Capstone Project Final Report

Deep Reinforcement Learning in Automated Stock Trading

Yuzhu Feng 903946259

Leiru Long 903940713

Ziyun Wang 903880511

Xinhao Zhou 903949668

SCHELLER COLLEGE OF BUSINESS

GitHub Repo: Capstone-DRL-for-Automated-Stock-Trading

December 8, 2024

Table of Contents

1 Introduction

Financial markets are complex and dynamic systems where predicting asset prices and executing profitable trades require both strategic decision-making and adaptability to rapidly changing conditions. Automated stock trading, powered by machine learning, has emerged as a powerful tool for navigating these challenges. Among machine learning techniques, Deep Reinforcement Learning (DRL) has gained prominence due to its ability to learn optimal trading strategies directly from data without requiring explicit modeling of market dynamics.

DRL offers a unique advantage by framing the trading problem as a Markov Decision Process (MDP). This allows trading agents to make sequential decisions by maximizing cumulative rewards over time. Unlike traditional rule-based or supervised learning methods, DRL models can adapt to diverse market conditions, learning from their interactions with the environment to improve performance. These models have been applied successfully in areas like portfolio optimization, market-making, and algorithmic trading.

This project builds on the methodologies proposed in the *Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy* [6]. The baseline framework uses an ensemble of three actor-critic algorithms—Advantage Actor-Critic (A2C), Deep Deterministic Policy Gradient (DDPG), and Proximal Policy Optimization (PPO)—to develop robust trading strategies. However, empirial tests questioned the model's robustness and adaptability to the evolving market condition.

Our work extends this framework by addressing its limitations and enhancing its capabilities. Cumulative reward functions are implemented to balance short-term gains with long-term performance optimization. Additionally, we experimented alternative stop-loss mechanisms, including Return-Based Stop-Loss and Value-Based Stop-Loss, to better handle extreme market conditions. Furthermore, we introduce two additional algorithms, Twin Delayed Deep Deterministic Policy Gradient (TD3) and Soft Actor-Critic (SAC), to improve algorithmic diversity and robustness. These algorithms bring advanced features like reduced overestimation bias and better exploration-exploitation balance, making them well-suited for volatile markets. As we added two more agents and used cumulative reward function, which require extensive computational costs, we proposed to use transfer learning to leverage insights from past training to enhance training efficiency.

By integrating these advancements, this project aims to develop a more resilient and adaptive automated trading system. The outcomes provide valuable insights into the application of DRL in financial markets and its potential to revolutionize stock trading practices.

2 Literature Review

Deep Reinforcement Learning (DRL) has emerged as a transformative tool for designing trading strategies in complex and dynamic financial markets. Unlike traditional approaches that rely on predefined rules or statistical models, DRL frameworks adapt dynamically to changing market conditions through data-driven policies. By modeling trading as a Markov Decision Process (MDP), DRL enables the optimization of sequential decision-making tasks, including portfolio management, high-frequency trading, and market making.

In automated stock trading, Yang et al. (2020) [6] proposed an ensemble strategy combining three DRL algorithms—PPO, A2C, and DDPG—to enhance robustness and profitability. Their approach achieved superior Sharpe ratios compared to traditional methods like minimum-variance portfolios. The integration of a turbulence index added a risk-aversion component, enabling the model to respond effectively to extreme market conditions, such as financial crises. Bouzgarne et al. (2023) [3] further emphasized the importance of balancing exploration and exploitation in DRL to achieve consistent performance across diverse portfolios.

In the context of market making, Sun et al. (2022) and Gašperov and Kostanjčar (2021) [1] utilized DRL to optimize strategies for quoting bid and ask prices. Their work demonstrated the potential of DRL agents to capture market spreads while minimizing associated risks, providing a robust framework for adapting to dynamic market conditions.

The design of the state space is critical for the success of DRL in trading, as it encapsulates the agent's perception of the environment. For instance, Yang et al. (2020) [6] represented the state as a vector comprising stock prices, inventory levels, and cash balances, ensuring that the agent's actions were guided by its current environment to achieve optimal portfolio performance.

Building on these foundations, Sun et al. (2022) [5] extended the state space to include internal agent states (e.g., inventory and remaining orders) and external market dynamics captured from the limit order book (LOB). By using deep recurrent neural networks such as LSTMs, their approach extracted complex temporal patterns, replacing traditional handcrafted features and adapting effectively to high-frequency trading environments.

Similarly, Gašperov and Kostanjčar (2021) [1] incorporated predictive signals, such as price range and trend forecasts, derived from supervised learning models into the state space. This inclusion provided the agent with forward-looking insights, enhancing its ability to navigate dynamic market conditions. Furthermore, integrating financial metrics like turbulence indices improved risk management during periods of extreme volatility. These advancements collectively enhanced the robustness and adaptability of DRL agents in complex trading environments.

The reviewed studies underscore the suitability of DRL models for addressing the complexities of trading tasks in financial markets. By leveraging dynamic state representations, integrating predictive insights, and balancing risk and reward, DRL frameworks demonstrate the ability to adapt to volatile market conditions and optimize decision-making processes. The versatility of DRL in modeling sequential actions, combined with its capacity to handle high-dimensional data, makes it a promising approach for developing robust, efficient, and profitable trading strategies in a highly dynamic and uncertain environment.

3 Baseline Model Methodology

For our project, we build upon the methodology proposed by Yang et al. (2020) [6], which serves as the baseline framework for applying deep reinforcement learning (DRL) models to automated trading. Their study provides a foundational approach to modeling stock trading as a Markov Decision Process (MDP) and utilizing actor-critic-based algorithms to optimize trading strategies. This baseline methodology is extended and adapted to address specific challenges and enhance performance in dynamic financial markets.

