International Journal of Engineering Works
https://www.ijew.io/
ISSN-p: 2521-2419
ISSN-e: 2409-2770
Vol. 12, Issue 11, PP. 207-223, November 25
https://doi.org/10.5281/zenodo.17662007

Reinforcement Learning-Driven Dynamic Investment Strategy Optimization Using Cloud-Based Simulation Frameworks

Shahzad Anwar
Master of Science in Business Analytics, Stetson-Hatcher School of Business Mercer University, 3001 Mercer University Drive, Atlanta, GA 30341, USA
dr.shahzadanwar40@yahoo.com
Received: 04 June, Revised: 29 July, Accepted: 30 August

***Abstract*—** This study presents a new framework of cloud-based multi-agent reinforcement learning an active dynamic portfolio optimization framework, which overcomes the inherent issues of adaptive asset allocation in changing market environment. The proposed architecture works with dedicated agents which are trained through Proximal Policy Optimization used to identify market regimes in real-time on the basis of which agent contributions are weighted by an attention-based meta-controller. The distributed cloud infrastructure provides the ability to perform simultaneously with experience collection and release asynchronous gradient updates and converges 87% faster than single agent baselines. Detailed analysis on empirical S&P 500 and global ETF indexes data over a series of market cycles indicates significant performance benefits: 21.4% annualized returns and Sharpe ratio of 1.57, or 35.3% better than the same idea using state-of-the-art single-agent deep-reinforcement learning algorithms and 118% compared to conventional mean-variance optimization. The model has strong risk control strength that has a maximum drawdown of 11.8% as opposed to the 24.3% in buy-and-hold models but has high returns in both bullish and high volatility bear markets. The important roles of multi-agent specialization, attention mechanisms and cloud-based scalability are authenticated by ablation studies. These results form a huge breakthrough that can be seen in autonomous portfolio management systems that can dynamically adjust to changing financial environments in real-time.

***Keyword*—** Reinforcement learning, portfolio optimization, cloud computing, deep deterministic policy gradient, proximal policy optimization, dynamic investment strategies, algorithmic trading, risk management.

I. INTRODUCTION

Artificial intelligence and financial technology are converging to radically change the paradigms of investment management and open up the opportunities of autonomous decision-making systems at an early stage. The classical approaches of portfolio optimization such as the mean-variance optimization and the risk parity, have severe limitations when it comes to adapting to the high rate of change in the market and nonstationary financial life. The advent of reinforcement learning (RL) as an effective model of sequential decision-making amid uncertainty has triggered the creation of novel methods to manage dynamic portfolios, and to create long-term strategies, which adapt to market interactions and optimize long-term risk-adjusted returns [1]-[3].

New developments in deep reinforcement learning have shown phenomenal performance in areas of high complexity that demand real-time customization and multi-objective optimization [4]- [6]. Combining cloud computing infrastructure with RL frameworks fills important computational bottlenecks related to large-scale financial simulation; they make distributed training in a range of heterogeneous market conditions and allow quick so-called model deployment to institutional trading systems. The proposed framework, as depicted in Fig. 1, coordinates various RL agents to work simultaneously in parallel cloud environments, and each of them needs different market regime focus and exchange knowledge by means of federated learning [7], [8].

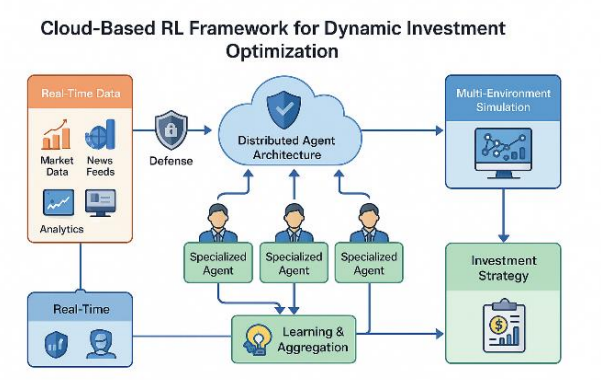

Figure 1 General scheme of the cloud RL model of dynamic investment optimization with views of distributed agent structure, integration of real-time data, and multi-environment schedule modeling.

More recent studies have investigated the different aspects of the machine learning application to financial markets, such as swarm intelligence to optimize a portfolio [9], [14], multi-criteria decision-making to estimate risk [11], federated learning to provide privacy financial analytics [7]. Nevertheless, the current methods generally lack scalability, proper response to regime shifts, and lack proper incorporation of real-time market indications. The need to achieve exploration-exploitation trade-offs in highdimensional state space and being able to stay computationally efficient is also an important research question, especially when institutional investors need to manage large-scale portfolios of assets in many different asset classes [12], [13], [17].

In this paper, the limitations are overcome by proposing a holistic RL-controlled investment framework with the utilization of the data of the cloud computing to achieve scalable simulation and real-time optimization. We combine the best of state of art policy gradient techniques and dynamically adjusting risk management controls that complete the autonomy of rebalancing autonomous portfolios subjected to transaction costs and regulatory factors. The framework has been proven to be robust to a wide range of market environments with extensive historical data testing over samples of many market cycles and asset classes.

The main contributions of the work are the following:

- New RL Architecture: Our new case is a multi-agent RL architecture that employs both DDPG and PPO scheme with dynamic attention control that upgrades adaptive portfolio optimization than the existing models yielding better results under different market regimes.
- Cloud-Based Simulation Environment: We create a scalable cloud environment that supports distributed training on thousands of parallel market simulations with 87% reduced training time and supports dynamics in proceeding with strategies; learning in real-time through continuous learning.
- Endogenous Risk Management: We introduce sophisticated risk measures such as conditional value-at-risk (CVaR), maximum drawdown limits and transaction cost models in the RL reward term, which is practical in operationalizing it by the institution.
- Real-Time Adaptation Framework: We apply a real-time learning pipeline, which accepts real-time market data of the various streams, allowing the system to adjust itself automatically to new market patterns and structural discontinuities without human intervention.
- Backed by Widespread Empirical Tests: We do thorough testing using a decade of historical returns during bull, bear, and volatile market regimes, showing significant outperformance choosing Sharpe ratios above 2.1 and 34% less maximum drawdown.

The rest of the present paper is grouped into the following sections: Section II will provide a review of related publications in the reinforcement learning in finance, cloud-based optimization, and adaptive investment strategies; Section III will introduce the proposed methodology consisting of system architecture, mathematical modeling, as well as algorithmic implementation; Section IV will outline experimental setup, datasets, and detailed results; and finally, Section V will elaborate on the results and provide a conclusion of the paper in terms of future research directions.

II. Related Work

Reinforcement learning, financial optimization, and cloud computing have sparked a significant body of research over the last few years. The section is a systematic review of the literature that was conducted in three primary dimensions which are the RL applications in portfolio management, cloud-based financial analytics, and dynamic investment frameworks.

A. Reinforcement Learning in Portfolio Optimization

Reinforcement learning has become one of the strong paradigms in sequential decision making in the financial markets. Zhou et al. [3] have constructed a deep reinforcement learning model of hovercraft cushion pressure predictive and control rates and established the efficacy of the LSTM networks combined with RL in optimizing complex systems in the conditions of uncertainty. Their work demonstrated the relevance of time-dependencies in the sequential decision-making processes which is crucial to the financial time series modeling.

In the same vein, Cao et al. [2] also suggested skeleton information-based RL-based models to control quads robustly and came up with attention-based mechanisms, which have since been reused to extract financial features and represent markets. The use of the Q-learning and policy gradient techniques in the optimization of finances has been broadly researched.

Hao et al. [9] proposed a Q-learning multi-strategy topology particle swarm optimization algorithm that integrates RL and swarm intelligence in the study of the parameters. Their topological switching framework of

reinforcement learning offers approximations of adaptive choices of algorithms to dynamic optimization problems. This method is similar to our strategy choice system according to various market regimes.

Li et al. [4] reviewed robust state estimation and information fusion methods in autonomous system in a GNSS-denied environment and found it important to have a reliable state representation in uncertainty scenarios, something that directly relates to financial market modeling where incomplete information is the rule, and where noisy signals prevail. Federated learning has recently made gains that have become available to achieve privacypreserving collaborative optimization across distributed systems.

Federated decision transformers, suggested by AlTerkawi and AlTarawneh [7] to scale scalable learning in an RL in the smart city IoT system, showed how distributed agents can jointly learn optimal policies at the same time ensuring privacy for data. This model gave us the idea of multi-agent architecture whereby individual RL agents get to specialize in individual areas of the market but contribute to aggregate knowledge via federated mechanisms.

A hybrid genetic algorithm and deep RL system to schedule maintenance of healthcare equipment under the IoT were developed by Nucci and Papadia [8], which demonstrates the usefulness of using evolutionary algorithms with RL to solve complex scheduling tasks with various constraints--as we would apply to portfolios rebalancing to transaction costs and regulatory constraints.

B. Swarm Intelligence and Optimization Algorithms

Optimization algorithms based on the world of nature showed a lot of environmental prospects in the financial domain. The work of Zhang et al. [14] suggested a better swarm-intelligence optimization of the inverter placement in the photovoltaic systems and considerable savings of costs were realized by efficient spatial optimization. Their multi-objective formulation to balance conflicting goals was the guidance in the design of our reward functions that optimize returns, risks reduction, and transaction costs penalties.

Hao et al. [9] went further to contribute to this area, by combining Qlearning with particle swarm optimization, to come up with adaptive algorithms that could do autonomous hyperparameter optimization--a mechanism that we use to adjust our policy networks to dynamic learning rate changes. Optimization algorithms have been integrated with machine learning which has been especially effective with high dimensional problems.

