent buying more shares of these during periods of growth and selling off during other periods which can be seen by the fact that the max percentage of these stocks in the portfolio was greater than the starting percentage.

The GNN-PPO agent consistently beat all the baselines analyzed, however, the margin of improvement varied considerably between baseline, with the training window, the feature set, and with how the edge-weights were designed and calculated. We see that limiting the training to 2019-2022 actually increased both the Sharpe and the total return. This is likely due to a combination of three factors. First, the 2019-2022 window has a market regime that looks a lot similar to the 2023-2025 one than the 2015-2022 training window and the addition of more years likely adds heterogeneity. Since we were unable to do more than 100K PPO updates, the model is likely to notice and recognize patterns within the 2015-2022 period that may never reappear again. This can result in a model that is underfitting the eight-year time span. If the window is limited to 2019-2022, the amount of noise that could have occurred is decreased and the model is better able to fit the time span within 100K updates. Another reason why is because the 2019-2022 sample is rich with volatile samples. These include cycles such as the COVID crash and spikes in meme stocks to name a few. This variety helps PPO discover risk-control strategies. While these data points are present in 2015-2022, the relative presence of these points are far fewer when compared to 2019-2022 because of the extra time. Furthermore, the volatility that arose during that time is likely more predictive of the volatility that occurred in 2023-2025 due to the presidential election.

Another result we see is one of the limitations that arise because of the Pearson edges. Rapid correlation changes such as in Jan 2024 when AAPL's price was on a downtrend while the prices of other tickers were on an an uptrend might not be caught quickly because of the lag in the 60-day window. Thus, a method that includes such correlations like the a 20-day window might better be able to capture such relationships and quick changes between two tickers. However, one of the benefits we see of this graph setup is that large edge weights can effectively share momentum signals across nodes. Thus, sector-level draw downs can be detected earlier and allow for diversification before it happens.

5.1 Quantitative Evaluation

Across the evaluation metrics, we see that the graph-aware PPO model outperforms the baselines (Table 2). The total return is over 30 percentage points higher than the next closest- the equal weight buy and hold. The GNN-PPO return is also 3.5 times that of the moving-average crossover strategy indicating that the GNN-PPO makes riskier trades that also pay off in the long run. An analysis of the Sharpe Ratios presents very similar results in that GNN-PPO has the highest Sharpe Ratio which indicates that the returns the agent saw can be attributed to smarter trading strategies rather than luck. Equal-weight buy and hold has similar a similar Sharpe Ratio indicating that it is also a relatively robust strategy, but it does not see as high of returns as the GNN-PPO model saw. The GNN-PPO does outperform the market and the moving-average crossover strategy by close to an entire point,

Table 1: Portfolio Weights for Each Ticker

Ticker	% Portfolio Weight in 2023	% Portfolio Weight in 2025	max % of portfolio over 2023-2025
AAPL	13%	6.5%	13%
AMZN	10%	7.5%	11%
JPM	13%	7.5%	13%
META	15%	23%	27%
MSFT	8.7%	5%	8.7%
NVDA	15%	35%	41%
TSLA	12%	10.5%	17%
XOM	15%	5%	15%

Table 2: Trading Strategy Evaluation

Method	Total Return	Sharpe Ratio	Max Drawdown
Market (SPY)	+49%	0.86	-22%
Market (VOO)	+52%	0.92	-19%
Moving-Average Crossover	+63%	1.02	-15.56%
Equal Weight Buy and Hold	+197.88%	1.84	-28.91%
GNN-PPO(train: 2015-2022)	+224.63%	1.86	-28.62%
GNN-PPO(train: 2019-2022)	+249.74%	1.91	-28.46%

which clearly indicates that our agent follows a better strategy than trading the market or only when there is a certainty that the price will follow a trend. Thus, it follows that our agent's strategy is more volatile than the moving-average crossover. The moving-average crossover strategy only sees a max drawdown of -15.56% while the GNN-PPO model sees a max drawdown of -28.62%. This shows that the former is a much safer trading strategy than the GNN-PPO. Similarily, the market-trading strategy experiences a max drawdown of around -20%, which is also better than the GNN-PPO strategy. However, we do see that the GNN-PPO's max drawdown is slightly better than the equal weight buy and hold indicating slightly less volatility in the strategy.