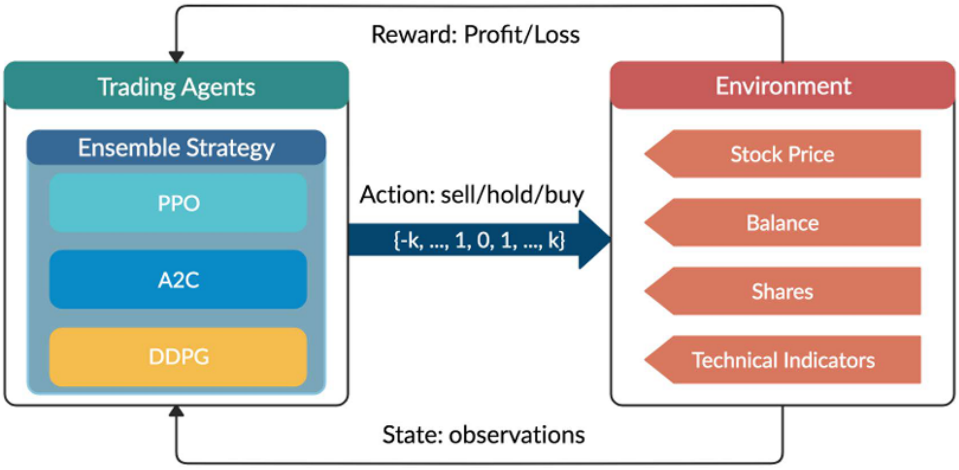


Figure 3-1 Overview of reinforcement learning-based stock trading strategy [6]

3.1 Reinforcement Learning Framework

Markov Decision Process (MDP)

Stock trading is modeled as a **Markov Decision Process (MDP)**, characterized by a tuple (S, A, P, R, γ):

- **State Space** (S): The state includes features that capture market conditions and portfolio status, such as stock prices ($\mathbf{p}_t$), the number of shares held ($\mathbf{h}_t$), available balance (b_t), and technical indicators like Moving Average Convergence Divergence ($MACD_t$), Relative Strength Index (RSI_t), Commodity Channel Index (CCI_t), and Average Directional Index (ADX_t). At time t, the state is represented as a vector:

$$\mathbf{s}_t = [b_t, \mathbf{p}_t, \mathbf{h}_t, MACD_t, RSI_t, CCI_t, ADX_t] \tag{3.1}$$

- **Action Space** (A): The action space defines continuous decisions to buy, sell, or hold stocks. Actions are defined as $a_t \in \{-k, \ldots, 0, \ldots, k\}^D$, where D is the number of stocks, and k is the maximum transaction size. The actions are further normalized to $[-1, 1]$, since the RL algorithms A2C and PPO define the policy directly on a Gaussian distribution, which needs to be normalized and symmetric [2].
- **Reward Function** (R): The reward is the change in portfolio value resulting from action a_t in state s_t, accounting for transaction costs. In Equation 3.3, c_t represents transaction costs for buying ($\mathbf{k}_B$) and selling ($\mathbf{k}_S$) shares.

$$r(s_t, a_t, s_{t+1}) = \left(b_{t+1} + \mathbf{p}_{t+1}^\top \mathbf{h}_{t+1}\right) - \left(b_t + \mathbf{p}_t^\top \mathbf{h}_t\right) - c_t \tag{3.2}$$

$$c_t = 0.001 \cdot (\mathbf{p}_t^\top \mathbf{k}_B + \mathbf{p}_t^\top \mathbf{k}_S) \tag{3.3}$$

Turbulence Index

To manage extreme market conditions, a **turbulence index** is incorporated into the reward function. When turbulence exceeds a predefined threshold, the agent halts buying actions and liquidates all positions to minimize losses. The turbulence index d_t is calculated in Equation 3.4.

$$d_t = (\mathbf{y}_t - \mu)^\top \Sigma^{-1} (\mathbf{y}_t - \mu) \tag{3.4}$$

where

$d_t =$ Turbulence for a particular time period t (scalar),
$\mathbf{y}_t =$ Vector of asset returns for period t ($1 \times n$ vector),
$\mu =$ Sample average vector of historical returns ($1 \times n$ vector),
$\Sigma =$ Sample covariance matrix of historical returns ($n \times n$ matrix).

3.2 Algorithms for Policy Optimization

The methodology employs three actor-critic-based reinforcement learning algorithms, each chosen for their strengths in handling continuous action spaces, stability, and adaptability:

1. **Advantage Actor-Critic (A2C):** A2C utilizes an advantage function to reduce the variance of policy gradients, improving stability. The advantage function is defined as $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$, where $V(s_t)$ is the state value function. Policy gradients are updated as:

$$\nabla_\theta J(\theta) = \mathbb{E}\left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) A(s_t, a_t)\right] \tag{3.5}$$

2. **Deep Deterministic Policy Gradient (DDPG):** DDPG combines Q-learning with policy gradients for continuous action spaces. The Q-value is updated using:

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^\mu)) \tag{3.6}$$

and the critic network minimizes the loss:

$$L(\theta^Q) = \mathbb{E}\left[(y_i - Q(s_i, a_i|\theta^Q))^2\right] \tag{3.7}$$

3. **Proximal Policy Optimization (PPO):** PPO employs a clipped objective function to constrain policy updates and enhance training stability. $r_t(\theta)$ in Equation 3.9 is the probability ratio between new and old policies.

$$J_{\text{clip}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta) A(s_t, a_t), \text{clip}(r_t(\theta), 1 - \varepsilon, 1 + \varepsilon) A(s_t, a_t)\right)\right] \tag{3.8}$$

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \tag{3.9}$$

3.3 Ensemble Strategy and Validation

An ensemble strategy integrates the strengths of A2C, DDPG, and PPO to ensure robustness. The methodology employs a dynamic approach for training and validation:

- **Training Period:** Agents are trained on a growing window of historical data to learn optimal policies. 01/01/2009-09/30/2015 is used as the initial training period of the agents. For the next rebalancing window, an extended period of 01/01/2009-12/31/2015 will be used to train the next set of agents.
- **Validation Period:** A three-month window is used to evaluate agents based on the **Sharpe ratio**. The best-performing agent is selected for the subsequent 3-month trading period. Here, the first validation window is 10/01/2015-12/31/2015. For the next rebalancing window, 01/01/2016-3/31/2016 will be used to valid the performance of the agents.

$$\text{Sharpe ratio} = \frac{r_p - r_f}{\sigma_p} \tag{3.10}$$

- **Trading Period:** After validation in the previous window, the selected agent will be used to trade in the next rebalancing window of 3 months. Here, the first trading window is 01/01/2016-3/31/2016.