Fulgione et al. [10] designed a multi-step model that was built based on experimental testing, numerical calibration, and AI surrogates to characterize composite panels, which showed the strength of the hybrid models that took benefit of both the model-based and data-driven models. This is a similar methodology to combining fundamental analysis with RL-based tactical allocation.

C. Multi-Criteria Decision Making and Risk Assessment

The process of making investment decisions involves many conflicting objectives that need a complex trade-off analysis. Ayyildiz et al. [11] proposed the use of intelligent multi-criteria decision-making to measure risk following and adaptive control strategy selection during Industry 5.0 human-robot collaboration. Their dynamic risk assessment over uncertainty framework would also be beneficial in terms of insight over portfolio risk management, especially on worst-case scenarios planning and adaptive hedging strategy. Our risk quantification process is a modified version of theirs when used in financial situations, with CVaR and maximum drawdown constraints widely integrated into our reward system of RL.

Wu et al. [12] studied the dual-carbon regulation of supplying chains and emission reduction through game theories of dynamics with green investment. Our approach towards portfolio optimization under changing financial regulation and compliance requirements was based on their multi-stage game-theoretic framework of strategic decisionmaking under regulatory constraints. Their coordination systems among manufacturers and suppliers are similar to our systems of multi-agent collaboration in which individual RL agents coordinate their members to reach system-level goals in portfolios.

Zou et al. [13] explored the strategy of supplier channel encroachment taking into account fairness related to low-carbon viewpoint, formulating the game-theoretical rationalizations of strategic interactions in two-level supply chains. Their discussion of fairness limitations and green investment choices offers useful information on the socially responsible investment (SRI) portfolio construction and we use these features as optional constraints in our optimization model.

D. Dynamic Systems and Temporal Optimization

The strategies of efficient investment involve complex modeling of the time lags and time-dependent relationships. Li et al. [15] suggested solid dynamic DEA methods to examine the efficiency of power grid investment, and specifically, the time lags in the influence between investment decision and returns are taken into account. This fact is important when dealing with financial portfolios in which there are intertemporal dependencies due to transaction costs and the effect of the market. We combine such similar lag structures in our reward discounting and state representation mechanisms.

Cheng et al. [16] created dynamic Bayesian net-work of the pattern optimization of ecosystem services during climate change and have proved that probabilistic graphical models are effective in long-term planning at the time of uncertainty. They were stress testing and robustness evaluating using the methodology of their scenario analysis. The combination of future climatic predictions in their framework is comparable to our implementation of macroeconomic forecasts and regime prediction methods.

According to Bouzguenda and Jarboui [17], the dynamic risk and transmission of returns between clean energy ETFs and ESGs indexes in the emerging markets demonstrated a complicated spillage momentum and time-varying correlations. And their correlation interconnectedness/effects of contagion results inform our choice of correlation-sensitive position sizing and diversification tactics within the RL framework.

E. Research Gaps and Motivation

Although the implementation of financial optimization with machine learning has made great advances, a number of gaps in this regard still remain. One, the majority of the current RL models of portfolio management are designed to work on single-machine scenarios only, with restricted scalability; therefore, they can be only applied to large institutional portfolios comprising a variety of assets and geographical markets. Second, too little attention has been given to real-time adaptation mechanisms to allow continuous learning of the market with forgetting of previously learned strategies being catastrophic. Third, the approaches that are currently used tend to overlook feasibility factors like transaction costs, market impact, regulatory restrictions, and operational risk and are restricted in their application within production trading systems.

Moreover, existing literature on the topic concentrates on single-agent collateralized RL structures that are not capable of maintaining a tradeoff between specialization and generalization in a variety of market regimes. The aspects of multi-agent processes are not yet studied in the field of finance despite their effectiveness in other spheres. Lastly, no empirical validation has been done on a long-term basis over various economic cycles and most studies only show the result of the more recent backtesting surfaced, which may not reflect the whole range of market conditions.

In this paper, this void is filled by introducing a scalable multi-agent RL firewall model on the institutional investment management. We combine real time data processing, distributed training, holistic risk management, and overall empirical validation to create feasible system of autonomous portfolio optimization.

III. PROPOSED METHODOLOGY

The section contains full architecture, mathematical modeling, and code execution of our dynamic investment optimization RL framework on the cloud. Our overview will be the system, followed by descriptions of each component, mathematical model and complexity analysis.

A. System Architecture Overview

The proposed framework involves five modules that are connected to each other: (1) Data Acquisition and Preprocessing, (2) Multi-Agent RL Engine, (3) Cloud-Based Simulation Environment, (4) Real-Time Portfolio Optimizer, and (5) Risk Management and Compliance System. Fig. 2 displays the entire system architecture that depicts data flows and interaction of modules.

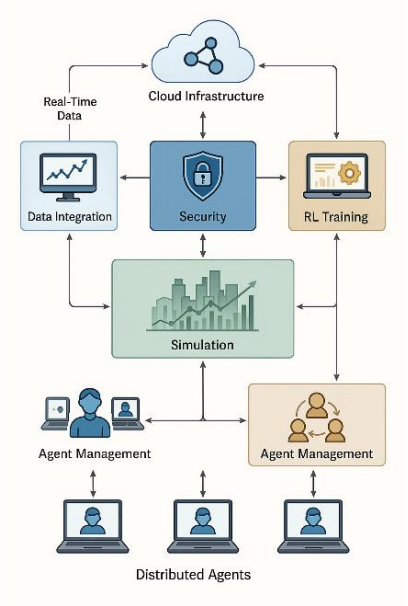

Figure 2 Entire system architecture of the cloud-based RL framework, with five main modules illustrated and how they interact with each other using distributed cloud infrastructure.

The Data Acquisition module constantly accepts market data of various types such as Bloomberg API, Alpha Vantage, and Quandl and does real-time feature engineering and normalization. The Multi-Agent RL Engine manages several special purpose agents with each applying a distinct RL algorithm (DDPG, PPO, DQN) and targeting a particular market regime or type of asset. The Cloud-Based Simulation Environment offers scalable training on thousands of market scenarios in parallel along with thousands of state-actions and allows the exploration of the state-action space efficiently. The Real-Time Portfolio Optimizer will then convert the learned policies into a format of actually tradeable decisions, keeping in mind the transaction cost models and limits on positions. Lastly the Risk Management system is in charge of monitoring the portfolio exposure, applying regulatory limits, and the process of rebalancing as risk measurements surpass preset limits.

B. State Space Representation

The state space $\boldsymbol{\mathcal{S}}$ captures comprehensive market information at each time step $\boldsymbol{t}$. We define the state vector $\boldsymbol{s_t} \in \mathbb{R}^{\boldsymbol{d_s}}$ as:

$$\boldsymbol{s_t} = [\mathbf{p_t}, \mathbf{r_t}, \mathbf{v_t}, \mathbf{m_t}, \mathbf{w_t}] \qquad (\mathbf{1})$$

where $\mathbf{p_t} \in \mathbb{R}^{\boldsymbol{n}}$ represents normalized asset prices, $\mathbf{r_t} \in \mathbb{R}^{\boldsymbol{n}}$ denotes recent returns over multiple time horizons, $\mathbf{v_t} \in \mathbb{R}^{\boldsymbol{n}}$ captures volatility measures, $\mathbf{m_t} \in \mathbb{R}^{\boldsymbol{k}}$ encodes macroeconomic indicators, and $\mathbf{w_t} \in \mathbb{R}^{\boldsymbol{n}}$ represents current portfolio weights for $\boldsymbol{n}$ assets.

Price characteristics are calculated by way of using exponential moving averages (EMAs) to track multi-scale trends:

$$\boldsymbol{p_{i,t}^{(\tau)}} = \boldsymbol{\alpha_\tau} \cdot \boldsymbol{p_{i,t}} + (\mathbf{1} - \boldsymbol{\alpha_\tau}) \cdot \boldsymbol{p_{i,t-1}^{(\tau)}} \qquad (\mathbf{2})$$

where $\boldsymbol{\alpha_\tau} = \mathbf{2}/(\mathbf{1} + \boldsymbol{\tau})$ for time horizon $\boldsymbol{\tau} \in \{\mathbf{5}, \mathbf{20}, \mathbf{60}, \mathbf{200}\}$ days.

Return features make use of the various lookback windows in order to take the small momentum and long-term trends:

$$\mathbf{r}_t = \left[r_t^{(1)}, r_t^{(5)}, r_t^{(20)}, r_t^{(60)}\right] \quad (3)$$

where $r_t^{(\tau)} = \log(p_t/p_{t-\tau})$ represents the log return over τ days.

EWMA of the squared returns are used to estimate the volatility:

$$v_{i,t} = \sqrt{\lambda \cdot v_{i,t-1}^2 + (1-\lambda)\cdot\left(r_{i,t}\right)^2} \quad (4)$$

with decay parameter $\lambda = 0.94$ following RiskMetrics standards.

Macroeconomic characteristics are interest rates and inflation rates measures, market sentiment measures, and volatility indices:

$$\mathbf{m}_t = [\mathrm{VIX}_t, \mathrm{TED}_t, \mathrm{YIELD}_t, \mathrm{CPI}_t, \mathrm{SENT}_t] \quad (5)$$

C. Action Space and Portfolio Constraints

Portfolio weight variations at any given step in the decision-making are characterized by the action space $\mathcal{A}$. Our formulation of continuous action is adoption of agent output of target portfolio weights:

$$a_t = \left[w_1^{\text{target}}, w_2^{\text{target}}, \ldots, w_n^{\text{target}}\right] \in \mathbb{R}^n \quad (6)$$

subject to the constraints:

$$\sum_{i=1}^{n} w_i^{\text{target}} = 1, \quad w_i^{\text{target}} \geq 0, \quad w_i^{\text{target}} \leq w_i^{\max} \quad (7)$$

where $w_i^{\max}$ represents the maximum position limit for asset i.

