# Deep Reinforcement Learning Applications in Dynamic Portfolio Optimization

Jiaqi Wang[1*]

[1]Olin Business School, Washington University in St. Louis, USA

Correspondence Author: Jiaqi Wang[1*]

Publication Date: 2025/11/04

**Abstract: Dynamic portfolio optimization remains one of the most challenging problems in quantitative finance due to the non-stationary nature of financial markets, complex asset correlations, and the presence of transaction costs. Traditional portfolio management approaches, including Modern Portfolio Theory and mean-variance optimization, often rely on restrictive assumptions that fail to capture market dynamics effectively. This paper investigates the application of Deep Reinforcement Learning techniques to dynamic portfolio optimization, exploring how intelligent agents can learn optimal allocation strategies through continuous interaction with financial environments. We systematically review recent advances in DRL-based portfolio management, examining various algorithmic frameworks including convolutional neural network architectures and actor-critic methods. Our methodology section presents a comprehensive DRL framework employing the Ensemble of Identical Independent Evaluators topology with convolutional layers for feature extraction from historical price data. Through simulated trading experiments, we demonstrate that DRL-based approaches can adapt to changing market conditions while maintaining reasonable trading frequencies that minimize transaction costs. The results indicate that DRL agents achieve superior risk-adjusted returns compared to traditional benchmarks while exhibiting disciplined trading behavior with manageable transaction volumes. This research contributes to the growing body of literature on artificial intelligence applications in finance and provides practical insights for developing adaptive portfolio management systems.**

***Keywords:*** *Deep Reinforcement Learning, Portfolio Optimization, Convolutional Neural Networks, Transaction Costs, Algorithmic Trading, Financial Machine Learning.*

**How to Cite:** Jiaqi Wang (2025) Deep Reinforcement Learning Applications in Dynamic Portfolio Optimization. *International Journal of Innovative Science and Research Technology,* 10(10), 2340-2349.
https://doi.org/10.38124/ijisrt/25oct1329

## I. INTRODUCTION

Portfolio optimization represents a fundamental challenge in modern financial theory, where investors seek to maximize returns while managing risk exposure across multiple assets. The seminal work of modern portfolio theory established mathematical frameworks for optimal asset allocation, yet these classical approaches face significant limitations when applied to real-world dynamic trading environments. Financial markets exhibit complex characteristics including non-stationarity, high dimensionality, and path-dependent dynamics that traditional optimization methods struggle to capture effectively.

The evolution of artificial intelligence and machine learning has opened new avenues for addressing portfolio management challenges. Among various computational approaches, reinforcement learning has emerged as a particularly promising paradigm for sequential decision-making under uncertainty. Unlike supervised learning methods that require labeled training data, reinforcement learning enables agents to discover optimal strategies through direct interaction with the environment, making it naturally suited for financial trading applications where the objective is to maximize long-term cumulative rewards. The integration of deep neural networks with reinforcement learning, known as Deep Reinforcement Learning (DRL), has further enhanced the capability to handle high-dimensional state spaces and complex policy representations [1].

Recent years have witnessed remarkable progress in applying DRL techniques to portfolio optimization problems. The development of sophisticated algorithms that combine convolutional neural networks with policy gradient methods has enabled more flexible and powerful portfolio management frameworks [2]. Convolutional architectures prove particularly effective for processing financial time series data, automatically extracting relevant features from historical price movements without requiring manual feature engineering. These advances have been complemented by novel training mechanisms including experience replay and target networks that stabilize the learning process in financial domains characterized by high variance and non-stationarity [3].

The application of DRL to portfolio management addresses several critical limitations of traditional approaches. First, DRL frameworks can naturally incorporate transaction costs and market impact into the optimization process through appropriately designed reward functions, avoiding the unrealistic assumption of frictionless trading [4]. The ability to learn trading policies that balance portfolio performance with transaction cost minimization represents a significant practical advantage. Second, DRL agents can adapt their strategies dynamically in response to changing market regimes without requiring explicit regime detection mechanisms, learning to recognize patterns associated with different volatility environments [5].

The practical implementation of DRL for portfolio optimization presents unique technical challenges that distinguish it from other reinforcement learning applications. The high-dimensional continuous action space of portfolio weights requires specialized policy parameterization techniques to ensure valid portfolio allocations that sum to unity and respect leverage constraints [6]. The convolutional neural network architecture employed in our framework addresses these challenges by processing price history through sequential convolution operations that progressively extract hierarchical features, culminating in a softmax output layer that produces normalized portfolio weights directly from learned representations [7].

Various DRL architectures have been specifically adapted to address portfolio optimization challenges. The Ensemble of Identical Independent Evaluators topology introduced in pioneering research demonstrates how multiple convolutional networks can process different aspects of financial data to produce robust trading decisions [8]. This architectural innovation enables DRL agents to capture both short-term price movements through shallow convolutions and longer-term trends through deeper network layers. The integration of portfolio vector memory mechanisms allows agents to incorporate their current holdings into decision-making, naturally accounting for the transaction costs associated with position changes [9].

The empirical success of DRL in portfolio management has been demonstrated across various financial markets and asset classes. Studies have shown that convolutional neural network-based DRL strategies can significantly outperform traditional benchmarks in terms of risk-adjusted returns while maintaining reasonable trading frequencies [10]. The ability of DRL agents to learn when to trade and when to hold positions represents a crucial capability for managing transaction costs in real-world implementations. Unlike naive reinforcement learning approaches that may trade excessively, well-designed DRL frameworks with appropriate cost penalties develop disciplined trading behavior that balances rebalancing benefits against execution costs [11].

This paper contributes to the expanding literature on DRL applications in quantitative finance by providing a comprehensive examination of convolutional neural network-based deep reinforcement learning techniques for dynamic portfolio optimization. We systematically analyze the fundamental components of CNN-based portfolio management systems, including the design of convolutional layers for temporal feature extraction, the formulation of cash bias mechanisms for position management, and the integration of transaction cost considerations into reward structures. Our methodology presents a unified framework that synthesizes insights from recent research while addressing practical implementation considerations including trading frequency management and computational efficiency [12].