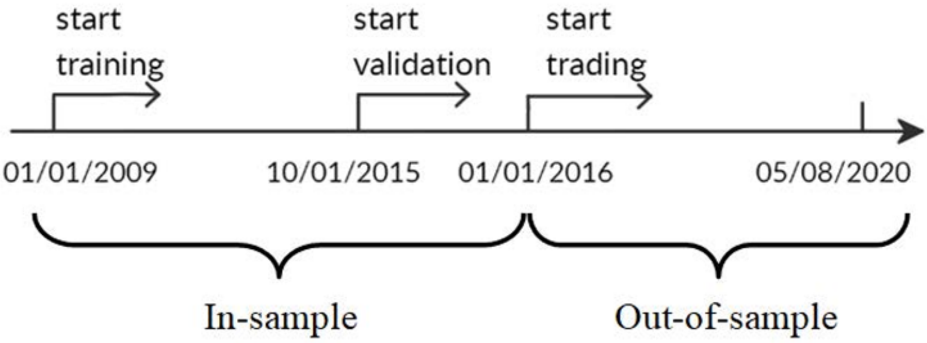


Figure 3-2 Strategy Rebalancing Mechanism [6]

3.4 Data Sources and Results

According to Yang et. al. [6], the trading universe consists of the **Dow Jones Industrial Average (DJIA) 30 stocks**, with historical daily data from **2009 to 2020** obtained via the *Compustat database*. The dataset is further splitted, with 01/01/2009-09/30/2015 as the initial training period, 10/01/2015-12/31/2015 as the initial validation period, and 01/01/2016-05/08/2020.

Key results of the baseline method include:

- **Performance Metrics:** The ensemble strategy achieves a Sharpe ratio of **1.30**, compared to **0.47** for the DJIA index and **0.45** for a minimum-variance portfolio. It also delivers a cumulative return of **70.4%** (vs. **38.6%** for the DJIA) and maintains low volatility (**9.7%** annualized).

- **Market Crash Adaptability:** During the 2020 market crash, the turbulence index effectively mitigated losses, halting risky trading actions.
- **Dynamic Adaptation:** The ensemble strategy adapts to market conditions, with A2C excelling in bearish markets and PPO in bullish conditions.

Building upon the existing methodology, we aim to further enhance the performance of the model to address the complexities of more dynamic and unpredictable market conditions, particularly in the period following 2020. This era, characterized by heightened volatility and structural market changes, was not included in the baseline study. Our approach seeks to adapt the model to these evolving challenges, ensuring its robustness and effectiveness in modern financial markets.

4 Proposed Enhancements

This section outlines a series of improvement ideas designed to enhance the baseline model's performance and adaptability to complex market conditions. These concepts will be evaluated through implementation, with detailed results and analyses presented in Section 5.

4.1 Cumulative Reward Functions

In the baseline methodology, the reward function r was defined as the Profit and Loss (PnL) of daily trades, computed as

$$r_t = V_t - V_{t-1} \tag{4.1}$$

where V_t = end-of-day account value and V_{t-1} = start-of-day account value.

To better capture the long-term performance and smooth the rewards, we propose two cumulative reward function designs:

Cumulative Reward Function 1: Weighted Combination of Current and Cumulative Returns

The first approach incorporates both the daily return (R_t) and the cumulative return (R_t^c), which are calculated as

$$R_t = \frac{V_t - V_{t-1}}{V_{t-1}} \qquad R_t^c = \left(\frac{V_t}{V_0}\right)^{\frac{1}{T}} - 1 \tag{4.2}$$

where V_0 = initial account balance and T =number of days. The cumulative reward function r' is then computed as a weighted combination of these two returns:

$$r'_t = (1-\lambda)\cdot R_t + \lambda \cdot R_t^c \tag{4.3}$$

where λ = decay rate is set to 0.2 in our experiments.

Cumulative Reward Function 2: Exponentially Weighted Historical Returns

The second approach incorporates an exponentially decaying sum of historical daily returns. The reward r'' at time t is calculated as:

$$r''_t = R_t + \lambda \cdot R_{t-1} + (\lambda)^2 \cdot R_{t-2} + \ldots = \sum_{i=0}^{t-1} \lambda^i \times R_{t-i} \tag{4.4}$$

This method emphasizes recent returns while retaining a decayed influence of earlier returns, controlled by the decay rate. As with the first approach, λ is set to 0.2 in our implementation.

These proposed reward functions aim to encourage policies that optimize both short-term and long-term performance, while maintaining stability in training.

4.2 Stop-loss Mechanisms

The baseline methodology employed a turbulence-based stop-loss mechanism, which calculated a threshold from historical data and froze trading when market turbulence exceeded this threshold. While designed to protect the model during adverse conditions, this approach proved ineffective during periods of extreme market volatility, such as the COVID-19 pandemic. The exceptionally high turbulence during this time repeatedly triggered the stop-loss, effectively disabling the Reinforcement Learning (RL) model and preventing normal trading.

This limitation underscores the need for more adaptive stop-loss mechanisms that can manage extreme market conditions while allowing the model to function effectively.

Return-Based Stop-Loss

The return-based stop-loss mechanism evaluates performance using a rolling window cumulative return over the past n days. The window cumulative return $R_t^c(n)$ is calculated as:

$$R_t^c(n) = [(1+R_t)\cdot(1+R_{t-1})\cdots(1+R_{t-n+1})]^{\frac{1}{n}} - 1 \tag{4.5}$$

where R_t is the return at time t, and the window length n is set to 3 in our implementation. The trading strategy applies the following rules:

If the window cumulative return falls below -2%, all positions are cleared, and trading is frozen. During the freeze period, trading is simulated (but not executed) to continue tracking daily returns. When the simulated $R_t^c(n)$ exceeds 1%, trading resumes.

This approach ensures that the strategy avoids trading in persistently adverse conditions while remaining sensitive to market recovery.

Value-Based Stop-Loss

The value-based stop-loss mechanism focuses on the account value. It tracks the maximum account value achieved so far, denoted as $V_{\max}$, and applies the following rules:

If the account value drops by 5% from $V_{\max}$, trading is frozen, and $V_{\max}$ is updated to the dropped value, V_{dropped}. Similar to return-based stop-loss, when the simulated $R_t^c(n)$ exceeds 1%, trading resumes with the updated $V_{\max}$.

This mechanism dynamically adjusts to new market conditions, balancing the need for risk mitigation with the opportunity to capitalize on market recoveries.

4.3 Enhanced Ensemble Learning for Model Selection

The baseline methodology employed an ensemble learning approach, training three algorithms: Advantage Actor-Critic (A2C), Deep Deterministic Policy Gradient (DDPG), and Proximal Policy Optimization (PPO). During the validation period, the best-performing model was selected to execute the actual trading.

We improve upon this approach by adding two additional algorithms: Twin Delayed Deep Deterministic Policy Gradient (TD3) and Soft Actor-Critic (SAC). These enhancements were motivated by the following considerations:

- **Twin Delayed Deep Deterministic Policy Gradient (TD3)**: TD3 improves DDPG by reducing overestimation bias through target value smoothing and delayed updates, leading to more stable training in continuous action spaces.