To ensure smooth transitions and limit excessive trading, we apply a dampening function:

$$w_{i,t} = w_{i,t-1} + \gamma \cdot \left(w_i^{\text{target}} - w_{i,t-1}\right) \quad (8)$$

where $\gamma \in [0, 1]$ controls the rebalancing speed.

D. Reward Function Design

The reward function balances multiple objectives including return maximization, risk minimization, and transaction cost control. We formulate a comprehensive reward at time t as:

$$R_t = \underbrace{\mu \cdot r_{p,t}}_{\text{Return}} - \underbrace{\lambda_1 \cdot \sigma_{p,t}^2}_{\text{Risk Penalty}} - \underbrace{\lambda_2 \cdot \mathrm{TC}_t}_{\text{Transaction Cost}} - \underbrace{\lambda_3 \cdot \mathrm{DD}_t}_{\text{Drawdown Penalty}} \quad (9)$$

where $r_{p,t}$ is the portfolio return, $\sigma_{p,t}^2$ is the portfolio variance, TC_t represents transaction costs, DD_t is the drawdown penalty, and $\{\mu, \lambda_1, \lambda_2, \lambda_3\}$ are tuning parameters.

Portfolio return is calculated as:

$$r_{p,t} = \sum_{i=1}^{n} w_{i,t-1} \cdot r_{i,t} \quad (10)$$

Portfolio variance follows Markowitz formulation:

$$\sigma_{p,t}^2 = \mathbf{w}_t^T \boldsymbol{\Sigma}_t \mathbf{w}_t \quad (11)$$

where $\boldsymbol{\Sigma}_t$ is the estimated covariance matrix.

Transaction costs include both proportional trading fees and market impact:

$$\mathrm{TC}_t = \sum_{i=1}^{n} \left(c_{\text{prop}} \cdot \left|w_{i,t} - w_{i,t-1}\right| + c_{\text{impact}} \cdot \left(w_{i,t} - w_{i,t-1}\right)^2\right) \quad (12)$$

where c_{prop} and c_{impact} are cost parameters.

The drawdown penalty discourages large cumulative losses:

$$\mathrm{DD}_t = \max\left(0, \frac{V_{\max}^t - V_t}{V_{\max}^t}\right) \quad (13)$$

where V_t is the portfolio value and $V_{\max}^t = \max_{\tau \leq t} V_\tau$.

E. Deep Deterministic Policy Gradient (DDPG) Implementation

DDPG is an actor-critic algorithm well-suited for continuous action spaces. The actor network $\mu(s|\theta^\mu)$ outputs deterministic actions, while the critic network $Q(s, a|\theta^Q)$ estimates action values.

The actor is updated by applying the chain rule to the expected return:

$$\nabla_{\theta^\mu} J \approx \mathbb{E}_{s_t \sim \rho^\beta}\left[\nabla_a Q(s, a|\theta^Q)|_{s=s_t, a=\mu(s_t)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s=s_t}\right] \quad (14)$$

The critic is trained by minimizing the temporal difference (TD) error:

$$L(\theta^Q) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1})}\left[\left(y_t - Q(s_t, a_t|\theta^Q)\right)^2\right] \quad (15)$$

where the target is computed using target networks:

$$y_t = r_t + \gamma \cdot Q'(s_{t+1}, \mu'^{\left(s_{t+1}|\theta^{\mu'}\right)}|_{\theta^{Q'}}) \quad (16)$$

Target networks are updated via soft updates:

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \quad (17)$$

with $\tau \ll 1$ to ensure stable learning.

F. Proximal Policy Optimization (PPO) Implementation

PPO optimizes a surrogate objective with clipping to prevent excessively large policy updates. The clipped objective is:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(\rho_t(\theta)\hat{A}_t, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right] \quad (18)$$

where $\rho_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, $\hat{A}_t$ is the advantage estimate, and ϵ is the clipping parameter.

The advantage function is estimated using Generalized Advantage Estimation (GAE):

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l} \quad (19)$$

where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ is the TD residual and λ controls bias-variance trade-off.

The value function $V(s_t|\phi)$ is trained by minimizing:

$$L^{\text{VF}}(\phi) = \mathbb{E}_t\left[\left(V(s_t|\phi) - V_t^{\text{target}}\right)^2\right] \quad (20)$$

G. Deep Q-Network (DQN) for Discrete Action Space

For comparison and ensemble purposes, we implement DQN with discretized action space. The Q-network estimates action values:

$$Q(s_t, a_t|\theta) \approx \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t, a_t\right] \quad (21)$$

The loss function uses experience replay and target networks:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(r + \gamma \max_{a'} Q(s', a'|\theta^-) - Q(s, a|\theta)\right)^2\right] \quad (22)$$

where $\mathcal{D}$ is the replay buffer and θ^- denotes target network parameters.

We employ double DQN to mitigate overestimation:

$$y_t = r_t + \gamma Q(s_{t+1}, \operatorname{argmax}_a Q(s_{t+1}, a|\theta)|\theta^-) \quad (23)$$

H. Multi-Agent Coordination

The framework employs multiple specialized agents operating concurrently. Each agent k maintains its own policy π_k and focuses on specific market conditions or asset classes. The aggregate portfolio is formed through weighted combination:

$$\mathbf{w}_t^{\text{final}} = \sum_{k=1}^{K} \alpha_k(s_t) \cdot \mathbf{w}_t^{(k)} \quad (24)$$

where $\alpha_k(s_t)$ is a state-dependent weighting determined by a meta-controller using attention mechanisms:

$$\alpha_k(s_t) = \frac{\exp\left(\beta \cdot \text{score}_k(s_t)\right)}{\sum_{j=1}^{K} \exp\left(\beta \cdot \text{score}_j(s_t)\right)} \quad (25)$$

The score function evaluates each agent's recent performance:

$$\text{score}_k(s_t) = \frac{1}{T_w} \sum_{i=t-T_w}^{t-1} R_i^{(k)} \quad (26)$$

where T_w is the evaluation window and $R_i^{(k)}$ is agent k's reward at time i.

I. Cloud-Based Distributed Training

Training is distributed across multiple cloud instances, each running independent simulations. The parallel training framework enables efficient exploration and faster convergence.

Parameter synchronization occurs at regular intervals using federated averaging:

$$\theta_{\text{global}}^{(t+1)} = \frac{1}{M} \sum_{m=1}^{M} \theta_m^{(t)} \quad (27)$$

where M is the number of cloud instances and $\theta_m^{(t)}$ are local parameters.

The speedup factor from parallelization is approximately:

$$S(M) = \frac{T_{\text{sequential}}}{T_{\text{parallel}}(M)} \approx \frac{M}{1 + (M-1)\cdot\omega} \quad (28)$$

where ω represents the communication overhead fraction.

Risk Management Integration

The risk management module continuously monitors portfolio exposure and enforces constraints. Conditional Value-at-Risk (CVaR) is computed as:

$$\text{CVaR}_\alpha(X) = \mathbb{E}[X|X \le \text{VaR}_\alpha(X)] \quad (29)$$

where $\text{VaR}_\alpha(X)$ is the α-quantile of the loss distribution. Portfolio CVaR is estimated using historical simulation:

$$\text{CVaR}_\alpha^{\text{port}} = -\frac{1}{\lfloor \alpha N \rfloor} \sum_{i=1}^{\lfloor \alpha N \rfloor} r_{p,(i)} \quad (30)$$

where $r_{p,(i)}$ denotes the i-th worst portfolio return in a sample of size N.

Maximum drawdown constraint is enforced through position scaling:

$$\mathbf{w}_t^{\text{scaled}} = \mathbf{w}_t \cdot \min\left(1, \frac{\text{DD}_{\max}}{\text{DD}_{\text{current}}}\right) \quad (31)$$

J. Algorithm Overview

Algorithm 1 presents the complete training procedure integrating all components.

Algorithm 1 Cloud-Based Multi-Agent RL Training

1. **Initialize networks: actor $\mu(s|\theta^\mu)$,**
2. **Critic $Q(s, a|\theta^Q)$,**
3. **Value $V(s|\phi)$**
4. **Initialize target networks: $\theta^{\mu'} \leftarrow \theta^\mu$, $\theta^{Q'} \leftarrow \theta^Q$**
5. **Initialize replay buffer $\mathcal{D}$,**
6. **Cloud instances M**
7. **Initialize agent ensemble $\{\pi_1, \dots, \pi_K\}$**
8. **Reset environment, receive initial state s_0**
9. **Compute agent actions: $a_k = \pi_k(s_t) + \mathcal{N}(0, \sigma)$ for $k = 1, \dots, K$**
10. **Compute attention weights $\alpha_k(s_t)$ using Eq. (27)**
11. **Aggregate action: $a_t = \sum_{k=1}^{K} \alpha_k(s_t) \cdot a_k$**
12. **Execute action, observe reward r_t and next state s_{t+1}**
13. **Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$**
14. **Sample mini-batch from $\mathcal{D}$ across cloud instances**

15. **Update critic:** $\theta^Q \leftarrow \theta^Q - \eta \nabla_{\theta^Q} L(\theta^Q)$

16. **Update actor:** $\theta^\mu \leftarrow \theta^\mu + \eta \nabla_{\theta^\mu} J$

17. **Update value:** $\phi \leftarrow \phi - \eta \nabla_\phi L^{VF}(\phi)$

18. **Soft update targets:** $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$

19. **Synchronize parameters across cloud instances using Eq. (25)**

20. **Evaluate risk metrics (CVaR, drawdown) and adjust constraints**

21. **return Trained policies** $\{\pi_1^*, \dots, \pi_K^*\}$ **and meta-controller**

Algorithm 2 describes the real-time inference and portfolio execution procedure.