An experiment was also done to compare the effects of training the model on historical data from 2015-2022 vs data from 2019-2022. The results in table 2 show that the return for the model trained on 2019-2022 data had an extra 25% performance boost on the total return compared to the model trained from 2015-2022. The model trained on data from 2019 onwards also presents with a higher Sharpe ratio and lower max drawdown. This can likely be attributed to the fact that the 2019-2022 period likely resembled the 2023-2025 period more closely, the 2015-2022 time range likely added more heterogeneity, and the 2019-2022 contains a relatively higher percentage of volatile data points.

5.2 Qualitative Analysis

The plots show that the agent behaves in a pattern which mimics that of a well-trained RL stock trading agent. If we are to look at the net worth of the agent's portfolio, we see that it is able to grow throughout the 2023-2025 duration. Overall, it follows growths within the market with periods of growth and dips within the market. One notable period is that from late February 2025 to around April 2025. This period is notable for the tariffs and trade war which is why we see a considerable in the agent's net worth. This is the largest maximum drawdown, but if we look at the eight tickers that comprised our universe, we can see that most had a much larger price drop than our agent's portfolio did. 1

We can also analyze qualitatively, the effects the agent's actions had on each individual ticker.

- **AAPL:** As seen in Fig 2, the price of the stock increased from about $125 to $212 from 2023 to 2025. This corresponds to about a 1.7 increase in price. Most of the 8 tickers in our environment had larger price increases. Thus, we see that the relative weight of AAPL throughout the duration of the 600 trading days decreased within our portfolio. AAPL constituted around 13% of the portfolio at the beginning and only 6.5% near the end meaning that the importance of the stock in bringing in returns was halved

Figure 1: Net worth of stock portfolio from 2023-present

- **AMZN:** In Fig 3, we see the price of AMZN increased from about $85 to $210. This corresponds to around a 2.5 increase in price. We see that the shares of AMZN actually increased in the first couple of weeks to 11% from 10%, but as time passed, the relative weight of AMZN within the portfolio diminished to 7.5% as other tickers were given priority.
- **JPM:** Fig 4 shows that JPM started at a price around $130 in 2023 and grew to $270 in 2025. This is around a two-fold increase in the price and other tickers still outperformed the gains in this stock. Thus, the weight of JPM holdings decreased steadily throughout the duration starting at around 13% and plateauing at 7.5%.
- **META:** As seen in Fig 5, META experienced considerable price growth from 2023 to 2025 growing from about $130 to around $650. This corresponds to a 5x growth in the price of META. This stock is one of the biggest movers within our universe and thus, the agent was able to understand this relationship and purchase more shares during the 1.5 year duration. The agent initially had around 15% of the portfolio as META shares which grew to as high as 27% and ended at around 23% of the portfolio being META.
- **MSFT:** We see in Fig 6 that MSFT had a starting price in 2023 of $275 and ended at around $460 which is a growth of 1.67. Compared to some of the other tickers, this stock experience comparatively less growth and thus the agent initially weighted 8.7% of the portfolio as MSFT and ended around 5% with the portfolio weight declining steadily throughout the duration
- **NVDA:** Fig 7 shows us that the price of NVDA in 2023 was around $15 and grew to a staggering $135 in 2025. This corresponds to a 9x increase in price and the agent acted accordingly. Around 15% of the portfolio were shares of NVDA at the beginning and it increased to a staggering 41% of shares being NVDA at its highest. At the end of the evaluation period, around 35% of the shares in the portfolio were invested in NVDA. This is in according to what we expect as the stock improved its value nine-fold.
- **TSLA:** TSLA experienced a multitude of price changes over the years of 2023-2025 as seen in Fig 8. The price started at around $110 and grew to $350 in 2025. This is around a 3.2x in price, but the plot also shows major variations in the price throughout the duration. Thus, the agent initially weighted around 12% of the portfolio to be TSLA, but as prices fluctuated, the percentage of TSLA in the portfolio increased to as high as 17% and then went down to 10.5% in 2025 likely due to all the news surrounding Musk's involvement in DOGE.
- **XOM:** XOM is one of the stocks that saw almost no growth in the period of 2023-2025 which can be seen in Fig 9. The price of XOM in 2023 was around $98 and this increased to $108 in 2025, which is about 1.1 times. The agent clearly believed that XOM had growth potential in 2023 as 15% of the portfolio was XOM (one of the highest bought stocks), but quickly learned to sell off the stock until it remained as only 5% of the portfolio. This behavior is to be expected if the price is not changing as rapidly as the others in the environment.