- **Soft Actor-Critic (SAC)**: SAC uses a maximum entropy framework to balance exploration and exploitation, making it effective in volatile, high-dimensional environments like financial trading.

By integrating TD3 and SAC into the ensemble, we expand the diversity of algorithms available for selection. This provides a more robust model selection process during validation and enhances the potential for improved trading performance across different market conditions.

4.4 Transfer Learning

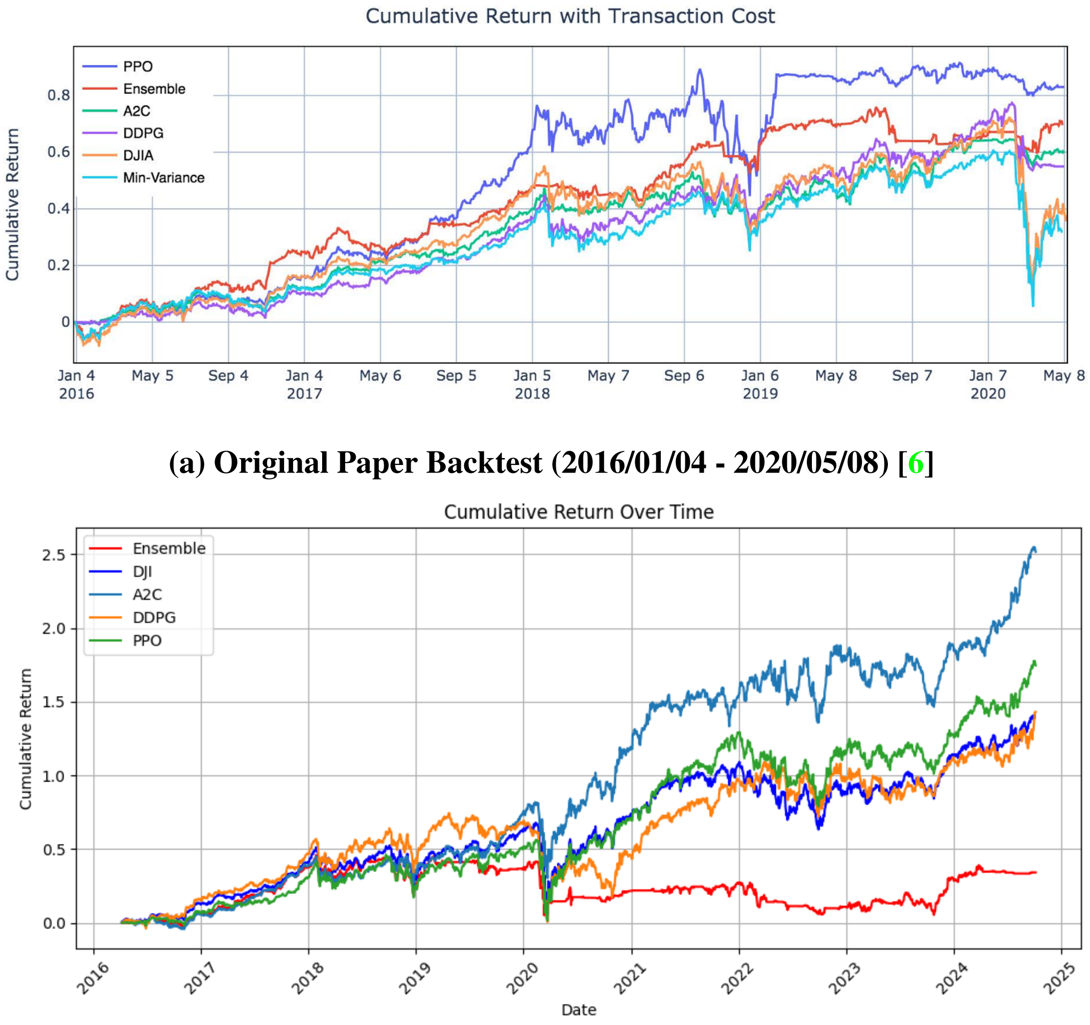
To reduce computational costs and leverage insights from past training, we propose using transfer learning. Instead of training a new model from scratch for each trading period, we initialize the model for the current period using the trained parameters from the previous period.

This approach allows the model to retain learned knowledge, adapt to new data more efficiently, and significantly reduce computational overhead as the dataset grows over time. By building on past performance, transfer learning enhances both training efficiency and continuity in trading strategy development.

5 Model Evaluation and Return Analysis

5.1 Limitations of the Baseline Strategy

The baseline strategy outlined in the study by Hongyang Yang et al. [6] demonstrated impressive performance during its empirical testing window (2016/01/04 to 2020/05/08). However, when the same model was tested on an extended period to 2024-10-31, the results were less satisfactory, indicating potential limitations in the model's robustness and adaptability to evolving market conditions.



(a) Original Paper Backtest (2016/01/04 - 2020/05/08) [6]

(b) Extended Empirical Testing (2016/01/04 - 2024/10/31)

Figure 5-3 Performance Comparison Between 2 Windows

Regarding the discrepancy, we asserted several possibles issues to consider and improve:

- **Market Turbulence Post-2020:** The market conditions post-2020 were characterized by unprecedented turbulence, driven by global crises such as the COVID-19 pandemic, geopolitical tensions, and rapid changes in economic policies, as shown in Figure 5-4. While the turbulence threshold implemented in the baseline model effectively mitigated risk during extreme events, it also significantly constrained profit opportunities. This is particularly evident in prolonged periods of elevated market volatility where the system's risk aversion limited its ability to capitalize on high-return opportunities. As shown in Figure 5-3, the profitability of the ensemble strategy is greatly constrained comparing to those individual agents without turbulence stop-loss concerns.

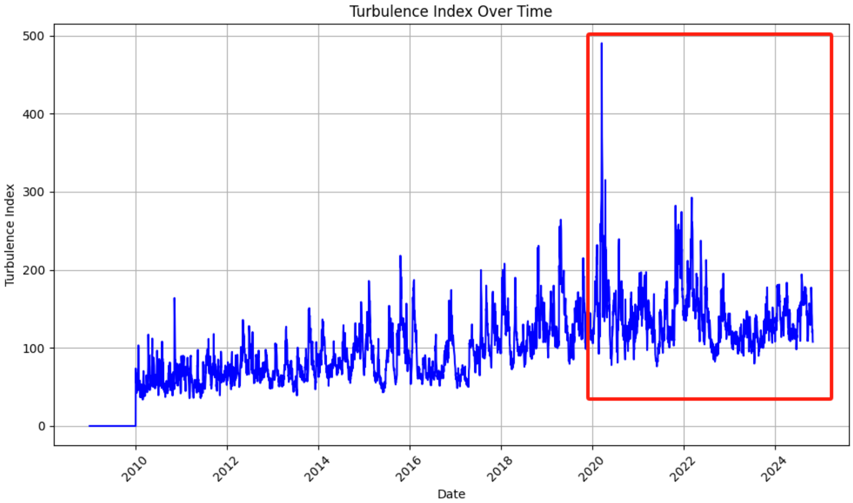


Figure 5-4 Turbulence Index from 2011 to 2024

Despite these challenges, the turbulence index remains a valuable metric for assessing market volatility, as suggested by Kritzman's [4]. Its integration into the model underscores the need for a more nuanced approach, such as adaptive thresholds or additional mechanisms to optimize performance under varying market conditions.