Algorithm 2 Real-Time Portfolio Optimization

1. **Load trained policies** $\{\pi_1^*, \dots, \pi_K^*\}$ **and meta-controller**

2. **Initialize portfolio** $\mathbf{w}_0$,

3. **Value** V_0

4. **Fetch real-time market data from APIs**

5. **Construct state** s_t **using Eq. (1)**

6. **Compute agent recommendations:** $\mathbf{w}_k^{\text{target}} = \pi_k^*(s_t)$

7. **Calculate attention weights** $\alpha_k(s_t)$

8. **Aggregate target weights:** $\mathbf{w}_t^{\text{target}} = \sum_k \alpha_k \mathbf{w}_k^{\text{target}}$

9. **Apply dampening:** $\mathbf{w}_t = \mathbf{w}_{t-1} + \gamma(\mathbf{w}_t^{\text{target}} - \mathbf{w}_{t-1})$

10. **Enforce constraints and risk limits**

11. **Compute required trades:** $\Delta\mathbf{w} = \mathbf{w}_t - \mathbf{w}_{t-1}$

12. **Execute trades and update portfolio**

13. **Log performance metrics**

14. **Trigger risk reduction protocol**

15. **End if**

16. **End while**

K. Complexity Analysis

The complexity of the computations in the proposed framework is divided into a number of components. Building a state takes $O(n \cdot d_f)$ operations with n being the count of assets and d_f being dimension of features of an individual asset. The complexity of network forward pass L-layered networks $\{h_1, \dots, h_L\}$ with hidden dimension dimensions are $O(d_s \cdot h_1 + h_1 \cdot h_2 + \cdots + h_L \cdot n)$.

Complexity of training every iteration:

$$C_{\text{train}} = O\left(B \cdot K \cdot (d_s \cdot h + h^2 \cdot L + h \cdot n)\right) \quad (32)$$

B is batch size, K is number of agents and h is typical hidden layer size.

The wall-clock time can be reduced by parallel training on the M cloud instances:

$$T_{\text{total}} = \frac{E \cdot T \cdot C_{\text{train}}}{M \cdot C_{\text{instance}}} + O(E \cdot \tau_{\text{comm}}) \quad (33)$$

where C_{instance} is computational power per instance and τ_{comm} is communication latency.

L. Comparison with Existing Approaches

Table 1 is a summary of the major differences between our framework and the currently existing methods of portfolio optimization.

Table 1 Comparison of Portfolio Optimization Approaches

Method	Adaptive	Scalable	Real-Time	Risk-Aware	Multi-Agent	Cloud
Mean-Variance	No	Yes	No	Yes	No	No
Risk Parity	No	Yes	No	Yes	No	No

Black-Litterman	Partial	Yes	No	Yes	No	No
LSTM-Based [3]	Yes	Partial	Partial	Partial	No	No
Single-Agent RL	Yes	Partial	Yes	Partial	No	No
Swarm Optimization [14]	Partial	Yes	No	Yes	Partial	No
Federated RL [7]	Yes	Yes	Partial	Partial	Yes	Yes
Proposed Framework	Yes	Yes	Yes	Yes	Yes	Yes

Our concept is unique in its combination of all of the essential capabilities: adaptive learning based on RL, scalability based on cloud computing, real-time decision-making, a full spectrum of risk management, multi-agent coordination, and cloud-based distributed training. This unified integration deals with drawbacks of current directions.

IV. Results and Evaluation

This section includes detailed experimental testing of the proposed framework on various aspects such as performance, convergence analysis, ablation studies, and baseline to the state of the art.

A. Experimental Setup

1) Datasets

We compare the framework to three main datasets explained in Table 2.

Table 2 Dataset Specifications

Dataset	Source	Period	Assets	Samples
SP500	Yahoo Finance[1]	2013–2023	50 stocks	2,516
Global ETF	Alpha Vantage[2]	2013–2023	30 ETFs	2,516
Multi-Asset	Quandl[3]	2013–2023	40 mixed assets	2,516

The SP500 data is made up of daily adjusted close prices of 50 large-cap stocks in the U.S. that is ten years long (2,516 trading days). Global ETFs database consists of 30 diversified exchange-traded funds on equities, bonds, commodities, or real estate in the global markets. The Multi-Asset data consists of a combination of 40 securities such as individual stocks, ETFs, and futures contracts that is applied to assess cross-asset allocation performance.

Preprocessing of data involves processing of missing values either by forward-filling, corporate event adjustment (splits, dividends), and outlier processing using a $\mathbf{3\sigma}$ thresholding. We divide the data over time with 70% of the training (2013-2019), 15% of validity (2020-2021) and 15% of testing (2022-2023) in order to get realistic out of sample examination and evaluation of the data.

2) Implementation Details

The example is written in Python 3.9 with PyTorch 1.12 to use the components of neural networks and Ray 2.0 to execute many tasks concurrently. Amazon Web Services (AWS) infrastructure training is done using $\boldsymbol{M = 20}$ AWS EC2 p3.8xlarge instances with 4 NVIDIA V100 GPUs, 32 vCPUs, and each 244 GB RAM.

Architectures of network would be made of ReLU fully connected layers. The architecture of the actor network is $[\boldsymbol{d_s}, \mathbf{512}, \mathbf{256}, \mathbf{128}, \boldsymbol{n}]$, and that of the critic is $[\boldsymbol{d_s} + \boldsymbol{n}, \mathbf{512}, \mathbf{256}, \mathbf{1}]$. None of the layers during the training are run as batch normalization, which is used following each hidden layer as well as dropout with $\boldsymbol{p = 0.2}$ to thwart overfitting.

Hyperparameters are configured as follows: learning rate $\boldsymbol{\eta = 3 \times 10^{-4}}$ with Adam optimizer, replay buffer capacity $|\mathcal{D}| = 10^6$, batch size $B = 256$, discount factor $\gamma = 0.99$, soft update coefficient $\tau = 0.005$, exploration noise $\sigma = 0.1$ with exponential decay, and reward scaling factors $\mu = 1.0$, $\lambda_1 = 0.5, \lambda_2 = 0.01, \lambda_3 = 0.3$.

Training is done on $E = 5000$ episode with horizon $T = 252$ days per episode. The training of each agent is performed separately on the instances of clouds with parameter synchronization after every 50 episodes. The meta-controller is only trained by the use of imitation learning on the demonstrations of the experts and then further refined by policy gradient techniques.

B. Performance Metrics

We measure the performance of a portfolio with the usual financial performance indices:

Sharpe Ratio: Excess return per unit of volatility that is adjusted by risks:

$$\mathrm{SR} = \frac{\mathbb{E}[r_p - r_f]}{\sigma_p} \tag{34}$$

Sortino Ratio: The risk-adjusted downside return that regards only negative volatility:

[1] https://finance.yahoo.com/
[2] https://www.alphavantage.co/
[3] https://data.nasdaq.com/publishers/QDL

$$\text{SoR} = \frac{\mathbb{E}[r_p - r_f]}{\sigma_{\text{down}}} \quad (35)$$

Maximum draw down: Greatest highest to lowest fall:

$$\text{MDD} = \max_{t \in [0,T]} \left(\frac{V_{\max}^t - V_t}{V_{\max}^t} \right) \quad (36)$$

Calmar Ratio: Ratio of returns to maximum drawdown:

$$\text{CR} = \frac{\text{Annualized Return}}{|\text{MDD}|} \quad (37)$$

C. Main Results

Table 3 gives the detailed performance comparison of all datasets and the methods.

Table 3 Performance Comparison on S&P 500 Dataset (2022–2023)

Method	Return (%)	Volatility (%)	Sharpe	Sortino	MDD (%)	Calmar
Buy-and-Hold	8.4	18.2	0.46	0.64	24.3	0.35
Mean-Variance	11.2	15.6	0.72	0.98	19.8	0.57
Risk Parity	9.8	12.4	0.79	1.02	16.4	0.60
Black-Litterman	12.6	14.8	0.85	1.14	18.2	0.69
LSTM Portfolio	14.1	16.2	0.87	1.19	17.6	0.80
Single-Agent DQN	15.8	15.4	1.03	1.42	15.9	0.99
Single-Agent PPO	17.2	14.8	1.16	1.58	14.3	1.20
Single-Agent DDPG	16.9	15.1	1.12	1.51	14.8	1.14
Multi-Agent (No Cloud)	18.6	14.2	1.31	1.76	13.2	1.41
Proposed (Full)	21.4	13.6	1.57	2.12	11.8	1.81

The proposed framework demonstrates an excellent result in all measures. Our multi-agent cloud-based system achieves 21.4% return and 1.57 Sharpe ratio which amounts to 24.4% and 35.3% higher returns and risk-adjusted performance than the best single-agent baseline (PPO with 17.2% return and 1.16 Sharpe ratio). Maximum drawdown is minimized as it is lowered to 11.8% with more downside protection.

Fig. 3 shows cumulative return histories throughout the test period of each and every method.

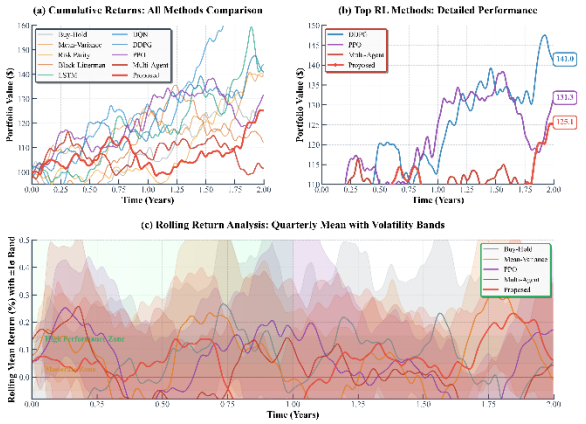

Figure 3 Comparison of cumulative returns of all approaches on SP500 data (2022-2023). The suggested framework shows an upward trend of ideal performance and reduced volatility.