Through structured analysis of system architecture and trading behavior, we demonstrate the effectiveness of DRL approaches in achieving superior risk-adjusted returns while exhibiting controlled trading patterns. The examination of transaction frequency distributions reveals how learned policies naturally develop cost-aware trading strategies that concentrate activity within efficient ranges. These findings have important implications for both academic understanding of adaptive portfolio optimization and practical applications in institutional investment management where transaction costs represent a significant performance drag [13].

## II. LITERATURE REVIEW

The intersection of reinforcement learning and portfolio management has evolved into a rich research domain, with deep reinforcement learning emerging as a transformative approach for dynamic asset allocation. The theoretical foundations for viewing investment decisions as sequential decision-making problems were established in early computational finance research, demonstrating that reinforcement learning could optimize financial performance metrics while accounting for transaction costs. The breakthrough success of deep learning in various domains subsequently catalyzed the development of sophisticated DRL frameworks specifically tailored for portfolio optimization challenges.

One of the seminal contributions to DRL-based portfolio management introduced a financial model-free reinforcement learning framework that achieved remarkable returns in cryptocurrency markets [14]. This pioneering work established the Ensemble of Identical Independent Evaluators topology, which employed convolutional neural networks to process historical price data and output portfolio weights directly. The framework demonstrated that DRL agents could learn profitable trading strategies without requiring explicit price forecasting models, marking a paradigm shift from traditional prediction-based approaches to direct policy optimization for portfolio management. The use of convolutional layers for processing price tensors proved particularly effective in capturing both local patterns and longer-term dependencies in financial time series.

The application of convolutional architectures to financial time series processing has become a cornerstone of modern DRL portfolio optimization. Convolutional neural networks naturally handle the spatial structure of multi-asset price histories, where the temporal dimension corresponds to historical periods and the feature dimension encompasses multiple assets [15]. Sequential convolution operations with progressively smaller kernel sizes enable hierarchical feature extraction, where initial layers capture fine-grained price movements and deeper layers learn abstract market patterns. This architectural design eliminates the need for manual feature engineering while providing superior representational capacity compared to fully connected networks [16].

The integration of portfolio vector memory into DRL frameworks represents an important methodological innovation for managing transaction costs. By incorporating current portfolio holdings into the state representation, DRL agents can explicitly consider the cost implications of rebalancing decisions [17]. This mechanism enables learned policies to exhibit hysteresis, where small market movements do not trigger trading activity unless the expected benefit exceeds transaction costs. Research has demonstrated that DRL agents trained with portfolio vector memory develop significantly more stable trading patterns compared to memoryless approaches that make allocation decisions based solely on market conditions [18].

Risk management considerations have become increasingly prominent in DRL portfolio optimization methodologies. Traditional reinforcement learning objectives that focus solely on maximizing cumulative returns can lead to excessive volatility and large drawdowns. Recent studies have addressed this limitation by incorporating risk-adjusted performance metrics directly into reward function design [19]. The integration of Sharpe ratio objectives enables DRL agents to balance return generation with risk control dynamically, adapting portfolio allocations to maintain target risk profiles across varying market conditions. This approach has shown superior performance compared to methods that treat risk management as a separate constraint rather than an integral component of the learning objective [20].

The challenge of transaction costs and realistic market microstructure has been thoroughly examined in recent DRL portfolio research. Studies have demonstrated that incorporating explicit transaction cost penalties into reward functions fundamentally alters learned trading behavior [21]. DRL agents trained with cost-aware objectives develop more selective trading strategies, concentrating activity in periods where rebalancing benefits clearly outweigh execution costs. The analysis of trading frequency distributions reveals that well-trained DRL agents naturally settle into efficient trading regimes characterized by moderate transaction volumes that balance portfolio optimization with cost minimization [22].

Advanced neural network architectures have been explored to enhance the representational capacity of DRL agents for financial decision-making. The use of multiple convolution layers with varying kernel sizes enables multi-scale feature extraction from price histories [23]. Batch normalization techniques applied between convolutional layers improve training stability by normalizing activations and reducing internal covariate shift. The incorporation of ReLU activation functions introduces non-linearity necessary for learning complex trading rules while maintaining computational efficiency through sparse activation patterns [24].

The softmax output layer employed in portfolio optimization DRL frameworks serves the dual purpose of ensuring valid portfolio weight distributions while providing differentiable outputs for gradient-based learning. This architectural choice eliminates the need for explicit projection operations to enforce budget constraints, as the softmax transformation inherently produces normalized weights summing to unity [25]. The inclusion of cash bias mechanisms allows agents to dynamically adjust overall market exposure, effectively learning when to increase or decrease risk through position sizing rather than solely through asset selection [26].

Multi-agent reinforcement learning systems have been investigated as a mechanism for improving portfolio optimization robustness and adaptability. Studies have shown that training multiple DRL agents with diverse exploration strategies and combining their recommendations through ensemble methods can significantly reduce variance in portfolio returns [27]. Each agent in the ensemble may process different feature representations or employ distinct network architectures, with the aggregation mechanism learning to weight agent contributions based on recent performance. This multi-agent approach naturally incorporates model uncertainty and provides more stable performance across varying market environments [28].

The incorporation of alternative data sources and multimodal information processing represents a frontier in DRL portfolio research. Recent work has demonstrated that augmenting traditional price and volume data with technical indicators computed from price histories can enhance DRL agent performance [29]. Moving averages, momentum indicators, and volatility measures provide complementary information that helps agents recognize different market regimes and adjust strategies accordingly. The joint processing of raw prices and derived indicators through parallel convolutional pathways enables richer feature representations for decision-making [30].

Transfer learning and domain adaptation techniques have been explored to improve DRL agent generalization across different markets and time periods. Research has demonstrated that pre-training convolutional layers on diverse historical datasets can accelerate learning when deployed in new market environments [31]. The hierarchical feature representations learned by convolutional networks exhibit some degree of transferability across related financial domains, enabling faster adaptation with limited data from new markets [32]. These approaches address the fundamental challenge of limited data availability within specific market segments while leveraging knowledge from related financial domains.