- **Reward Function Misalignment:** Another limitation stems from the misalignment between the reward function used in the baseline strategy and the current market dynamics. While the baseline reward function is defined as the PNL of daily trades, during periods of extreme market fluctuations, the reward function will overly focus short-term gains, leading to inconsistent and volatile outcomes. This short-sighted approach neglected the importance of long-term performance and reward smoothing, both of which are critical for achieving sustained returns and minimizing drastic fluctuations in portfolio value.
  To address this issue, we propose incorporating modifications to the reward function to better emphasize long-term performance metrics and mitigate excessive reward volatility. Such adjustments can help the agent focus on maintaining a more stable growth trajectory, even during turbulent market conditions. The comparison of reward function outcomes will be shown in the later section.

Our findings suggest that while the baseline strategy effectively captured market trends in a relatively stable environment, its applicability to highly volatile and rapidly evolving markets is limited. Addressing these shortcomings will require enhancements to the stop-loss mechanism and reward function, ensuring the strategy remains robust under diverse market conditions. After meticulous enhancement and modification, the upgraded ensemble strategy is able to achieve better performance.

5.2 Overall Strategy Performance

To evaluate the overall performance of the ensemble strategy, we conducted backtests and compared its results to those of individual agents under identical rebalancing and training mechanisms and the Dow Jones Industrial Average (DJI) index. The backtest graph (Figure 5-5) illustrates the cumulative return over time for each strategy. The ensemble strategy consistently outperformed individual agents and the DJI index, demonstrating its robustness and superior adaptability to market conditions.

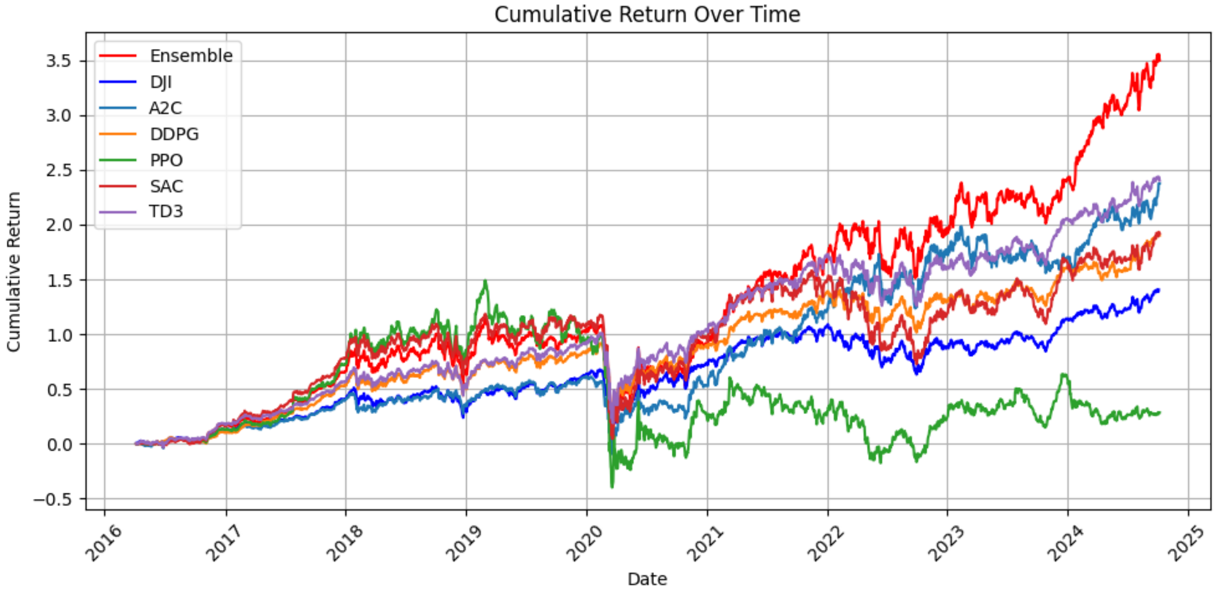


Figure 5-5 Strategy Performance Comparison with Individual Agents and DJI Index

The table below summarizes the performance of each strategy based on key financial metrics, including cumulative return, annual return, annual volatility, Sharpe ratio, and maximum drawdown.

Strategy	Cumulative Return (%)	Annual Return (%)	Annual Volatility (%)	Sharpe Ratio	Max Drawdown (%)
Ensemble	**349.95**	**41.17**	**24.05**	**1.71**	**-101.21**
DJI	140.60	16.55	17.87	0.93	-62.26
A2Cw	237.57	27.95	21.86	1.28	-71.84
DDPG	190.12	22.37	18.58	1.20	-76.91
PPO	28.48	3.35	37.18	0.09	-189.21
SAC	191.93	22.58	25.23	0.89	-114.10
TD3	240.19	28.26	19.41	1.46	-82.18

Table 5-5 Performance Metrics for Different Strategies

The ensemble strategy achieved a cumulative return of 349.95%, significantly higher than any individual agent or the DJI index. Its annual return of 41.17% also surpasses other strategies, reflecting its effectiveness in capturing profitable opportunities across diverse market conditions. The strategy also exhibited an annual volatility of 24.05%, slightly higher than that of most individual agents. This can be attributed to the rebalancing mechanism, which switches between different agents, introducing additional variance during transitions.

However, this variance is effectively managed, as evidenced by its high Sharpe ratio. With a Sharpe ratio of 1.71, the ensemble strategy outperforms all individual agents, indicating a better risk-adjusted return.

The maximum drawdown of the ensemble strategy was -101.21%, which, while significant, is smaller than the most extreme drawdowns observed in some individual agents (e.g., PPO with -189.21%).