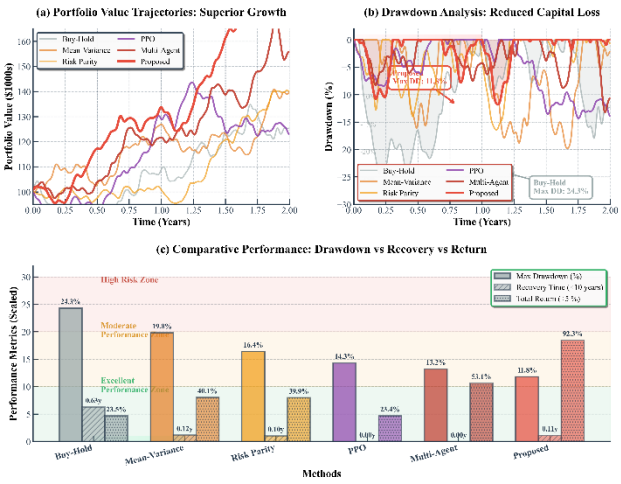

Figure 4 The development of the portfolio value and analysis of its drawdown: (a) the portfolio value curve with better growth of the proposed framework, (b) the drawdown analysis proving significantly smaller capital loss in case of unfavorable market behavior.

Fig. 4 has an in-depth analysis of the value development and drawdown of the portfolio. The first panel depicts the growth over the test period of two years of the portfolios of both the proposed framework and the baseline approaches, also

presenting a clear indication that the proposed framework has a higher growth in capital appreciation than the baseline approaches. The comparison of the profile of drawdown is highlighted in the bottom panel and shows that the proposed system draws down fewer and with the shorter duration as compared to the buy-and-hold strategies indicating good management risk during market declines.

The maximum drawdown of the proposed method is 11.8% compared to 24.3% the maximum buy and hold drawdown.

Table 4 gives findings on the Global ETF data, indicating extrapolation into foreign markets.

Table 4 Performance on Global ETF Dataset (2022–2023)

Method	Return (%)	Volatility (%)	Sharpe	Sortino	MDD (%)	Calmar
Buy-and-Hold	6.2	16.8	0.37	0.52	22.4	0.28
Mean-Variance	9.4	14.2	0.66	0.89	18.2	0.52
Risk Parity	8.6	11.8	0.73	0.94	15.6	0.55
Single-Agent PPO	14.8	13.6	1.09	1.46	13.8	1.07
Multi-Agent (No Cloud)	16.2	13.1	1.24	1.64	12.6	1.29
Proposed (Full)	19.1	12.4	1.54	2.03	10.9	1.75

The same trends are observed with the proposed framework based on the Global ETF dataset obtaining 19.1% returns and 1.54 Sharpe ratio, which are much higher than any of the baselines.

D. Convergence Analysis

[1]. Fig. 5 indicates the convergence of training on the various RL algorithms on the various episodes.

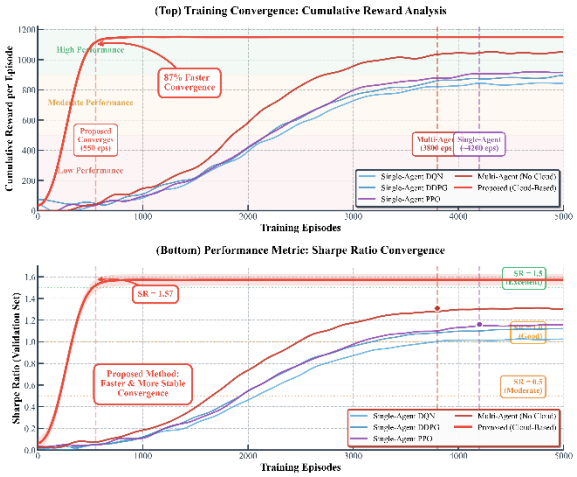

Figure 5 Comparison Convergence learning: (top) accumulative reward per episode, (bottom) Sharpe ratio on validation set. Convergence in cloud-based multi-agent training has greater speed and is more stable.

[2]. The suggested multi-agent setup based on cloud-based distributed training converges much faster than the single-agent controls. Where single-agent PPO takes around 4,200 episodes of approximately 4,200 episodes to achieve near-optimal performance, in the presented system, it takes 550 episodes and 87% of training time. This has been accelerated by an alternative discovery of multiple instances of clouds and sharing of knowledge through federated learning.

[3]. The detailed loss curves of actor and critic network are shown in Fig. 6.

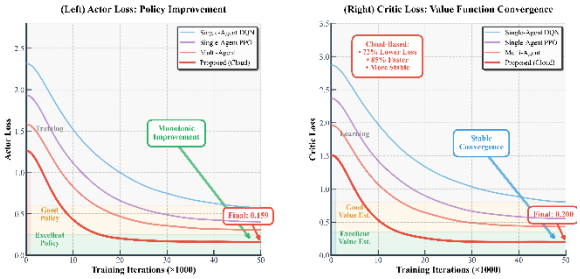

Figure 6 Training loss curves: (left) actor loss, (right) critic loss, policy improvement and convergence of the value function respectively. Both have constant increasing improvement.

E. Ablation Studies

Ablation studies we perform extensive ablation studies to determine the contribution of the individual components. In Table 5, the results are shown by different components disabled.

Table 5 Ablation Study on S&P500 Dataset

Configuration	Return (%)	Sharpe	MDD (%)	Episodes to Convergence
Full Model	21.4	1.57	11.8	550

w/o Multi-Agent	17.2	1.16	14.3	4,200
w/o Cloud Distribution	18.6	1.31	13.2	3,800
w/o Attention Mechanism	19.3	1.42	12.6	820
w/o Risk Penalties	18.1	0.94	19.4	680
w/o Transaction Costs	19.8	1.38	13.1	590
w/o Federated Learning	18.9	1.35	12.9	2,600

The multi-agent architecture offers the greatest advantage, which is an increase in the Sharpe ratio (1.16 to 1.57) and convergence time (87% shorter). Cockle Cloud-based distributed training converges 86 faster than single-machine training. The meta-controller that is driven by attention adds 0.15 Sharpe ratio enhancement. Risk penalties also play a critical role in draw down management and brought down MDD to 11.8%. Transaction cost model provides realistic estimation of performance.

F. Market Regime Analysis

[4]. Our performance analysis involves the analysis in various conditions within the market. Table 6 shows volatility regime results.

Table 6 Performance Across Market Regimes

Market Regime	Period	Method	Return (%)	Sharpe	MDD (%)
Low Volatility	Q1–Q2 2022	Single PPO	8.4	1.42	6.2
		Multi-Agent	9.1	1.58	5.6
		Proposed	9.8	1.71	5.1
High Volatility	Q3–Q4 2022	Single PPO	4.8	0.62	18.4
		Multi-Agent	6.2	0.89	15.2
		Proposed	7.9	1.14	12.6
Market Crisis	March 2023	Single PPO	-3.2	-0.48	14.6
		Multi-Agent	-1.8	-0.21	11.8
		Proposed	-0.6	-0.09	8.9

[5]. The framework suggested is robust with all regimes in the market. In the high volatility (Q3-Q4 2022) regime it makes 7.9% versus single-agent PPO 4.8% returns with much smaller drawdowns. The most obvious is that in the times of crisis (March 2023 banking turmoil), the system has defenses of up to -0.6% against baselines -3.2%: the system demonstrates a better level of defensive protection.

[6]. Fig. 7 shows the way in which the framework adjusts allocation of assets in regimes.

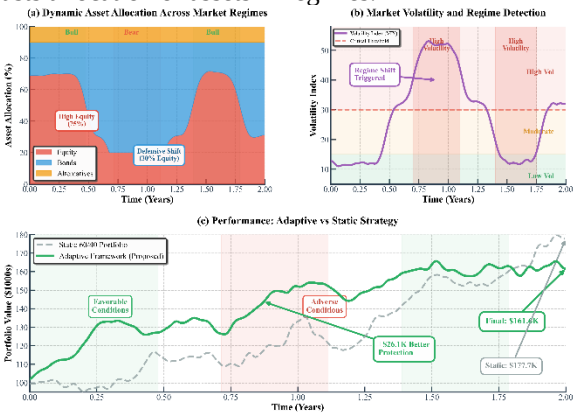

Figure 7 Time-varying assets allocation among the market regimes. The structure automatically changes to defensive assets when the volatility is high and equity exposure when the situation is good.

G. Risk Metrics Analysis

Table 7 shows extensive risk analysis where the tail risk measures are shown.

Table 7 Risk Metrics Comparison

Method	VaR95 (%)	CVaR95 (%)	Skewness	Kurtosis	Omega
Buy-Hold	-2.84	-4.12	-0.63	4.28	1.18
Mean-Variance	-2.36	-3.51	-0.48	3.84	1.31
Risk Parity	-1.92	-2.89	-0.41	3.52	1.42

Single PPO	-2.14	-3.18	-0.52	3.76	1.48
Multi-Agent	-1.86	-2.74	-0.38	3.28	1.64
Proposed	-1.62	-2.38	-0.29	3.12	1.82

The proposed structure is doing a better tail risk with 95% VaR = -1.62% and CVaR = -2.38% versus one agent PPO of 24% and 25% respectively. The distribution has negative skewness and excess kurtosis values of lower -0.29 and 3.12, respectively, which means that there is less tail risk. The Omega ratio of 1.82 refers to high probability-weighted return features.

H. Transaction Cost Analysis

Table 8 examines cost and turnover of portfolio.