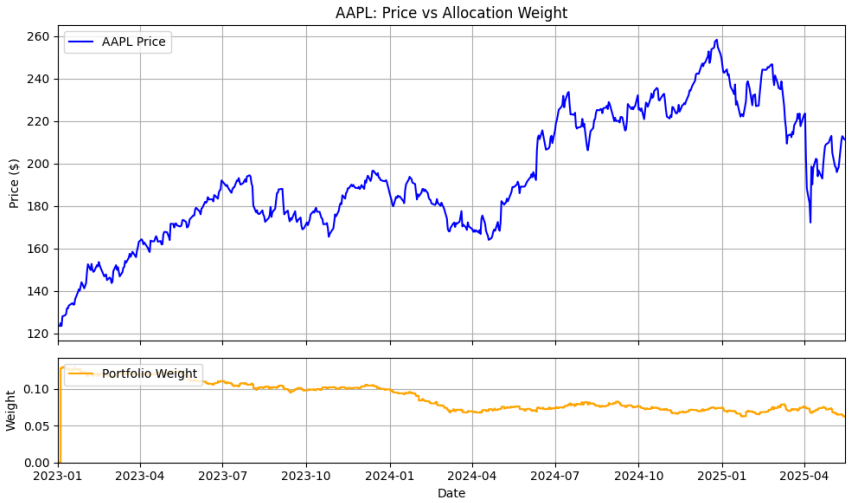

Figure 2: Price of AAPL from 2023-2025 and how the relative weight of AAPL in portfolio changes

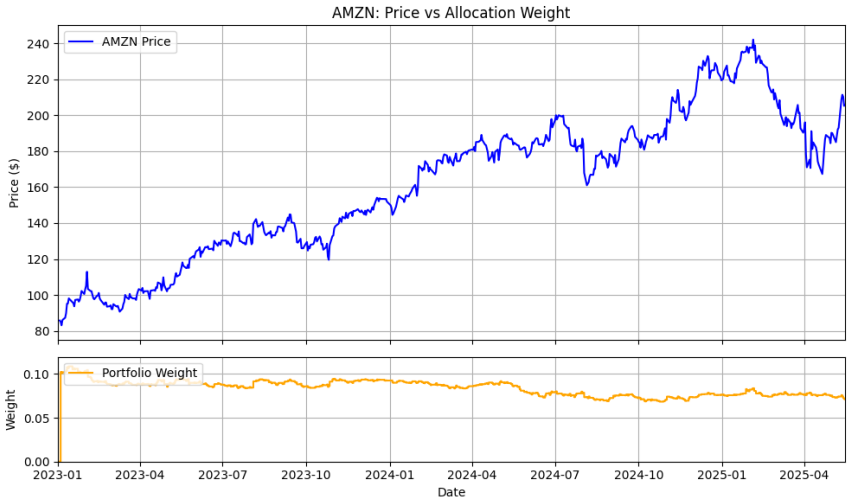

Figure 3: Price of AMZN from 2023-2025 and how the relative weight of AMZN in portfolio changes

6 Discussion

Despite the encouraging results, our model has some clear limitations- the biggest one being that we limit our universe to eight mega-cap tickers within similar sectors. This was selected to allow for deep enough liquidity to execute all trades and query from Yahoo Finance quickly, but it, therefore, simplifies some of the cross-sectional complexity and risks associated with trading in the real world. Extending to the full SP 500, or at least a more mixed subset of stocks + ETFs, would be able to test the model's ability to scale when there are more than 8 nodes from a multitude of sectors. Second, picking k=4 and using Pearson Correlation is only one of the possible edge-selection rules that we could have chosen. Implementing non-linear correlations or expanding to greater than k=4 could have given us different results and is a hyperparameter that could be tuned in the future. Third, the reward function is a simple clipped daily log-return. This successfully stabilizes PPO and allows the model to run, but it does not penalize any turnover, volatility or risk. Incorporating other factors into the reward function including Sharpe or Conditional Value at Risk (CVaR) adjusted reward could result in better performance that minimizes the max draw down. Finally, although we benchmark against a market strategy, buy-and-hold, and a moving-average crossover strategy, a stronger baseline