Given that market volatility severely hindered the baseline strategy, we further isolated the period from 2020/01/01 to 2020/12/31—the most volatile period in our testing window. During this time, the ensemble strategy demonstrated its ability to adapt dynamically to market conditions, as shown in Figure 5-6, with the gray area indicating the turbulence index.

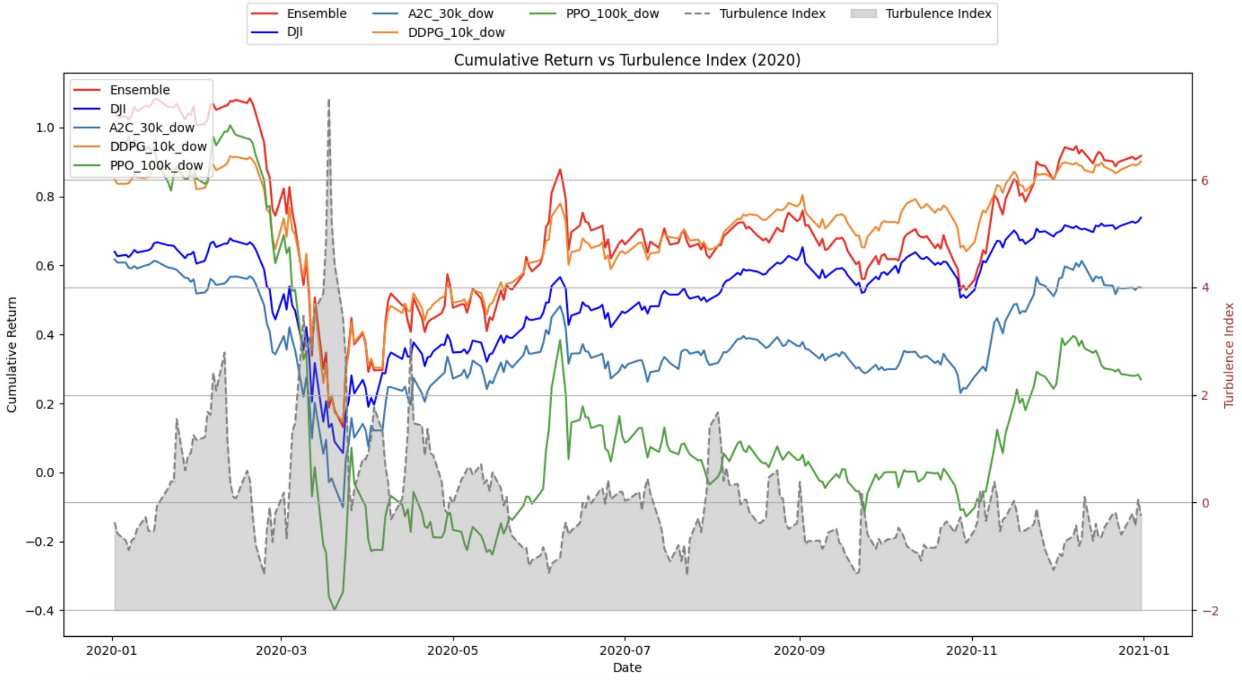


Figure 5-6 Volatile Backtest from 2020/01/01 to 2020/12/31

5.3 Separate Comparison

Reward Function

To evaluate the impact of different reward functions on the performance of the ensemble strategy, we compared the three designs as mentioned in Section 4.1:

- **The Profit and Loss (PnL)**;
- **The Weighted Combination of Current and Cumulative Returns**;
- **The Exponentially Weighted Historical Returns**.

The cumulative return results for these reward functions are presented in Figure 5-7, showing that the Exponentially Weighted reward outperformed the other methods, followed by the baseline PnL reward and the Weighted Combination reward.

Exponentially Weighted Reward achieved the highest cumulative return. This method's emphasis on recent returns while retaining a decayed influence from earlier returns allows the policy to adapt more effectively to market trends. It also stabilizes training by smoothing reward signals, reducing the noise associated with daily market fluctuations.

Baseline PnL Reward delivered a moderate performance, but falling short of the Exponentially Weighted reward. Its focus on short-term profits makes it more responsive to daily market changes but less effective in capturing long-term trends. Meanwhile, Weighted Combination Reward underperformed relative to the other two. Although it integrates a long-term component, the equal weight given to daily and cumulative returns appears to dilute its effectiveness.

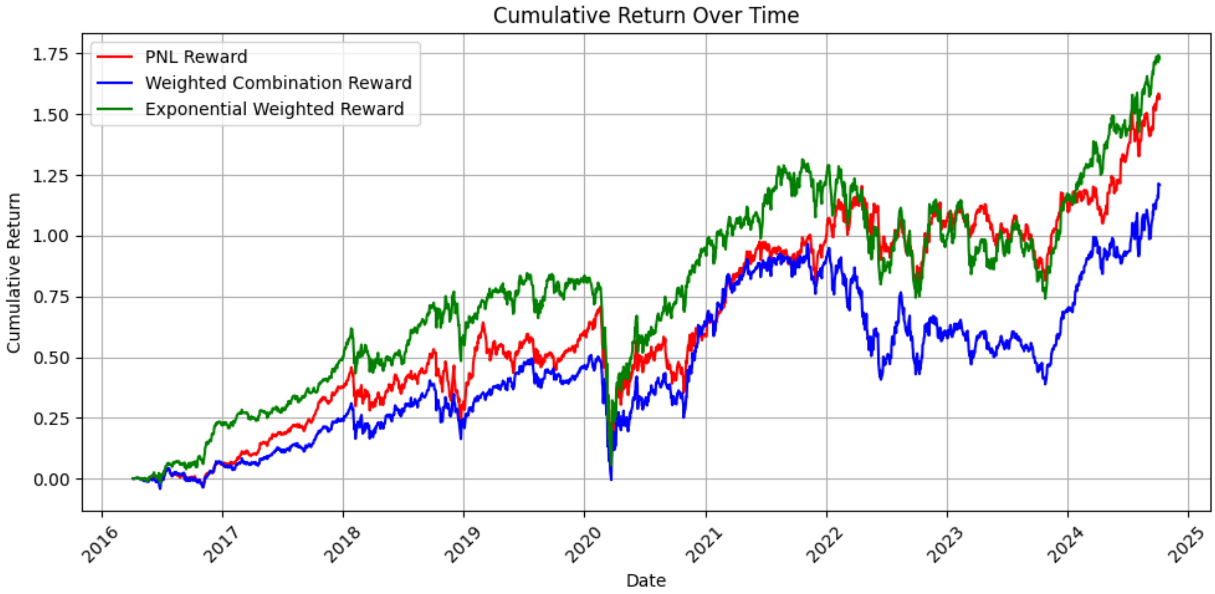


Figure 5-7 Reward Function Comparison from 2016 to 2024. The reward function comparison was conducted on the basis of 3 model ensemble and without transfer learning.

