Reinforcement Learning in Portfolio Management: A Survey of Methods and Trends

Silan Hu*†, Yulin Huang*†, Arjun Agarwal†, Tanya Warrier†, Wang Yuwen†, Haozhe Ma†‡ Zhengding Luo†§
† National University of Singapore, Singapore
§Nanyang Technological University, Singapore
silan.hu@u.nus.edu, huang_yulin@nus.edu.sg
E1503331@u.nus.edu, E1503355@u.nus.edu, E0492463@u.nus.edu, haozhe.ma@u.nus.edu, luoz0021@e.ntu.edu.sg
* The authors contributed equally to this work.
‡ Corresponding author.

***Abstract*—This paper presents a comprehensive survey of reinforcement learning (RL) algorithms applied to stock trading and portfolio management. We review the foundational contributions that introduced RL to financial decision-making, as well as more recent work incorporating deep learning architectures, margin constraints, and domain-specific heuristics. Our goal is to provide a structured overview of the field's evolution, highlight key methodological advances, and outline areas where current approaches face limitations. The survey serves as a reference for future work at the intersection of RL and financial systems.**

I. Introduction

Portfolio management (PM) is a fundamental activity in finance. It involves selecting and managing financial assets such as stocks, bonds, or cryptocurrencies, with the aim to maximize returns while minimizing risk according to investor preferences and constraints [1], [2]. High risk-adjusted returns, quantified by metrics such as the Sharpe ratio [3], represent a central goal. However, performing well consistently under changing market conditions remains a significant challenge [2]. Decisions in stock trading and portfolio allocation are sequential, made under uncertainty, with incomplete information and complex market dynamics.

Traditional PM approaches are based on Modern Portfolio Theory (MPT), introduced by Markowitz in 1952. MPT offers a framework for constructing portfolios by optimizing the return-risk trade-off, forming the efficient frontier [4]. Methods like Mean-Variance Optimization (MVO) are widely used [2]. However, these methods require accurate forecasts of future returns and covariances, which are difficult to obtain [5].

Reinforcement Learning (RL) offers an alternative by framing PM as sequential decision-making under uncertainty. Within the RL framework, an agent learns an optimal investment policy π through interaction with a market environment. At each step t, the agent observes state s_t (market conditions, portfolio status), takes action a_t (determines portfolio weights), receives reward r_t, and transitions to state s_{t+1} [6]. The agent aims to maximize the expected cumulative reward (following the Maximum Expected Utility). This approach allows learning adaptive strategies that optimize for objectives like risk-adjusted return without requiring explicit market forecasts [1].

Deep Reinforcement Learning (DRL) combines Deep Learning with RL with notable success. DRL enables handling high-dimensional state spaces, typical in financial markets, efficiently and learning non-linear policies through neural networks [7]. Early research used algorithms like Q-learning [8], Recurrent RL [9], and deep direct reinforcement learning [10]. Recent work employs advanced algorithms including Deep Q-Networks (DQN) [11], Deep Deterministic Policy Gradient (DDPG) [1], Proximal Policy Optimization (PPO) [2], [4], [12], [13], and Advantage Actor-Critic (A2C) [12], often combined with architectures like LSTMs [13], CNNs [1], Attention mechanisms [14], reward shaping[15]–[18], hierarchical policies[19]–[21], and Graph Convolutional Networks [22] to better model financial data.

A. Motivation and Scope

Financial markets exhibit challenging characteristics such as non-stationarity, partial observability, noisy data, heavy-tailed return distributions, and complex asset interactions. These traits make them a difficult testing ground. However, the availability of market replay for backtesting—combined with RL's adaptive learning capabilities—makes Reinforcement Learning a promising approach for handling these challenges.

This survey reviews Deep Reinforcement Learning (DRL) applications for stock portfolio management. We synthesize methodological advancements from key papers, identify common practices, and highlight limitations and opportunities in this field. We aim to bridge RL research and financial applications by providing a structured overview of how DRL techniques are adapted and evaluated for investment portfolios.

B. Structure of Our Survey

We analyze these works focusing on: (1) DRL algorithms and strategies, (2) network architectures, (3) evaluation methodologies, and (4) reported performance, comparisons, challenges, and future directions. The survey is structured thematically to provide a coherent overview of the field's progress.