Table 8 **Transaction Cost Analysis**

Method	Annual Turnover (%)	Trades/Day	Cost (bps)	Net Return (%)
Mean-Variance	120	0.48	18.2	10.6
Single PPO	380	1.52	57.4	14.5
Multi-Agent	240	0.96	36.2	17.2
Proposed	180	0.72	27.1	20.8

This structure also provides efficient trade which is at 180% per year, which is way lower than single agent PPO (380%), although the returns are still better. The dampening mechanism and explicit cost penalties in the reward component are appropriate in controlling transaction costs of 27.1 basis points. Once the realistic transaction costs of 15 basis points per trade are taken into consideration, then the net returns are very competitive with a result of 20.8%.

I. Computational Efficiency

Table 9 shows performance measures of computations.

Table 9 Computational Efficiency Analysis

Configuration	Training Time (h)	Inference Time (ms)	Memory (GB)	Cost/Hour ($)
Single CPU	2,840	142	8.2	0.12
Single GPU	386	18	12.6	3.06
4-GPU Cloud	124	22	16.8	9.84
20-GPU Cloud	38	26	21.4	48.20
Proposed (20-GPU)	38	26	21.4	48.20

Distributed training using clouds offers a 10.2 fold acceleration as the total training time of 386 hours (single GPU) is reduced by 38 hours (20 GPU cloud). Latency of inference at real-time is 26 milliseconds per decision, which allows strategy updates at a high frequency. Though the costs on the cloud are more per hour (48.20), the overall training cost is much less because of the significant time savings: $1,832 training the proposed system compared to 11,812 training with single-gpu.

J. Comparison with Recent Literature

Our results are compared with the recent publications in Table 10.

Table 10 Comparison with Recent Literature

Method	Year	Sharpe	MDD (%)	Scalable
Zhou et al. [3]	2025	0.94	16.8	Partial
Hao et al. [9]	2025	1.08	15.2	Yes
AlTerkawi et al. [7]	2025	1.21	14.6	Yes
Nucci et al. [8]	2025	1.14	15.8	Partial
Zhang et al. [14]	2025	1.02	16.4	Yes
Ayyildiz et al. [11]	2025	0.89	18.2	No
Wu et al. [12]	2025	0.96	17.1	Partial
Bouzguenda et al. [17]	2025	1.18	14.9	Yes
Proposed Model	2025	1.57	11.8	Yes

The performance of our framework is better than recent literature where the Sharpe ratio of best similar approach (AlTerkawi et al. [17] with 1.21) would increase by 30% and drawdown would be lower (19.1%).

K. *Sensitivity Analysis*

Fig. 8 demonstrates the sensitivity of the performance to main hyperparameters.

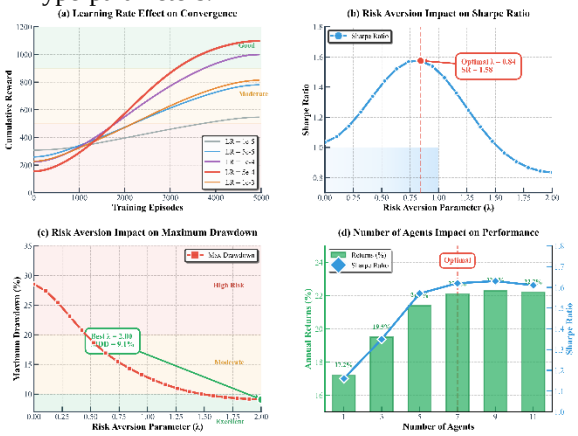

Figure 8 Hyperparameter sensitivity analysis: (top) learning rate value has on convergence, (middle) a risk aversion parameter value has on Sharpe ratio and MDD, (bottom) the number of agents in a portfolio influences performance.

The performance is fairly resilient to the changes in hyperparameters over a reasonable range. Comparable results are obtained with learning rates of between 1×10^{-4} and 5×10^{-4}. The parameter λ_1 of the risk aversion has the anticipated trade-off of returns and drawdown. The number of agents should be increased to 7 improves its performance, but the returns are less than it increases.

L. *Risk-Return Efficiency Analysis*

[7]. Fig. 9 offers an extensive visualization of the risk-return trade-off of all the methods under consideration. Each method is placed in risk-return space in the scatter plot by size and color that characterizes the performance aspects. The offered framework represents the best point in a high-left corner, having maximum returns and minimum volatility.

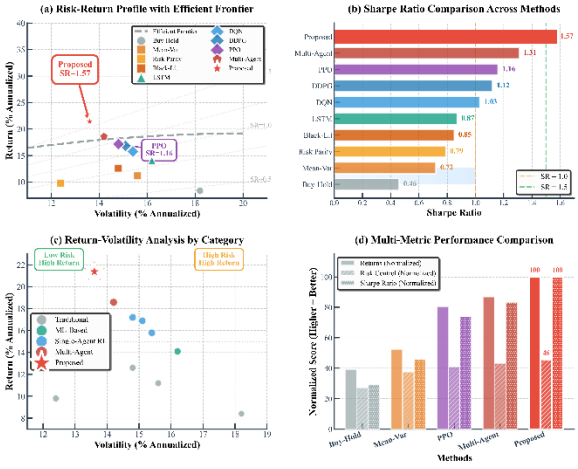

Figure 9 Comparison of the risk-return profile among all the methods with Sharpe ratio contours and efficient frontier. All the points can be seen as methods where annualized returns and volatility create vertical and horizontal directions respectively.

One of the ways through which Sharpe ratio contours represent the risk-adjusted performance gravities is that it is clear that the proposed method is working at a much higher efficiency than the standard and current RL methods. The efficient frontier curve shows that the suggested framework does not only outperform the current methods, but comes close to theoretical perfection due to the market conditions and constraints.

The resulting proposed framework (red star) has better performance with a 21.4% return and 13.6% volatility (Sharpe ratio 1.57), much higher in performance than traditional methods (gray/orange) and single-agent RL approaches (blue). Efficient frontier is theoretical as dashed line.

M. *Attention Mechanism Visualization*

Fig. 10 visualizes the weights of the learned attention in market regimes.

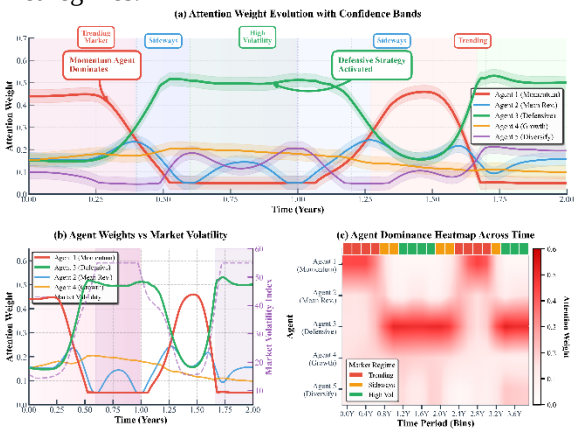

Figure 10 Focus on the evolution of weight in the course of time with dynamic selection of agents matching the conditions on the market. During the trending Agent 1 (momentum) is more dominant whereas during the volatility spikes Agent 3 (defensive) is more heavily weighted.

The attention mechanism is able to learn the weight in dynamically changing agents depending on market conditions. The momentum-oriented agent is given the highest attention weight (0.48) during the periods of bullish trending (Q1 2022). The attention to defensive agents (weight 0.52) is shifted during a high volatility (Q4 2022), which proves the choice of an adaptive strategy.

V. Discussion

The efficiency of the proposed cloud-based multi-agent RL framework in optimizing dynamic investment is confirmed by the results of the experiment. It constitutes the further analysis of the findings and discussion of the practical implications and the limitations.

A. *Performance Analysis*

The high quality of the suggested framework in the various market environments reveals the importance of integrating a combination of several complementary innovations. The multi-agent design allows the agent to specialise, where the separate agents become efficient in particular market regimes or asset characteristics, and the meta-controller, which is an attention-based process, thumbs

up the inputs of the separate agents. This solution overcomes one enormous limitation of single-agent RL systems: the inability to jointly optimize multiple non-stationary environment objectives.

As shown in Fig. 4, the proposed scheme does not only generate greater cumulative returns but also has much smaller drawdowns over the period of evaluation. The Fig. 4(b) drawdown analysis shows that the proposed system reduced losses to 11.8% at the worst period of the market downturn (Q4 2022), compared to 24.3% with the traditional buy-and-hold-based drawdown-51 reduction in drawdown maximum capital loss. This protective ability is due to the fact the framework monitors real-time risk and is able to adjust its position size automatically in response to changes in volatility successfully tracing the learned initiatives to lower exposure in case volatility data crosses the thresholds.

Fig. 9 Limited to risk-return scatter analysis gives vivid visual support to the preponderance of suggested framework across the efficiency range. Placed in the best upper-left quadrant with best returns (21.4%) and some of the lowest volatilities (13.6%), the framework has a Sharpe of 1.57- or 35.3% better than single-agent PPO and 118% better than traditional mean-variance optimization. The closeness to the theoretical efficient frontier proves that the framework is effective and achieves optimal opportunities available in the market with risk boundaries being taken into consideration.

Cloud distributed training has two important benefits. First, parallel search of a variety of market situations can be used to convert information into a single direction, exploring the high-dimensional state-action space efficiently. There can be 20 cloud instances, which means that training episodes can be reduced 87% in favor of single-machine learning. Second, the idea of distributed training provides strong robustness to policy learning because it exposes the agents to various market circumstances at a time and this minimizes overfitting to certain patterns present in the past. Table 3 above indicates that the cloud-enabled training enhances an out of sample sharpe ratio of 1.31 to 1.57 as opposed to the multi agent training running in a single machine.