The results indicate that reward function design significantly impacts policy performance. While the PnL reward is simple and intuitive, it lacks the ability to account for long-term stability, especially during volatile market period. The Weighted Combination reward improves long-term considerations but struggles to balance this with short-term responsiveness. In contrast, the Exponentially Weighted reward seems to strike an effective balance, leveraging recent market dynamics while maintaining historical context through decayed returns. Hence, in the final strategy we design, we optimized the reward as the exponential weighted historical returns as the empirical test results suggested.

Stop-Loss Mechanism

The performance analysis of the proposed stop-loss mechanisms, compared to the baseline without stop-loss, highlights the trade-offs between risk mitigation and return maximization. As outlined in Section 4.2, we evaluate three distinct stop-loss mechanisms:

- **The Return-Based Stop-Loss**;
- **The Value-Based Stop-Loss**;
- **The Turbulence-Based Stop-Loss**;

The original turbulence-based stop-loss mechanism struggled with performance degradation over time. We believe that this was attributed to the monotonic nature of the turbulence index, which exhibited an increasing trend as markets evolved. This trend caused the turbulence index to frequently exceed the threshold in later rebalancing windows, leading to overly conservative halts in trading and missed opportunities during recovery periods.

To address this limitation, we detrended the turbulence index using a rolling z-score, thereby normalizing its fluctuations over time. The detrended turbulence index provided a more adaptive measure of market volatility and allowed for a fairer comparison across different stop-loss mechanisms.

Figure 5-8 indicates that while both return-based and value-based stop-loss mechanisms effectively reduce portfolio volatility, they also significantly constrain cumulative returns.

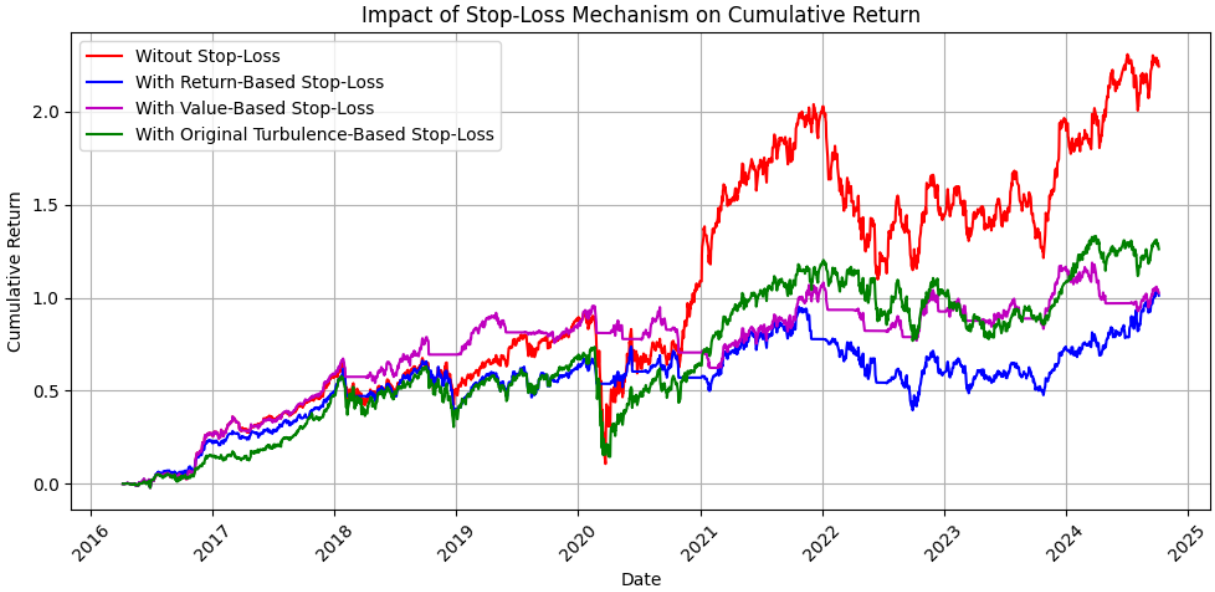


Figure 5-8 Stop-Loss Mechanism Comparison from 2016 to 2024. The stop-loss mechanism comparison was conducted on the basis of 5 model ensemble and without transfer learning.

All the stop-loss mechanisms significantly reduced annual volatility compared to the baseline. The value-based mechanism achieved the lowest volatility (10.70%), highlighting its stability during adverse market conditions.

Meanwhile, all stop-loss mechanisms constrained cumulative returns compared to the baseline. Considering Sharpe ratio, the effect of stop-loss mechanisms in terms of return constraints is more profound than risk control. All stop-loss mechanisms failed to meet higher Sharpe ratio than non-stop-loss strategy.

As a result, during strategy implementation, we didn't select a stop-loss mechanism for use.

Strategy	**Cumulative Return (%)**	**Annual Return (%)**	**Annual Volatility (%)**	**Sharpe Ratio**	**Max Drawdown (%)**
No Stop-Loss	**224.29**	**26.39**	**21.38**	**1.23**	**-94.11**
Return-Based Stop-Loss	101.22	11.91	13.43	0.89	-55.70
Value-Based Stop-Loss	102.57	12.07	10.70	1.13	-33.29
Turbulence Stop-Loss	126.11	14.84	18.62	0.80	-58.95

Table 5-8 Performance Metrics for Different Stop-Loss Strategies

Model Selection

The adoption of the 5-model ensemble strategy, combined with transfer learning, demonstrates clear advantages over the original 3-model approach. Through enhanced diversity in algorithm selection and improved training efficiency, our updated methodology achieves superior trading performance across various market conditions.