II. Overview of RL in Portfolio Management

Reinforcement learning for portfolio management trains an agent to make sequential investment decisions across assets. The agent learns a policy π mapping market states s_t to

979-8-3315-7206-8/25/$31.00

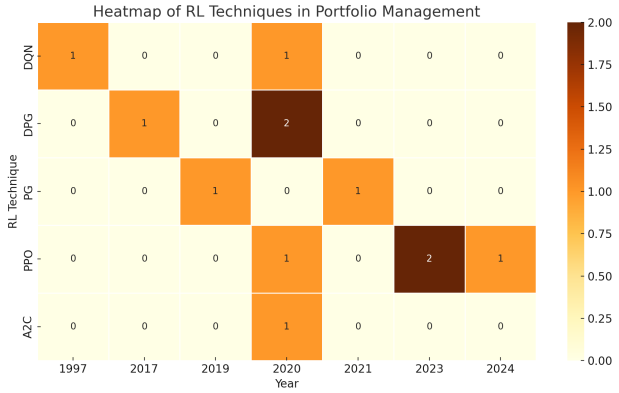

Fig. 1. Heatmap showing the temporal distribution of reinforcement learning techniques applied in PM studies.

TABLE I
OVERVIEW OF RL-BASED PORTFOLIO MANAGEMENT STUDIES

Author Reference, Year	RL	Type	Dataset
Neuneier, R., 1997	DQN	D	Germany Equity
Jiang, Z. et al., 2017	DPG	C	Cryptocurrency
Wang, J. et al., 2019	PG	D	US, China Equities
Ye, Y. et al., 2020	DPG	C	Bitcoin, High-Tech Stocks
Lucarelli & Borrotti, 2020	DQN, DDQN, DDDQN	D	Cryptocurrency
Yang, H. et al., 2020	A2C, PPO, DDPG	C	US Equity (Dow Jones)
Wang, Z. et al., 2021	PG	H	US, China, HK Equity
Zou, J. et al., 2023	PPO	C	US, China, India, UK
Sood, S. et al., 2023	PPO	C	US Equities
Acero, F. et al., 2024	PPO	C	US Equities (ESG stocks)

[c] Continuous, [d] Discrete, [h] Hybrid

actions a_t (portfolio allocations), maximizing a cumulative reward function related to risk-adjusted returns. This approach contrasts with traditional methods relying on single-period optimization or explicit market predictions.

RL methods applied to PM can be categorized:

- **Model-Free RL:** Learns policies or value functions directly from experience, without modeling market dynamics. This is the dominant approach due to market complexity.
  - *Value-Based Methods:* Estimate state or state-action values (Q-Learning, DQN, SARSA).
  - *Policy-Based Methods:* Directly optimize the policy (REINFORCE, PG, TRPO, PPO). Suitable for continuous actions.
  - *Actor-Critic Methods:* Combine policy and value estimation (A2C, A3C, DDPG, SAC). Used for stability and continuous actions.
- **Model-Based RL:** Learns an environment model for planning or generating simulated data. Less common but potentially sample-efficient.
- **Multi-Agent RL (MARL):** Models interactions between multiple agents.
- **Other Approaches:** Includes specialized methods like meta-learning, hierarchical RL, neuroevolution, etc.

To further illustrate the evolution and usage frequency of various RL techniques over time in Table I. Figure 1 presents a heatmap summarizing the adoption in a few portfolio management studies. This visualization reveals shifts in algorithmic preference—from an early interest in value-based methods like DQN to the recent dominance of PPO and actor-critic approaches.

III. RL ALGORITHMS FOR THE PM-MDP

This section examines how DRL algorithms address the specific challenges posed by the PM-MDP formulation.

A. Model-Free Approaches: Learning without Explicit Market Models

Model-free RL dominates PM research [3] primarily because accurately modeling complex, non-stationary market dynamics (P) is extraordinarily difficult. While these methods avoid potentially inaccurate market models, they face significant challenges, including sample inefficiency, hyperparameter sensitivity, and the need for substantial data—limitations rarely addressed thoroughly in the literature.

Value-Based Methods in Early Adaptations and Action Space Limitations: Value-based methods present fundamental compatibility issues with PM problems. Neuneier [8] adapted Q-learning by introducing an artificial deterministic transition step specifically to handle market stochasticity (P), which is mostly independent of the investor's actions. However, as mentioned in Lucarelli and Borrotti [11], these methods necessitate discretizing the inherently continuous action space (A) of portfolio weights. The discretization of action space is a critical limitation that not only hinders finding optimal fine-grained allocations, but also scales poorly as asset numbers increase [1].

These limitations have led to reduced use of value-based methods in recent portfolio management research, highlighting a significant gap between traditional RL techniques and the demands of financial applications.