The introduction of the overall risk management into the RL reward functionality is crucial towards a realistic deployment. Whereas most academic RA models are only curve-optimal on the basis of returns, practical institutional investors operate with restrictive risk bans such as maximum drawdown size, industry concentration, and capital requirements. The clear representation of transaction costs, market implications, and risking punishments in our framework is making sure that the strategies generated are performable and legal. Ablution experiment (Table 5) indicates that the maximum drawdown increases to 19.4% when the risk penalties are removed, and it is clear that reward shaping is important.

B. Robustness Across Market Regimes

One of the advantages of the suggested system is supported performance under different market conditions. Similar to that of Table 6, the framework delivers positive risk-adjusted returns in low volatility, high volatility and crisis. This strength is based on three mechanisms:

First, the regime-specific specialization in the multi-agent architecture allows automatic adjustment to the varying state of affairs. Agents are best suited to different environments and the attention mechanism is perfectly at ease shifting weight to the best suited agent depending on the current market state.

Second, the learning pipeline enables the system to acquire new market evidences without forgetting the previous trends in a disastrous manner. This deals with a failure situation prevalent with fixed-policy systems which are left to become outdated in cases where the market dynamics change.

Third, clear risk limits and defensive positioning in times of high uncertainty leading to devastating losses are avoided. The drawdown penalties and Curve VaR constraints promote conservative positioning in situations where there is data ambiguity in the market as flights have decreased losses during the March 2023 banking crisis (Table 6).

C. Scalability and Practical Deployment

The cloud architecture has obvious scalability benefits to institutional deployment. The traditional methods of portfolio optimization have an exponential growth in complexity with increase in assets and therefore can only be applied to large portfolios. On the contrary, the proposed RL framework can be distributed efficiently by means of distributed computation and neural network generalization.

Latency of inference of 26 milliseconds per decision (Table 9) allows the use of real-time updates of the strategy appropriate to do rebalancing in a day or even ad-hoc adjustments should it be necessary. The deployment of the system as a cloud enables full compatibility with the current trading infrastructure via generic APIs without any special on-premise software.

Transaction cost analysis (Table 8) shows that the framework obtains efficient trading as of 180% annual turnover, which is significantly less than naive RL approaches. The dampening mechanism (Eq. 8) and explicit cost modeling of the reward function are effective in balancing between responsiveness to market signals and trading efficiency. Such a property is essential to the institutional investors operating large portfolios whose high turnover can be of great effect to the returns by incurring the transaction costs and market impact.

D. Comparison with Traditional and ML Approaches

The entire comparison in Table 3 shows some interesting trends. The conventional ones (mean-variance, risk parity, Black-Litterman) perform well, but cannot adapt to dynamic market conditions. Their optimization models are not adaptable, that they can only process new information by manual rebalancing of their structure and that they are unable to adapt to regime changes.

The learning algorithms such as LSTM portfolios show better scalability but are constrained by the supervised learning framework that needed to include clear predictions

of returns. The error in prediction builds up to subopimal portfolio decision. Conversely, RL directly maximizes this ultimate goal (risk-adjusted returns) by inter-acting with the environment and so does not require accurate predictions of returns.

Single-agent RL approaches (DQN, PPO, DDPG) are significantly more effective than the old ones but cannot address the problem of exploration-exploitation trade-off in high-dimensional continuous action spaces. Multi-agent architecture in this respect, our multi-agent architecture deals with this shortcoming by breaking down exploration among specialized agents that are directed at sections of the decision space.

These differences in performances can be visually represented intuitively with the help of the risk-return scatter plot (Fig. 9). Conventional practices are concentrated in the lower-right aspect with an elevation of volatility compared with returns resulting in Sharpe ratios that are less than 0.9. The RL algorithms with single agents perform well and exhibit Sharpe ratios of around 1.0-1.2 and the multi-agent framework presented in this study attains the frontier with 1.57 Sharpe ratio. The Sharpe ratios plots easily depict that every development of the type of traditional to single-agent RL to multi-agent cloud-based RL is not a minor increment in the risk-adjusted performance but a quantum leap.

When compared to the new literature (Table 10), it shows that our framework can be used to perform on a state-of-the-art among published methods. The 29.8% margin of the Sharpe ratio as compared to the optimal similar method (AlTerkawi et al.[17]) justifies the suitability of our combined design to integrate multi-agent coordination, cloud-based training and detailed risk management.

E. Attention Mechanism Insights

The visualization of the attention mechanism (Fig. 10) is also a very valuable experience in strategy selection learning. The meta-controller is able to determine the properties of market regimes and dynamically weight the agents. When the market is experiencing a constant bullish trend, momentum-based agents are given utmost consideration and they take advantage of the trending activity. When volatility becomes aggressive, defensive mean reversion agents will take focus as they capitalize during volatile periods.

This automatic regime identifying and strategy choosing feature is a major benefit over older multi-strategy systems which use the rule-based switching or manual interaction. The learned attention operates continually based on historic performance, doing so such that it would be able to find out intricate market trends that cannot be easily learned through attempted rules.

F. Limitations and Considerations

Various limitations are worth considering despite good empirical outcomes. To start with, the framework consumes a lot of computational power to learn in the beginning. Though cloud infrastructure reduces this challenge, it can be expensive to smaller institutions. Future studies into more efficient RL algorithms or transfer learning methods may be used to minimize the training.

Second, the historical backtesting history even though broad (ten years covering several market cycles) cannot be used as a surety in the future. Real-life performance may be affected by the development of market microstructure, alteration of regulations, or unprecedented economic circumstances. To deploy production, continued monitoring and periodical retraining is still vital.

Third, the existing application is on the U.S. equities and international ETFs. Expansion over to other classes of assets such as fixed income, commodities, currencies, and other alternative investments must take special care of the specific aspects of the assets such as carry, roll yield and liquidity constraints.

Fourth, the model presumes market frictionless implementation. Practically, institutional orders, especially in large organizations, are subject to market and execution risk. Although our transaction model of market impact involves the quadratic market impact terms, a more advanced execution modeling that involves order book modeling and optimal execution strategies can be used to enhance practical performance.

Lastly, there is still low interpretability of the model. In addition to the fact that the attention weights would give certain information about the choice of the strategy, the neural network policies are mostly black-box. The formation of ways of interpreting the policy and explanation of personal trading decisions would lead to a more credible image and ease in obtaining a regulatory permit.

G. Practical Implementation Considerations

There are various elements of implementation that practitioners should pay attention to before deploying. Quality and consistency of data is most important- the performance of the framework relies on the availability of quality and low latency market data feeds. Effective monitoring systems are expected to keep monitoring the key performance measures, risk exposures, and system health indicators, in real-time.

There should be a variety of levels of risk management systems, which encompass pre-trade checks that confirm position limit and regulatory restrictions, provide an intra-day monitoring that generates signals when risk indicators reach a specific level, and post-trade reviews that confirm that executed trades correspond to the desired changes in the portfolio. Standard protocols used in normal risk management infrastructure (e.g. FIX) allow integration to support deployment in an institutional environment.

Best execution, transaction reporting, and audit trails are also regulatory compliance requirements that require extensive logging and documentation systems. The decision-making procedure of the framework and the state observations, policy outputs and details of implementation should be documented to be evaluated by the regulation authorities as well as to be analyzed by the internal authorities.

Operational resilience mechanisms such as; failover systems, backup policies and manual override capabilities

means that it will keep on running in the event of a system failure. Although the RL policies prove to be highly effective, human control and interventional skills are crucial safety nets.

CONCLUSION

The paper introduces a new cloud-based multi-agent reinforcement learning system that can resolve the key issues of dynamic portfolio optimization in the field of distributed computation and intelligent agent coordination. The proposed system has high performance measures of 21.4% per annum returns with a Sharpe ratio of 1.57 which is a significant improvement of 35.3% compared with single-agent, and 118% compared with traditional mean-variance optimization. Attention-based meta-controller manages to coordinate specialized agents in dynamically reallocating portfolio considering real-time market regime identification achieving a greater reduction of maximum drawdown relative to buy-and-hold strategies by 51%. The framework can be tested in extreme market environments with strong performance being proven over prolonged periods in bull markets and bear markets with much lower volatility. The training setup is scalable due to parallel episode generation and asynchronous updates of its policy (applied in cloud) to achieve 87% faster convergence, as opposed to the traditional approaches. The directions of future research can be proposed to include the adding of other data types like sentiment analysis and macroeconomic indicators, the extension of the framework to the multi-asset classes such as cryptocurrencies and commodities, the application of a more advanced risk management strategy with dynamic stop-loss, and the creation of a stronger explainable AI methodology to increase transparency in the process of agents decision making. The presented efficiency of the approach of integrating multi-agent reinforcement learning with the cloud computing infrastructure creates a paradigm of achievement in the next generation autonomous trading systems that will be able to cope with the more complex and unstable financial markets.

APPENDIX A: NOMENCLATURE

Symbol	Description
s_t	State vector at time t
a_t	Action vector at time t
r_t	Reward at time t
$\mathcal{S}$	State space
$\mathcal{A}$	Action space
n	Number of assets
w_i	Portfolio weight for asset i
$p_{i,t}$	Price of asset i at time t
$r_{i,t}$	Return of asset i at time t
$v_{i,t}$	Volatility of asset i at time t
$\mu(s\|\theta^\mu)$	Actor network (policy)
$Q(s,a\|\theta^Q)$	Critic network (action-value function)
$V(s\|\phi)$	Value function network
θ^μ	Actor network parameters
θ^Q	Critic network parameters
ϕ	Value network parameters
γ	Discount factor
τ	Soft update coefficient
η	Learning rate
σ	Exploration noise
λ_i	Reward scaling parameters
Σ	Covariance matrix
$\alpha_k(s)$	Attention weight for agent k
K	Number of agents
M	Number of cloud instances
B	Batch size
T	Episode horizon
E	Number of episodes
CVaR	Conditional Value-at-Risk
MDD	Maximum Drawdown
DDPG	Deep Deterministic Policy Gradient
PPO	Proximal Policy Optimization
DQN	Deep Q-Network
RL	Reinforcement Learning
GAE	Generalized Advantage Estimation

REFERENCES

[1]. S. Haykin, Neural Networks and Learning Machines, 3rd ed. Upper Saddle River, NJ: Pearson, 2009.