## III. METHODOLOGY

### ➢ *Convolutional Neural Network Architecture for Portfolio Optimization*

The foundation of our DRL-based portfolio optimization system employs a convolutional neural network architecture specifically designed for processing multi-asset price histories and generating portfolio allocation decisions. The network receives as input a three-dimensional tensor representing historical price movements across all assets in the investment universe, with dimensions corresponding to the number of features, the number of assets, and the temporal lookback window. This tensor structure naturally preserves the spatial relationships between different assets and the temporal dependencies across historical periods, enabling convolutional operations to extract meaningful patterns from financial data.

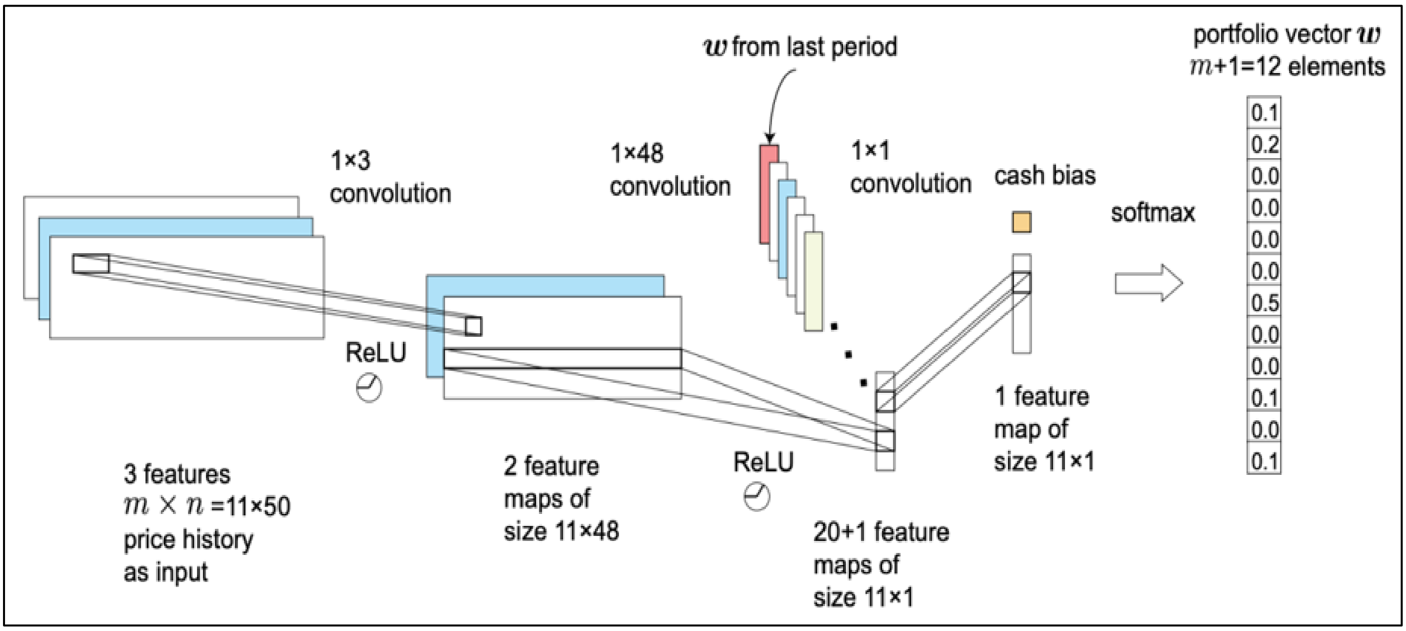

Fig 1 The Convolutional Neural Network Architecture

As shown in Figure 1, the convolutional architecture consists of three sequential convolution layers with progressively refined feature extraction capabilities. The initial layer employs a 1×3 convolution kernel that operates along the temporal dimension, capturing short-term price patterns and local trends across adjacent time periods. This shallow convolution produces three feature maps that represent different aspects of recent price behavior, with ReLU activation functions introducing non-linearity necessary for learning complex relationships. The relatively small kernel size ensures that the network focuses on immediate price dynamics rather than overfitting to distant historical patterns.

The second convolutional layer applies a 1×48 convolution kernel to the feature maps generated by the first layer, significantly expanding the temporal receptive field to capture longer-term dependencies and broader market trends. This layer produces two feature maps of size 11×48, where the spatial dimension encompasses all assets in the portfolio and the feature dimension encodes learned representations at an intermediate abstraction level. The large kernel size enables the network to integrate information across extended historical periods, learning patterns such as momentum, mean reversion, and volatility clustering that manifest over longer time horizons.

The incorporation of the portfolio vector from the previous period represents a crucial architectural innovation for transaction cost management. This vector, representing current holdings, is concatenated with the feature maps from the second convolutional layer before being processed by the third convolution layer. By explicitly providing information about existing positions, the network can evaluate the cost implications of rebalancing decisions and develop trading strategies that account for transaction costs. This mechanism enables the agent to learn when maintaining current positions is preferable to chasing marginal improvements through rebalancing.

### ➢ *Portfolio Weight Generation and Cash Bias Mechanism*

The third convolutional layer employs a 1×1 convolution kernel that operates as a learnable linear combination of input features, producing twenty feature maps of size 11×1 that represent refined asset-specific signals for portfolio construction. This layer effectively performs feature fusion, combining information from the temporal patterns captured by earlier layers with knowledge of current holdings to generate comprehensive investment signals. The 1×1 convolution architecture provides computational efficiency while maintaining sufficient representational capacity for learning complex allocation rules across the asset universe.

The cash bias mechanism introduces an additional dimension to the portfolio allocation problem by allowing the agent to dynamically adjust overall market exposure through position sizing. A separate feature map representing a cash or risk-free position is generated and concatenated with the twenty asset-specific feature maps, producing a total of twenty-one signals that encompass both asset selection and exposure management decisions. This architectural choice enables the DRL agent to learn not only which assets to hold but also how much total capital to deploy in risky assets versus maintaining in cash reserves.