Policy-Based Methods in Directly Optimizing Continuous Portfolio Policies: Policy-based methods directly learn policies $\pi_\theta(a|s)$, providing a natural framework for PM's continuous action space. The research progression shows increasingly sophisticated adaptations to financial complexities:

Jiang et al. [1] selected DPG [23] explicitly for its continuous control capabilities. Their EIIE architecture addressed the multi-asset state challenge (S) with shared parameters while incorporating Portfolio-Vector Memory to handle transaction costs (R). By feeding previous allocations w_{t-1} as input, the agent learned to implicitly penalize large, costly portfolio shifts—demonstrating an early structural adaptation to financial realities.

AlphaStock [14] applies attention-based networks to encode momentum trading signals and optimize the Sharpe Ratio (R) directly using gradient ascent. Each stock's historical state $r_t^{(i)}$ is represented via an LSTM-HA, and a Cross-Attention Attention Network (CAAN) enables cross-asset feature comparison, generating a winner score $s_t^{(i)}$. Stocks are ranked by this score: the top G is longed, the bottom G is shorted, and

softmax is optionally used for capital allocation. To enhance interpretability, AlphaStock performs a sensitivity analysis by computing partial derivatives of the winner score with respect to each input and feature. These influences are averaged across stocks and holding periods to assess the contribution of each input dimension to selection decisions.

DeepTrader [22] extends AlphaStock by modeling market regimes (P) and enhancing risk-return trade-offs (R). It decouples the asset scoring and market sentiment modules. In the asset scorer, a Temporal Convolutional Network (TCN) replaces LSTM for parallel temporal modeling, mitigating vanishing gradients. Extracted temporal features $\hat{H}^l$ feed into spatial attention [24] and GCN units, producing short-term (S^l) and long-term (Z^l) correlation weights, which are multiplied to yield final spatial weights. The market scorer processes macro sentiment via an LSTM with temporal attention, outputting μ, σ to sample short allocation $\rho \sim \mathcal{N}(\mu, \sigma^2)$.

The final trading policy π combines portfolio weights and short ratios. Optimization involves two objectives: - Asset scorer: $\nabla J^a(\theta) = \sum_{\tau\sim\pi_\theta} \sum_{t=0}^{|\tau|} \log(\mathbf{y}_t \nabla \pi_\theta^a)$ - Market scorer: $\nabla J^m(\theta_m) = \sum_{\tau\sim\pi_\theta} \sum_{t=0}^{|\tau|} R_t \nabla \log(\pi_\theta^m)$. The combined policy gradient is: $\nabla J(\theta) = \nabla J^a(\theta_a) + \iota \nabla J^m(\theta_m)$.

Gradient ascent updates $\theta \leftarrow \theta + \eta \nabla J(\theta)$. These works represent a critical trend toward embedding financial logic directly in RL frameworks rather than treating PM as a generic control problem.

PPO has gained prominence in recent PM literature [2], [4], [12], [13], chosen primarily for its empirical stability when facing noisy financial data (S) and complex reward landscapes (R). As emphasized by Sood et al. [2], PPO's consistent and stable performance in volatile markets makes it more practical than theoretically stronger but less stable alternatives. CLSTM-PPO [13] builds upon this foundation by adding specialized components for temporal feature modeling in the state space.

Notably, portfolio constraints (e.g., $\sum w_i = 1$) typically require post-processing steps like softmax functions.

Actor-Critic Methods in Stabilized Learning for Continuous Actions: Actor-critic methods offer a critical balance in PM: the actor handles continuous actions (A), while the critic stabilizes learning through variance reduction in noisy financial settings (S, P, R).

The ensemble approach by Yang et al. [12] reveals important insights about algorithm selection. Motivated by the observation that no single algorithm consistently performs across market conditions (P), they combined DDPG [25] for continuous control (A), PPO for overall stability, and A2C [26] for its parallelism benefits. Their dynamic selection mechanism tackles the challenge of the non-stationary market by adaptively switching between algorithms, offering a practical solution when standard theoretical assumptions of RL do not hold in real-world market conditions.

The relative absence of exploration-focused methods like SAC [27] in PM literature, despite their theoretical appeal for volatile markets, suggests a significant bias toward exploitation and stability in practice—a potentially limiting factor in discovering novel trading strategies.

In practice, PPO implementations in PM typically follow actor-critic designs using GAE for advantage estimation, as evidenced in recent studies [2], [4], [13].