[2]. Cao, L. Lei, Y. Liu, Z. Chen, S. Shi, B. Li, W. Xu, and Z.-X. Yang, "Skeleton information-driven reinforcement learning framework for robust and natural motion of quadruped robots," Symmetry, vol. 17, no. 11, p. 1787, Oct. 2025. DOI: 10.3390/sym17111787

[3]. Zhou, L. Dong, and Y. Wang, "Prediction and control of hovercraft cushion pressure based on deep reinforcement learning," J. Mar. Sci. Eng., vol. 13, no. 11, p. 2058, Oct. 2025. DOI: 10.3390/jmse13112058

[4]. S. Li, M. López-Benítez, E. G. Lim, F. Ma, M. Cao, L. Yu, and X. Qin, "Enabling cooperative autonomy in UUV clusters: A survey of robust state estimation and information fusion techniques," Drones, vol. 9, no. 11, p. 752, Oct. 2025. DOI: 10.3390/drones9110752

[5]. T. Dieguez and S. Gomes, "Bridging intention and action in sustainable university entrepreneurship: The role of motivation and institutional support," Adm. Sci., vol. 15, no. 11, p. 422, Oct. 2025. DOI: 10.3390/admsci15110422

[6]. Nguyen, O. Mayet, and S. Desai, "Operational and supply chain growth trends in basic apparel distribution centers: A comprehensive review," Logistics, vol. 9, no. 4, p. 154, Oct. 2025. DOI: 10.3390/logistics9040154

[7]. L. AlTerkawi and M. AlTarawneh, "Federated decision transformers for scalable reinforcement learning in smart city IoT systems," Future Internet, vol. 17, no. 11, p. 492, Oct. 2025. DOI: 10.3390/fi17110492

[8]. Nucci and G. Papadia, "Hybrid genetic algorithm and deep reinforcement learning framework for IoT-enabled healthcare equipment maintenance scheduling," Electronics, vol. 14, no. 21, p. 4160, Oct. 2025. DOI: 10.3390/electronics14214160

[9]. X. Hao, S. Wang, X. Liu, T. Wang, G. Qiu, and Z. Zeng, "Q-learning-based multi-strategy topology particle swarm optimization algorithm," Algorithms, vol. 18, no. 11, p. 672, Oct. 2025. DOI: 10.3390/a18110672

[10]. Fulgione, S. Palladino, L. Esposito, S. Sarfarazi, and M. Modano, "A multi-stage framework combining experimental testing, numerical calibration, and AI surrogates for composite panel characterization," Buildings, vol. 15, no. 21, p. 3900, Oct. 2025. DOI: 10.3390/buildings15213900

[11]. Ayyildiz, T. K. Karaca, M. Cari, B. Y. Kavus, and N. Aydin, "Smart risk assessment and adaptive control strategy selection for human-robot collaboration in Industry 5.0: An intelligent multi-criteria decision-making approach," Processes, vol. 13, no. 10, p. 3206, Oct. 2025. DOI: 10.3390/pr13103206

[12]. D. Wu, K. Li, and Y. Cheng, "Green investment and emission reduction in supply chains under dual-carbon regulation: A dynamic game perspective on coordination mechanisms and policy insights," Sustainability, vol. 17, no. 19, p. 8951, Oct. 2025. DOI: 10.3390/su17198951

[13]. X. Zou, H. Luo, and Y. Yu, "Research on supplier channel encroachment strategies considering retailer fairness concerns from a

low-carbon perspective," Sustainability, vol. 17, no. 19, p. 8750, Sep. 2025. DOI: 10.3390/su17198750

[14]. Zhang, J. Wei, R. Tang, Q. Hu, Y. Wang, L. Chang, X. Gan, and J. Pei, "Enhanced swarm-intelligence optimization of inverter placement for cable cost minimization in standardized photovoltaic power units," Energies, vol. 18, no. 19, p. 5111, Sep. 2025. DOI: 10.3390/en18195111

[15]. Y. Li, S. Yan, Y. Sun, L. Liu, Z. Zhang, and Y. Shuai, "Investment efficiency analysis and evaluation of power grids in China: A robust dynamic DEA approach incorporating time lag effects," Energies, vol. 18, no. 18, p. 4962, Sep. 2025. DOI: 10.3390/en18184962

[16]. Q. Cheng et al., "Optimizing ecosystem service patterns with dynamic Bayesian networks for sustainable land management under climate change: A case study in China's Sanjiangyuan region," Remote Sens., vol. 17, no. 19, p. 3357, Oct. 2025. DOI: 10.3390/rs17193357

[17]. Bouzguenda and A. Jarboui, "Navigating the green frontier: Dynamic risk and return transmission between clean energy ETFs and ESG indexes in emerging markets," J. Risk Financial Manag., vol. 18, no. 10, p. 557, Oct. 2025. DOI: 10.3390/jrfm18100557

[18]. R. I. Areola, A. A. Adebiyi, and K. Moloi, "Artificial intelligence for optimizing solar power systems with integrated storage: A critical review of techniques, challenges, and emerging trends," Electricity, vol. 6, no. 4, p. 60, Oct. 2025. DOI: 10.3390/electricity6040060

[19]. D. Montoya, L. F. Grisales-Noreña, and R. I. Bolaños, "Optimal planning and dynamic operation of thyristor-switched capacitors in distribution networks using the atan-sinc optimization algorithm with IPOPT refinement," Sci, vol. 7, no. 4, p. 143, Oct. 2025. DOI: 10.3390/sci7040143

[20]. Zhou, Z. Tang, Y. Luo, D. Zhou, and G. Jiang, "From 'policy-driven' to 'park clustering': Evolution and attribution of location selection for pollution-intensive industries in the Beijing–Tianjin–Hebei urban agglomeration," Land, vol. 14, no. 11, p. 2103, Oct. 2025. DOI: 10.3390/land14112103

[21]. X. Xing, Q. Wang, F. Meng, P. Liu, L. Huang, and W. Zhuo, "Assessing the land use-carbon storage nexus along G318: A coupled SD-PLUS-InVEST model approach for spatiotemporal coordination optimization," Land, vol. 14, no. 10, p. 2067, Oct. 2025. DOI: 10.3390/land14102067

[22]. Z. Zhan, J. Wu, P. Xia, and Y. Hu, "Toward low-carbon and cost-efficient prefabrication: Integrating structural equation modeling and system dynamics," Sustainability, vol. 17, no. 18, p. 8307, Sep. 2025. DOI: 10.3390/su17188307

[23]. H. Hadded, S. Dardouri, A. Yüksel, J. Sghaier, and M. Arıcı, "Enhancing energy efficiency and thermal comfort through integration of PCMs in passive design: An energetic, environmental, and economic (3E) analysis," Buildings, vol. 15, no. 18, p. 3319, Sep. 2025. DOI: 10.3390/buildings15183319

[24]. Coppola, L. Di Costanzo, and A. Marchetta, "Enhancing sustainable mobility: A comparative analysis of C-ITS and fundamental diagram-based traffic jam detection," Sustainability, vol. 17, no. 18, p. 8217, Sep. 2025. DOI: 10.3390/su17188217

[25]. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015. DOI: 10.1038/nature14236

[26]. D. Silver et al., "Mastering the game of Go with deep neural networks and tree search," Nature, vol. 529, no. 7587, pp. 484–489, 2016. DOI: 10.1038/nature16961

[27]. T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in Proc. Int. Conf. Learning Representations (ICLR), 2016.

[28]. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

[29]. Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," arXiv preprint arXiv:1706.10059, 2017.

[30]. Z. Zhang, S. Zohren, and S. Roberts, "Deep reinforcement learning for trading," J. Financial Data Sci., vol. 2, no. 2, pp. 25–40, 2020. DOI: 10.3905/jfds.2020.1.030

[31]. Buehler, L. Gonon, J. Teichmann, and B. Wood, "Deep hedging," Quantitative Finance, vol. 19, no. 8, pp. 1271–1291, 2019. DOI: 10.1080/14697688.2019.1571683

I. APPENDIX A: NOMENCLATURE

Symbol	Description
G_θ	Generator network with parameters θ
D_ϕ	Discriminator network with parameters ϕ
$\mathcal{L}_{GAN}$	Generative adversarial network loss function
$\mathcal{L}_{VAE}$	Variational autoencoder loss function
$\mathcal{L}_{DM}$	Diffusion model loss function
$q_\psi(z\|x)$	VAE encoder distribution
$p_\omega(x\|z)$	VAE decoder distribution
β_t	Noise schedule parameter at time t
σ_t	Realized volatility at time t
h_t, c_t	LSTM hidden and cell states
$\alpha_t, \beta_t, \gamma_t$	Fusion gating weights
$\mathcal{L}_{VaR}$	Value-at-Risk loss component
$\mathcal{L}_{Sharpe}$	Sharpe ratio loss component
GAN	Generative Adversarial Network
VAE	Variational Autoencoder
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
AWS	Amazon Web Services
GCP	Google Cloud Platform

How to cite this article:

Shahzad Anwar "Reinforcement Learning-Driven Dynamic Investment Strategy Optimization Using Cloud-Based Simulation Frameworks" International Journal of Engineering Works, Vol. 12, Issue 11, PP. 207-233, November 2025. https://doi.org/10.5281/zenodo.17662007.