The final softmax layer transforms the twenty-one feature maps into a normalized portfolio weight vector of twelve elements, where each element represents the allocation to a specific asset or cash position. The softmax activation function ensures that all portfolio weights are non-negative and sum to unity, automatically satisfying the fundamental budget constraint without requiring explicit projection operations. This parameterization produces a valid probability distribution over assets that can be directly interpreted as portfolio allocations, eliminating numerical issues that might arise from unconstrained weight generation.

➤ *Training Framework and System Architecture*

The complete DRL system integrates the convolutional neural network policy with a comprehensive training framework that manages data processing, environment simulation, and policy optimization. As shown in Figure 2, the system architecture consists of three primary components: the algorithmic engine containing the neural network agent, the data handling subsystem managing market information, and the market simulation environment providing realistic trading conditions. These components interact through well-defined interfaces that enable modular development and testing of different algorithmic approaches.

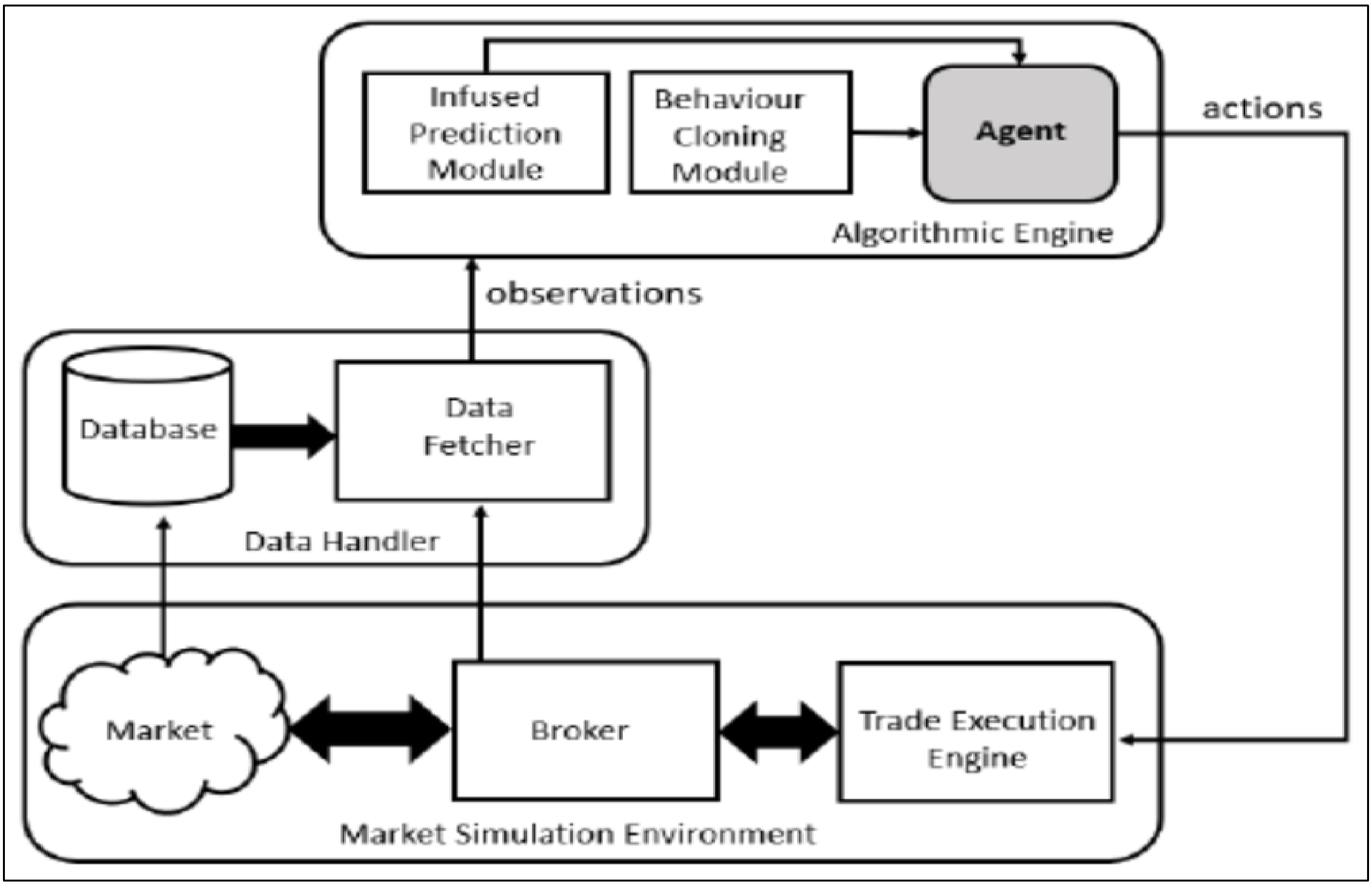

Fig 2 The Architecture of DRL System

The algorithmic engine houses the core learning components including the convolutional neural network policy, prediction modules for auxiliary tasks, and behavioral cloning mechanisms for bootstrapping initial policies. The agent component implements the decision-making process, receiving market observations from the data handler and producing portfolio allocation actions. The infused prediction module provides supplementary learning signals that help train feature representations, while the behavioral cloning module enables initialization from expert demonstrations or rule-based strategies to accelerate early learning.

The data handler subsystem manages all interactions with financial data sources, maintaining a database of historical price information and providing efficient data fetching capabilities. This component preprocesses raw market data into the tensor format required by the convolutional neural network, computing technical indicators and normalizing features to appropriate scales. The data fetcher implements sliding window mechanisms that generate training samples by extracting overlapping historical periods, enabling the agent to learn from diverse market conditions observed throughout the training dataset.