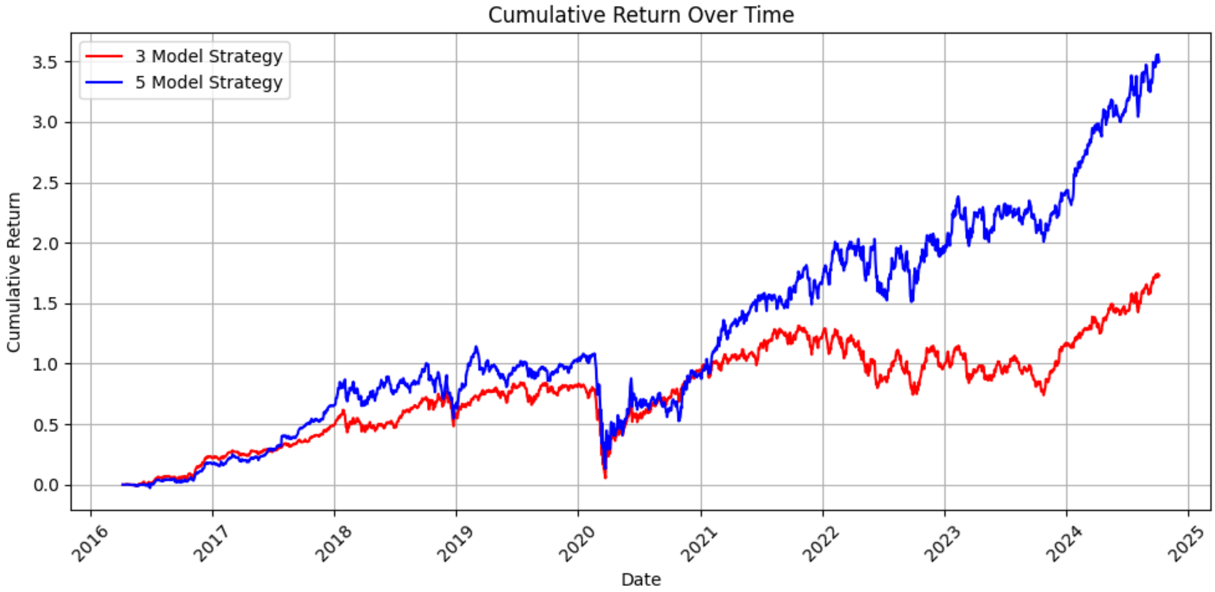


Figure 5-9 Model Selection Comparison from 2016 to 2024

Figure 5-9 illustrates the performance comparison between the original 3-model strategy (A2C, DDPG, PPO) and the enhanced 5-model strategy (A2C, DDPG, PPO, TD3, SAC). The 5-model ensemble outperformed the 3-model approach in cumulative returns across almost all validation periods.

By incorporating TD3 and SAC, the ensemble benefited from TD3's stability in continuous action spaces and SAC's adaptability to high-dimensional environments, resulting in more consistent returns. The additional diversity introduced by TD3 and SAC mitigated overfitting to specific market conditions. The 5-model ensemble demonstrated greater resilience, as SAC's maximum entropy framework facilitated better exploration in uncertain environments, while TD3's delayed updates provided more stable policy learning.

To address the increased computational demands of training five models, we also implemented transfer learning to improve efficiency while maintaining performance continuity. The computing time and efficiency has been greatly improved.

6 Closing Remarks and Future Directions

The results of this study highlight both the strengths and limitations of the deep reinforcement learning method for automated stock trading, providing valuable insights for further refinement. In particular, we identified key areas for improvement and future exploration:

- **Stop-Loss Mechanism:** Our analysis revealed that the original turbulence-based stop-loss mechanism, while effective during relatively stable periods, struggled to perform in high-risk environments post-2020. This issue, attributed to the increasing monotonic trend of the turbulence index, frequently triggered premature trading halts during volatile markets, leading to suboptimal returns.

  To address this, we experimented with two alternative mechanisms—return-based and value-based stop-loss strategies. While both effectively reduced portfolio volatility and drawdowns, they significantly constrained cumulative returns, ultimately underperforming the no-stop-loss baseline.

  This result underscores the complexity of balancing return generation and risk control. Future work could focus on hybrid approaches, such as adaptive stop-loss mechanisms that dynamically adjust thresholds based on market conditions, or machine learning models that predict the likelihood of extreme events to guide trading pauses.
- **Model Selection:** our transition from a 3-model ensemble to a 5-model strategy demonstrated the importance of model diversity in improving trading performance. By incorporating TD3 and SAC, we successfully enhanced robustness and adaptability across varying market conditions. However, the growing complexity of model selection comes at a cost: the computational power required to train and validate five models is substantial.

  Looking ahead, optimizing model selection will remain a critical research focus. As the number and sophistication of reinforcement learning algorithms continue to evolve, selecting the right combination of models for a given market environment will become increasingly important. Future efforts could explore multiple areas. For dynamic model selection, future scholars can implement meta-learning or ensemble learning methods to dynamically adjust the model pool based on current market characteristics. It is also possible to investigate ways to balance performance gains with computational efficiency by using lightweight architectures or distributed training frameworks.
- **Transfer Learning for Efficiency:** The incorporation of transfer learning into our methodology has demonstrated significant potential in addressing computational challenges. By leveraging pre-trained parameters from previous periods, we effectively reduced training time and computational overhead while enhancing model adaptability to new data. To further improve the models' responsiveness to evolving market conditions, future research could explore assigning higher weights to more recent samples, thereby prioritizing current market dynamics. Promising future directions include domain-specific pretraining to enhance robustness and continual learning to enable incremental updates without compromising prior knowledge.
- **Broader Topics:** Future research could delve into how reinforcement learning models can better align with portfolio-level optimization objectives, such as maximizing the Sharpe ratio, minimizing drawdowns, or achieving more consistent risk-adjusted returns, rather than concentrating solely on trade-by-trade rewards. Additionally, integrating alternative data sources—such as news sentiment, macroeconomic indicators, and real-time geopolitical events—into the reinforcement learning framework could further refine decision-making

and improve adaptability to diverse market conditions. Exploring multi-objective optimization frameworks and hybrid strategies that combine quantitative and qualitative data analysis could open new avenues for developing more robust and versatile trading strategies.

References

[1] Bruno Gašperov and Zvonko Kostanjčar. Market making with signals through deep reinforcement learning. *IEEE Access*, 9:61611–61622, 2021.

[2] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. https://github.com/hill-a/stable-baselines, 2018.

[3] Bouzgarne Itri, Youssfi Mohamed, Qbadou Mohammed, Bouattane Omar, and Touil Mohamed. Deep reinforcement learning strategy in automated trading systems. pages 1–8, 2023.

[4] Mark Kritzman and Yuanzhen Li. Skulls, financial turbulence, and risk management. *Financial Analysts Journal*, 66(5):30–41, 2010.

[5] Tianyuan Sun, Dechun Huang, and Jie Yu. Market making strategy optimization via deep reinforcement learning. *IEEE Access*, 10:9085–9093, 2022.

[6] Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. Deep reinforcement learning for automated stock trading: an ensemble strategy. 2021.