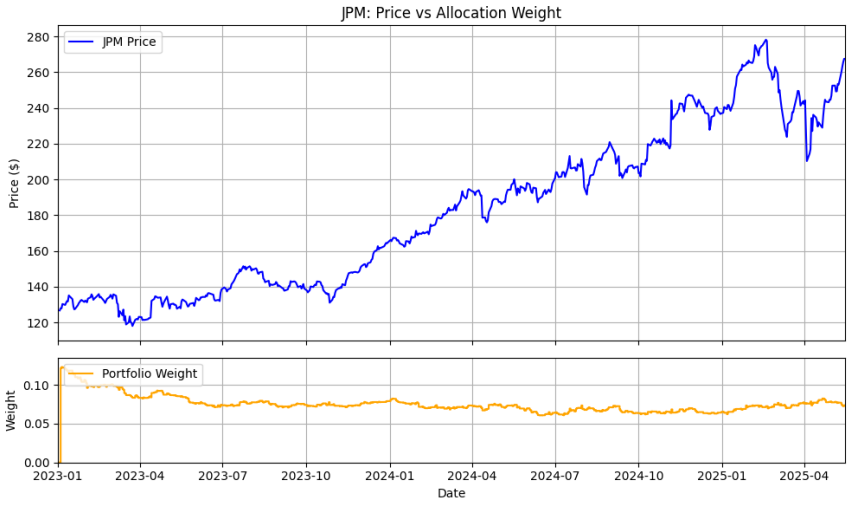

Figure 4: Price of JPM from 2023-2025 and how the relative weight of JPM in portfolio changes

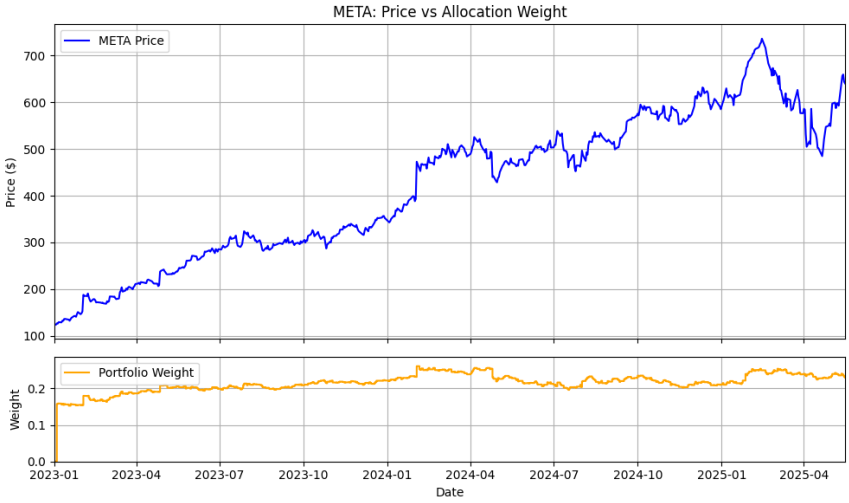

Figure 5: Price of META from 2023-2025 and how the relative weight of META in portfolio changes

such as another RL model would make the performance of the model more persuasive. These are all to be included in the future direction this project can take.

The impact of this model on the broader investment world is huge. A GNN-PPO agent that reallocates capital dynamically improves portfolio diversification while remaining relatively inexpensive to implement. However, the converse is true in that if many such models like this become popular on the market, it could increase the fragility of the market- simultaneous action by these models would result in sudden spikes that can cause the models to de-risk. In addition, it shows that using and entirely data-driven model can work at market prediction. It ignores aspects such as market news or global actions, but it is still able to encode with these historical biases in the model.

This project also did face difficulties. While reading the data from Yahoo's OHCLV was relatively straightforward, engineering of the features resulted in NaNs and values such as infinity that made our first couple of runs of the model not useful. Exhaustive cleaning of the data and clamping were eventually required to allow the agent to train without issues. Another hurdle was dealing with the instability inside the RL loops. Initially, the ability to short-sell was allowed for the model, but this resulted in the agent buying millions of shares and trading outside the contained cash limits. Thus,