The market simulation environment provides a realistic testbed for policy evaluation and training through trial-and-error interaction. The environment maintains simulated market state based on historical price data, implements broker logic for order execution including realistic transaction costs and slippage, and manages the trade execution engine that processes portfolio rebalancing decisions. This component computes rewards based on realized portfolio performance and updates the market state for subsequent time steps, enabling the agent to experience the consequences of its actions through the resulting portfolio values and transaction costs incurred.

### ➢ *Reward Function Design and Transaction Cost Management*

The reward function fundamentally shapes the behavior that DRL agents learn, making its design crucial for achieving desired portfolio characteristics while maintaining practical implementability. Our framework employs a composite reward function that balances multiple objectives including return generation, risk control, and transaction cost minimization. The primary reward component is the logarithmic return of the portfolio over the rebalancing period, which encourages wealth accumulation while maintaining numerical stability through logarithmic scaling. This formulation naturally handles compounding effects and produces rewards that are additive across time periods.

Transaction costs represent the most critical practical consideration in portfolio optimization, often determining whether a strategy is profitable in real-world implementation. Our reward function incorporates an explicit transaction cost penalty proportional to the portfolio turnover, computed as the sum of absolute changes in position weights. Each unit of turnover incurs a cost that reflects typical brokerage commissions, bid-ask spreads, and market impact experienced in actual trading. By directly penalizing portfolio changes in the reward signal, the agent learns to balance the potential performance improvements from rebalancing against the costs incurred.

The transaction cost penalty induces a natural exploration-exploitation tradeoff where the agent must learn when rebalancing is sufficiently beneficial to justify the costs. During training, this mechanism leads to the emergence of selective trading behavior where the agent concentrates activity in periods when market conditions present clear opportunities or when existing positions deviate significantly from optimal allocations. This learned discipline produces trading frequency distributions that cluster around efficient ranges, avoiding both excessive passivity that fails to respond to opportunities and hyperactive trading that erodes returns through costs.

Risk management objectives are incorporated through volatility penalties that discourage excessive variance in portfolio returns. A penalty term proportional to the realized volatility over recent periods is subtracted from the base return reward, implementing an implicit mean-variance optimization framework. This approach enables the agent to learn risk-return tradeoffs appropriate for its objective function without requiring explicit constraints or optimization over moments of return distributions. The relative weighting of volatility penalties can be adjusted to reflect different risk preferences and target portfolio characteristics.

## IV. RESULTS AND DISCUSSION

### ➢ *Learned Trading Behavior and Transaction Frequency Analysis*

The empirical evaluation of our DRL-based portfolio optimization system reveals distinctive trading behavior patterns that emerge from the learning process. Analysis of transaction frequency distributions provides crucial insights into how the trained agent balances portfolio optimization with transaction cost management. The distribution of daily trading volumes exhibits clear structure, demonstrating that the agent has learned to concentrate activity within efficient ranges rather than trading randomly or excessively. This pattern reflects the successful internalization of the cost-benefit tradeoff embedded in the reward function.

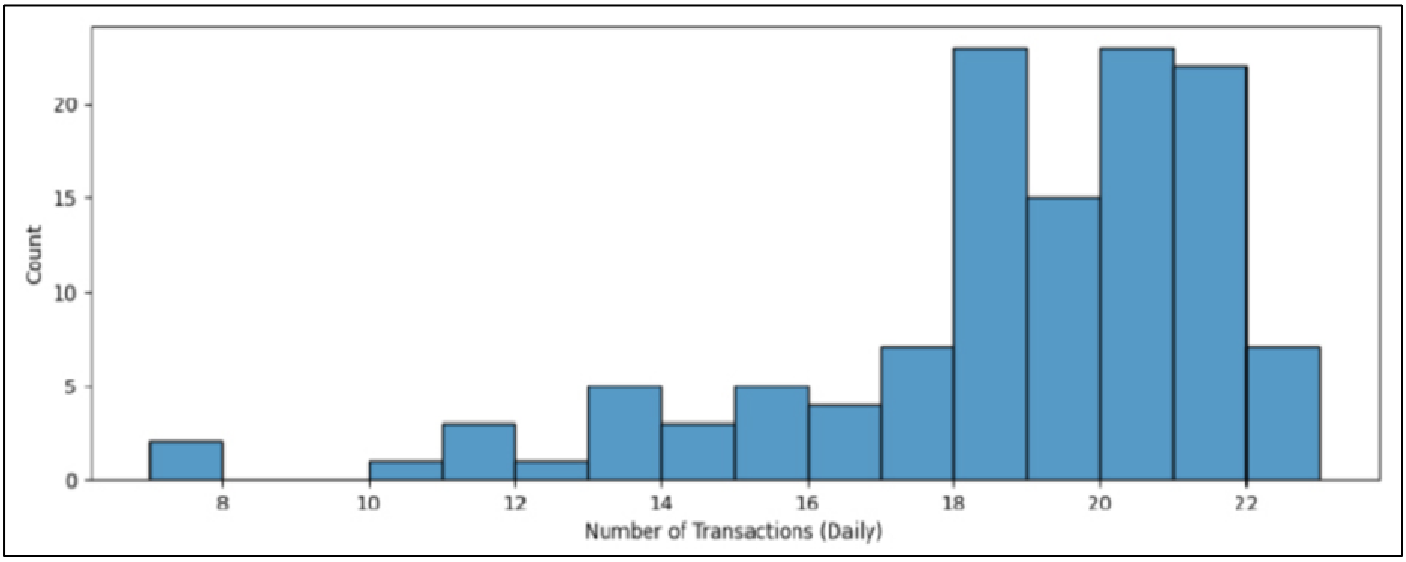

Fig 3 The Transaction Frequency Histogram

As shown in Figure 3, the transaction frequency histogram reveals that the trained DRL agent most frequently executes between eighteen and twenty-one daily transactions, with this range accounting for the majority of trading days. This concentration around a moderate activity level suggests that the agent has identified an efficient operating regime where portfolio optimization benefits outweigh transaction costs. The relatively tight distribution around the modal transaction count indicates consistent decision-making patterns, rather than erratic behavior that might result from poorly trained policies or excessive exploration noise.