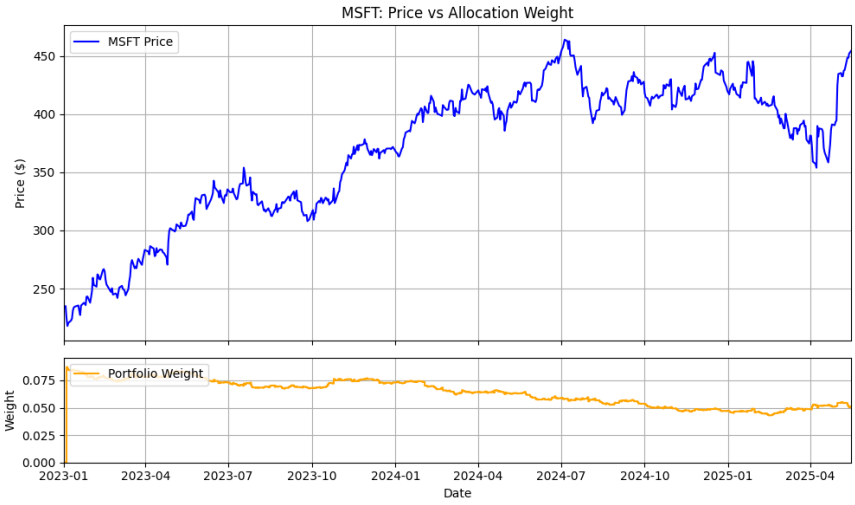

Figure 6: Price of MSFT from 2023-2025 and how the relative weight of MSFT in portfolio changes

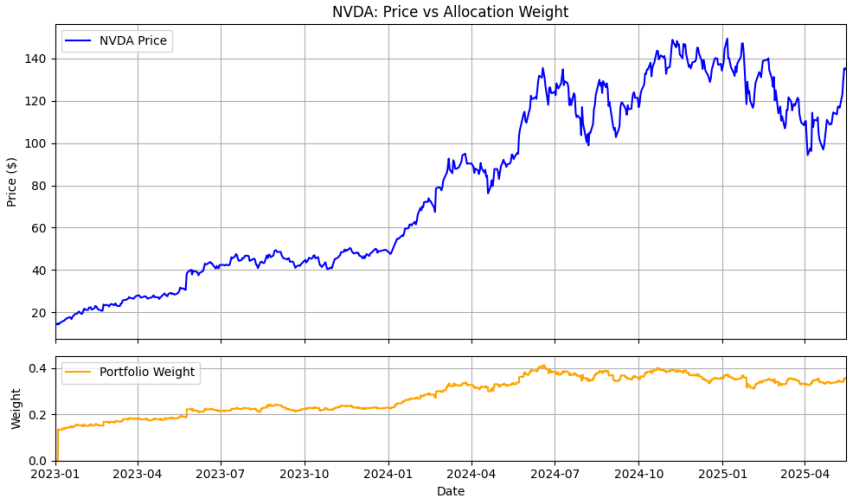

Figure 7: Price of NVDA from 2023-2025 and how the relative weight of NVDA in portfolio changes

actions were limited to long-only and a single T4 GPU was used which does have limited compute power and hindered our ability to focus on hyper-parameter tuning.

7 Conclusion

Overall, this work shows that by coupling a dynamic, correlation-based graph representation of the equities with a graph-aware PPO allows for substantially higher returns than classical baselines such as trading the market, buy-and-hold, and moving-average crossover. By reasoning over and understanding how tickers co-move, the agent is able to reallocate funds towards/ against those shifting patterns. This allows us to see over triple the capital in a held out validation set of 2023-2025 while reducing drawdown and improving the Sharpe ratio. This signifies that modeling the relationships between tickers as graphs better allows a PPO RL model to exploit those structural relationships and understand market movements. Future work will aim to generalize to a larger set of tickers, incorporate adaptive or learned edge definitions, and improve the reward function so that our current graph-aware RL can evolve from an academic prototype to an industry-grade trading agent.