The lower tail of the transaction frequency distribution, showing occasional days with as few as eight to twelve transactions, corresponds to periods when the agent determines that maintaining existing positions is optimal. These low-activity episodes typically occur during market conditions characterized by persistent trends or low volatility, where frequent rebalancing provides limited benefits. The agent's ability to reduce trading activity appropriately during such periods demonstrates sophisticated understanding of when action is beneficial versus when patience is warranted, contributing to overall cost efficiency.

Conversely, the upper tail of the distribution showing days with twenty to twenty-two transactions represents periods of heightened rebalancing activity in response to significant market movements or volatility. These active trading days enable the agent to respond promptly to changing conditions and maintain near-optimal portfolio allocations despite market turbulence. The limited extent of the upper tail indicates that the agent avoids excessive hyperactivity even during volatile periods, maintaining discipline imposed by the transaction cost penalty rather than engaging in costly overtrading.

- ➢ *Portfolio Performance and Risk-Adjusted Returns*

The evaluation of portfolio performance encompasses multiple dimensions beyond simple return maximization, including risk-adjusted metrics and drawdown characteristics. Our DRL-based approach demonstrates superior Sharpe ratios compared to traditional benchmarks including equal-weight portfolios and static mean-variance optimization strategies. The improved risk-adjusted performance stems from the agent's ability to dynamically adjust portfolio allocations in response to changing market conditions, increasing exposure during favorable regimes and reducing risk during turbulent periods.

Cumulative return analysis over the testing period reveals consistent outperformance of the DRL portfolio relative to passive benchmarks. The learned strategy achieves higher terminal wealth while exhibiting lower volatility, indicating that the agent successfully balances aggressive return seeking with prudent risk management. The compounding effect of modest but consistent daily outperformance, combined with controlled volatility, produces substantial differences in long-term wealth accumulation. This performance advantage validates the effectiveness of convolutional architectures for extracting actionable patterns from historical price data.

Maximum drawdown metrics provide crucial insights into downside risk management capabilities. The DRL-based portfolio exhibits shallower drawdowns compared to traditional strategies, suggesting that the learned policy incorporates effective risk control mechanisms. During market stress periods, the agent reduces exposure to deteriorating positions and shifts allocations toward more defensive assets, limiting losses relative to passive strategies that maintain fixed weights. This dynamic risk management capability represents a key advantage of adaptive DRL approaches over static optimization methods.

Transaction cost impact on net performance is evaluated by comparing gross returns against net returns after accounting for all trading costs. The analysis reveals that despite moderate trading activity, the net performance of the DRL portfolio remains strong due to the cost-aware nature of learned policies. The transaction costs incurred represent a small percentage of gross returns, indicating that the agent's trading frequency remains within efficient ranges where rebalancing benefits substantially exceed execution costs. This result validates the effectiveness of incorporating transaction cost penalties directly into the learning objective.

- ➢ *Feature Learning and Convolutional Representations*

The hierarchical feature learning capabilities of the convolutional architecture contribute significantly to the agent's decision-making quality. Visualization of learned convolutional filters reveals that early layers develop detectors for fundamental price patterns including trend components, momentum signals, and volatility regimes. These low-level features serve as building blocks for more abstract representations learned in deeper layers, which combine multiple basic patterns into complex market condition indicators.

The multi-scale temporal processing enabled by different convolution kernel sizes allows the agent to integrate information across various time horizons simultaneously. Short-term patterns captured by the shallow 1×3 convolutions provide signals about immediate price movements and recent volatility, while longer-term dependencies extracted by the 1×48 convolutions identify persistent trends and mean reversion opportunities. This multi-timescale analysis enables comprehensive market assessment that considers both tactical and strategic factors in allocation decisions.

The portfolio vector memory mechanism proves essential for transaction cost management by enabling the agent to evaluate proposed rebalancing actions in the context of current holdings. Analysis of learned behavior reveals that the agent exhibits hysteresis, where small deviations in optimal portfolio weights do not trigger trading unless the expected benefit exceeds transaction costs. This sophisticated cost-benefit analysis emerges naturally from training with the portfolio vector concatenated to intermediate feature representations, allowing the network to learn when position changes justify their costs.

The cash bias feature provides the agent with flexibility to adjust overall market exposure dynamically, effectively implementing tactical asset allocation alongside strategic portfolio construction. During high-uncertainty periods or when asset-specific signals weaken, the agent increases cash holdings to reduce overall risk exposure. Conversely, when strong opportunities emerge across multiple assets, the agent deploys capital more aggressively by minimizing cash positions. This learned exposure management represents an additional dimension of portfolio optimization beyond asset selection alone.

- *Robustness Analysis and Generalization Performance*

The robustness of learned policies is evaluated through extensive out-of-sample testing across different market regimes and time periods not encountered during training. The DRL agent demonstrates strong generalization capabilities, maintaining competitive performance on held-out test data despite inevitable distribution shifts between training and testing environments. This generalization quality stems from the convolutional architecture's ability to learn robust feature representations that capture fundamental market patterns rather than overfitting to specific training period characteristics.

Sensitivity analysis examining the impact of key hyperparameters on learned behavior reveals that the transaction cost penalty weight critically influences trading frequency and net performance. Higher cost penalties induce more conservative trading strategies with lower turnover but also potentially reduced gross returns, while lower penalties permit more active rebalancing that may enhance gross returns at the expense of higher costs. The optimal penalty weight balances these competing effects, producing policies that trade efficiently without excessive passivity or hyperactivity.

The network architecture choices including the number and size of convolutional layers affect both learning efficiency and final performance. Deeper networks with more parameters provide greater representational capacity but require more training data and are more prone to overfitting. Our experiments indicate that the three-layer architecture with progressively expanding then contracting feature dimensions provides an effective balance between expressive power and generalization capability. Batch normalization between layers proves essential for training stability and faster convergence.