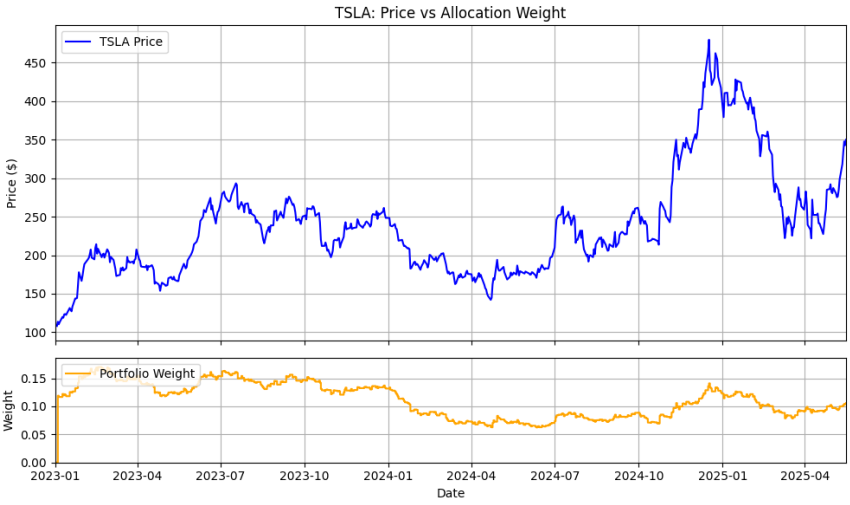

Figure 8: Price of TSLA from 2023-2025 and how the relative weight of TSLA in portfolio changes

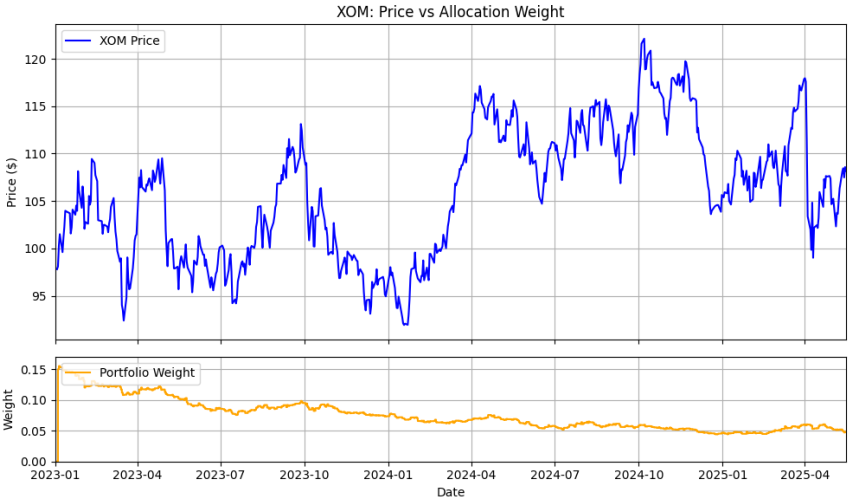

Figure 9: Price of XOM from 2023-2025 and how the relative weight of XOM in portfolio changes

8 Team Contributions

- **Nevin Aresh:** Worked on this project alone and thus did the work for this project

Changes from Proposal This entire project is a major change from the proposal as we expected to do a project on biomedical reasoning LLM's but this project could not be completed within the duration of the class. Thus, I pivoted projects halfway through the quarter to work on this project and modeling the stock market as graphs and training a PPO agent on these snapshots.

References

Zafar Ahmed and Sachin Kumar. 2018. Pearson's correlation coefficient in the theory of networks: A comment. arXiv:1803.06937 [cond-mat.dis-nn] `https://arxiv.org/abs/1803.06937`

Paul Almasan, José Suárez-Varela, Krzysztof Rusek, Pere Barlet-Ros, and Albert Cabellos-Aparicio. 2022. Deep reinforcement learning meets graph neural networks: Exploring a routing optimization

use case. *Computer Communications* 196 (Dec. 2022), 184–194. https://doi.org/10.1016/j.comcom.2022.09.029

Taylan Kabbani and Ekrem Duman. 2022. Deep Reinforcement Learning Approach for Trading Automation in the Stock Market. *IEEE Access* 10 (2022), 93564–93574. https://doi.org/10.1109/access.2022.3203697

Bharti Khemani, Shruti Patil, Ketan Kotecha, and Sudeep Tanwar. 2024. A review of Graph Neural Networks: Concepts, architectures, techniques, challenges, datasets, applications, and Future Directions - Journal of Big Data. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-023-00876-4#citeas

Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang. 2022. FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance. arXiv:2011.09607 [q-fin.TR] https://arxiv.org/abs/2011.09607

Faezeh Sarlakifar, Mohammadreza Mohammadzadeh Asl, Sajjad Rezvani Khaledi, and Armin Salimi-Badr. 2025. A Deep Reinforcement Learning Approach to Automated Stock Trading, using xLSTM Networks. arXiv:2503.09655 [cs.CE] https://arxiv.org/abs/2503.09655

Akshit Sinha, Sreeram Vennam, Charu Sharma, and Ponnurangam Kumaraguru. 2025. Higher Order Structures For Graph Explanations. arXiv:2406.03253 [cs.LG] https://arxiv.org/abs/2406.03253