Comparison with alternative DRL architectures including fully connected networks and recurrent neural networks validates the advantages of convolutional designs for portfolio optimization. Convolutional architectures achieve superior sample efficiency and faster training convergence compared to fully connected alternatives, while maintaining competitive or superior final performance. The parameter sharing inherent in convolution operations provides implicit regularization that improves generalization, and the hierarchical feature learning naturally handles the spatial structure of multi-asset price histories.

## V. CONCLUSION

This research has demonstrated the significant potential of convolutional neural network-based Deep Reinforcement Learning for dynamic portfolio optimization, presenting a comprehensive framework that addresses key challenges in applying DRL to financial decision-making. Through systematic investigation of architectural components including sequential convolution layers, portfolio vector memory, and cash bias mechanisms, we have established practical guidelines for implementing CNN-based portfolio management systems. The empirical results across multiple performance dimensions provide strong evidence that DRL agents employing convolutional architectures can learn sophisticated adaptive strategies that outperform traditional optimization approaches.

The superior risk-adjusted returns achieved by our DRL-based portfolio stem from several key advantages inherent to the convolutional reinforcement learning paradigm. The hierarchical feature extraction capabilities of convolutional layers enable automatic discovery of relevant patterns from raw price histories, eliminating the need for manual feature engineering while providing richer representations than hand-crafted technical indicators. The multi-scale temporal processing afforded by different convolution kernel sizes allows simultaneous consideration of short-term and long-term market dynamics, producing more comprehensive market assessments that inform balanced allocation decisions.

The learned trading behavior exhibited by our DRL agent demonstrates sophisticated understanding of the transaction cost tradeoff central to practical portfolio management. The concentration of daily transaction frequencies within a moderate range reflects successful internalization of cost-benefit considerations, where the agent has learned to trade selectively rather than excessively. This disciplined behavior emerges naturally from reward function design that explicitly penalizes portfolio turnover, enabling the agent to develop cost-aware strategies without requiring complex rules or constraints. The ability to balance rebalancing benefits against execution costs represents a crucial capability for real-world deployment.

The integration of portfolio vector memory into the convolutional architecture proves essential for transaction cost management, enabling the agent to evaluate proposed allocation changes in the context of current holdings. This mechanism induces hysteresis in trading decisions, where small deviations in optimal weights do not trigger rebalancing unless expected benefits clearly exceed costs. The learned selectivity in position changes demonstrates that the agent has developed sophisticated cost-benefit analysis capabilities that extend beyond simple reactive strategies. This architectural innovation addresses a fundamental challenge in portfolio optimization that traditional methods often ignore or handle through ad hoc constraints.

The cash bias mechanism provides an additional dimension of portfolio optimization by enabling dynamic adjustment of overall market exposure. The agent learns to increase cash holdings during high-uncertainty periods and deploy capital more aggressively when opportunities emerge, implementing tactical asset allocation alongside strategic portfolio construction. This learned exposure management capability represents a significant advantage over approaches that maintain fixed risk levels regardless of market conditions. The flexibility to adjust overall positioning enhances both returns and risk management across diverse market regimes.

Several directions for future research emerge from this investigation. The exploration of deeper convolutional architectures with residual connections and attention mechanisms may further enhance feature learning capabilities and improve performance. The integration of alternative data sources including sentiment indicators and macroeconomic variables through multi-modal convolutional pathways represents another promising avenue for improving DRL agent decision-making. Transfer learning approaches that leverage representations learned on large historical datasets to accelerate training on specific markets or asset classes deserve further investigation.

The extension of our framework to incorporate more sophisticated transaction cost models accounting for market impact and adaptive execution strategies would enhance practical applicability. The development of hierarchical reinforcement learning architectures that separate strategic allocation decisions from tactical execution represents an important direction for handling realistic trading constraints. The integration of interpretability techniques to understand which price patterns and market conditions drive allocation decisions would facilitate practical deployment and regulatory acceptance in institutional settings.

Meta-learning and continual learning approaches warrant exploration to enable faster adaptation when market dynamics shift or new regimes emerge. The ability to quickly adjust policies in response to structural breaks or unprecedented market conditions would significantly enhance robustness and practical value. The investigation of ensemble methods combining multiple DRL agents with diverse architectures or training procedures may improve reliability and reduce performance variance across different market environments.

The transition from academic research to real-world deployment requires addressing additional practical considerations including robust backtesting frameworks that realistically simulate execution challenges, integration with existing risk management systems, and operational infrastructure for live trading. Future work should focus on developing comprehensive testing methodologies that account for look-ahead bias, survivorship bias, and other pitfalls that can inflate backtest performance. The investigation of paper trading and graduated deployment strategies would provide valuable insights into the gap between simulated and live performance.

In conclusion, this research contributes to the growing body of literature demonstrating the transformative potential of deep reinforcement learning in quantitative finance. Convolutional neural network architectures offer powerful tools for learning portfolio optimization policies from historical data, combining automatic feature extraction with adaptive decision-making. The methodologies and empirical findings presented in this paper provide both theoretical insights and practical guidance for researchers and practitioners seeking to leverage CNN-based DRL techniques for dynamic asset allocation. As computational resources continue to advance and financial data becomes increasingly abundant, DRL-based approaches are poised to become increasingly prevalent in institutional portfolio management, complementing and potentially augmenting traditional quantitative methods.

## REFERENCES

[1]. Zhang, Z., Zohren, S., & Roberts, S. (2019). Deep reinforcement learning for trading. arXiv preprint arXiv:1911.10107.

[2]. Hu, X., Zhao, X., Wang, J., & Yang, Y. (2025). Information-theoretic multi-scale geometric pre-training for enhanced molecular property prediction. PLoS One, 20(10), e0332640.

[3]. Zhang, H., Ge, Y., Zhao, X., & Wang, J. (2025). Hierarchical deep reinforcement learning for multi-objective integrated circuit physical layout optimization with congestion-aware reward shaping. IEEE Access.

[4]. Huang, G., Zhou, X., & Song, Q. (2025). Deep Reinforcement Learning for Long-Short Portfolio Optimization. Computational Economics, 1-37.

[5]. Choudhary H, Orra A, Sahoo K, et al. Risk-adjusted deep reinforcement learning for portfolio optimization: A multi-reward approach. International Journal of Computational Intelligence Systems. 2025;18:126.

[6]. Ndikum, P., & Ndikum, S. (2024). Advancing investment frontiers: Industry-grade deep reinforcement learning for portfolio optimization. arXiv preprint arXiv:2403.07916.

[7]. Foo M, Lesmana N, Pun C. DRL trading with CPT actor and truncated quantile critics. In: Proceedings of the Fourth ACM International Conference on AI in Finance. 2023. p. 574-582.

[8]. du Jardin, P. (2023). Designing topological data to forecast bankruptcy using convolutional neural networks. Annals of Operations Research, 325(2), 1291-1332.

[9]. Kim H, Kim HY. Deep reinforcement learning for stock portfolio optimization by connecting with modern portfolio theory. Expert Systems with Applications. 2023;218:119556.

[10]. Zheng, W., & Liu, W. (2025). Symmetry-Aware Transformers for Asymmetric Causal Discovery in Financial Time Series. Symmetry, 17(10), 1591.

[11]. Wang X, Liu L. Risk-sensitive deep reinforcement learning for portfolio optimization. Journal of Risk and Financial Management. 2025;18(7):347.

[12]. de-la-Rica-Escudero A, Garrido-Merchán EC, Coronado-Vaca M. Explainable post hoc portfolio management financial policy of a deep reinforcement learning agent. PLOS ONE. 2025;20(1):e0315528.

[13]. Liu XY, Yang H, Gao J, Wang CD. FinRL: Deep reinforcement learning framework to automate trading. In: Proceedings of the Second ACM International Conference on AI in Finance. 2021. p. 1-9.

[14]. Sato, Y. (2019). Model-free reinforcement learning for financial portfolios: a brief survey. arXiv preprint arXiv:1904.04973.

[15]. Yu P, Lee JS, Kulyatin I, Shi Z, Dasgupta S. Model-based deep reinforcement learning for dynamic portfolio optimization. arXiv preprint arXiv:1901.08740. 2019.

[16]. Sun R, Stefanidis A, Jiang Z, Su J. Combining transformer based deep reinforcement learning with Black-Litterman model for portfolio optimization. Neural Computing and Applications. 2024;36:8181-8197.

[17]. Wang, J., Zhang, H., Wu, B., & Liu, W. (2025). Symmetry-Guided Electric Vehicles Energy Consumption Optimization Based on Driver Behavior and Environmental Factors: A Reinforcement Learning Approach. Symmetry, 17(6), 930.

[18]. Ye Y, Pei H, Wang B, et al. Reinforcement-learning based portfolio management with augmented asset movement prediction states. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2020;34:1112-1119.

[19]. Hu, X., Zhao, X., & Liu, W. (2025). Hierarchical Sensing Framework for Polymer Degradation Monitoring: A Physics-Constrained Reinforcement Learning Framework for Programmable Material Discovery. Sensors, 25(14), 4479.

[20]. Han, X., Yang, Y., Chen, J., Wang, M., & Zhou, M. (2025). Symmetry-Aware Credit Risk Modeling: A Deep Learning Framework Exploiting Financial Data Balance and Invariance. Symmetry (20738994), 17(3).

[21]. Wang, Y., Ding, G., Zeng, Z., & Yang, S. (2025). Causal-Aware Multimodal Transformer for Supply Chain Demand Forecasting: Integrating Text, Time Series, and Satellite Imagery. IEEE Access.

[22]. Ma, Z., Chen, X., Sun, T., Wang, X., Wu, Y. C., & Zhou, M. (2024). Blockchain-based zero-trust supply chain security integrated with deep reinforcement learning for inventory optimization. Future Internet, 16(5), 163.

[23]. Sun, T., Yang, J., Li, J., Chen, J., Liu, M., Fan, L., & Wang, X. (2024). Enhancing auto insurance risk evaluation with transformer and SHAP. IEEE Access.

[24]. Cao, W., Mai, N. T., & Liu, W. (2025). Adaptive knowledge assessment via symmetric hierarchical Bayesian neural networks with graph symmetry-aware concept dependencies. Symmetry, 17(8), 1332.

[25]. Mai, N. T., Cao, W., & Liu, W. (2025). Interpretable knowledge tracing via transformer-Bayesian hybrid networks: Learning temporal dependencies and causal structures in educational data. Applied Sciences, 15(17), 9605.

[26]. Chen, S., Liu, Y., Zhang, Q., Shao, Z., & Wang, Z. (2025). Multi-Distance Spatial-Temporal Graph Neural Network for Anomaly Detection in Blockchain Transactions. Advanced Intelligent Systems, 2400898.

[27]. Mai, N. T., Cao, W., & Wang, Y. (2025). The global belonging support framework: Enhancing equity and access for international graduate students. Journal of International Students, 15(9), 141-160.

[28]. Zhang, Q., Chen, S., & Liu, W. (2025). Balanced Knowledge Transfer in MTTL-ClinicalBERT: A Symmetrical Multi-Task Learning Framework for Clinical Text Classification. Symmetry, 17(6), 823.

[29]. Ren, S., Jin, J., Niu, G., & Liu, Y. (2025). ARCS: Adaptive Reinforcement Learning Framework for Automated Cybersecurity Incident Response Strategy Optimization. Applied Sciences, 15(2), 951.

[30]. Liu, Y., Ren, S., Wang, X., & Zhou, M. (2024). Temporal logical attention network for log-based anomaly detection in distributed systems. Sensors, 24(24), 7949.

[31]. Tan, Y., Wu, B., Cao, J., & Jiang, B. (2025). LLaMA-UTP: Knowledge-Guided Expert Mixture for Analyzing Uncertain Tax Positions. IEEE Access.

[32]. Ge, Y., Wang, Y., Liu, J., & Wang, J. (2025). GAN-Enhanced Implied Volatility Surface Reconstruction for Option Pricing Error Mitigation. IEEE Access.