B. Tackling Market Complexity: Interactions and Hierarchy

Standard single-agent formulations often oversimplify market realities. Yang et al. [12] ensemble serves as a heuristic for robustness against market non-stationarity rather than explicitly modeling agent interactions. True MARL approaches like MAPS [28] assign a separate agent to each asset, enabling decentralized portfolio control while sharing the same market data. By jointly optimizing for high individual returns and low inter-agent action correlation, MAPS adapts to changing market conditions more robustly than single-agent approaches.

Hierarchical RL [29] addresses the complexity of PM by decomposing it into multiple levels—strategic allocation versus tactical execution—mirroring hierarchical planning principles. This approach aims to improve efficiency and interpretability in managing complex, multi-timescale financial decisions.

IV. Practical Considerations in DRL for Portfolio Management

The successful deployment of Deep Reinforcement Learning algorithms for portfolio management requires careful navigation of several practical design choices. This section examines how reward functions, state representations, and evaluation methodologies interact with algorithm selection, impacting the agent's ability to achieve meaningful financial objectives.

A. Reward Function Design: Translating Financial Goals into Learning Signals

The reward function fundamentally defines the problem the RL agent solves. Translating complex, often conflicting financial goals—maximizing return, minimizing risk, controlling costs—into a single scalar signal is arguably the most critical aspect of applying RL to portfolio management [3]. Different reward formulations represent different trade-offs and implicit assumptions about investor preferences.

Return-Based Rewards and Transaction Costs: Simple profit or return-based rewards, while intuitively appealing, often prove insufficient for real-world portfolio management due to their risk-neutral nature. Common implementations include raw portfolio value changes (ΔPV_t, e.g., calculated as $v' - v$ in y Yang et al. [12]; or incorporating holdings and costs as $(b_{t+1} + p_{t+1}^T h_{t+1}) - (b_t + p_t^T h_t) - c_t$ in Zou et al. [13]) or logarithmic returns ($r_t = \ln(PV_t / PV_{t-1})$ used by [1]).

To address risk-neutrality and discourage excessive trading, transaction costs are frequently integrated. Jiang et al. [1] incorporate a cost factor μ_t into the log-return ($r_t = \ln(\mu_t \mathbf{y}_t^T \mathbf{w}_{t-1})$) to penalize portfolio turnover based on weight changes ($\|\mathbf{w}_t - \mathbf{w}_t'\|_1$). Zou et al. [13] achieve a similar effect by directly subtracting transaction costs c_t from the portfolio value change in their reward.

Risk-Adjusted Rewards (DSR and DSoR): Risk-adjusted rewards align the RL objective more closely with standard financial metrics. The Differential Sharpe Ratio (DSR) [9],

applied in Sood et al. [2] and Acero et al. [4], offers a per-step optimization target reflecting instantaneous risk-adjusted return. It relies on online moment estimates (R_t is portfolio return at step t, η is learning rate):

$$A_t = A_{t-1} + \eta(R_t - A_{t-1}), \quad B_t = B_{t-1} + \eta(R_t^2 - B_{t-1})$$

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

where $\Delta A_t = R_t - A_{t-1}$ and $\Delta B_t = R_t^2 - B_{t-1}$. To specifically penalize downside risk, Acero et al. [4] introduced the Differential Sortino Ratio (DSoR), modifying the second-moment calculation:

$$B_t^{\text{Sortino}} = B_{t-1}^{\text{Sortino}} + \eta(\min(R_t, r_f)^2 - B_{t-1}^{\text{Sortino}})$$

While principled, the reliance on potentially noisy online estimates, especially in non-stationary markets, is a practical challenge [3].

Multi-Objective Rewards and Constraint Integration: Beyond risk-adjusted returns, DRL can incorporate diverse objectives. Acero et al. [4] demonstrate integrating ESG scores ($\mathbf{w}^T\mathbf{s}$) with financial rewards (u_{fin} like DSR/DSoR) via additive or multiplicative utility functions, where $\mathbf{u}$ represents uniform portfolio weights and α balances objectives:

$$\mathcal{U}_{\text{add}} = u_{\text{fin}} + \alpha\frac{\mathbf{w}^T\mathbf{s}}{\mathbf{u}^T\mathbf{s}}, \quad \mathcal{U}_{\text{mult}} = u_{\text{fin}} \cdot \left(1 + \alpha\frac{\mathbf{w}^T\mathbf{s}}{\mathbf{u}^T\mathbf{s}}\right)$$

This reward-shaping approach allows direct optimization of complex preferences. It differs from external rule-based systems like Yang et al. [12]'s use of a market turbulence index ($T = (\mathbf{y}_t - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{y}_t - \boldsymbol{\mu})$) to override the agent's actions (halt buying) in extreme market states (e.g., when T exceeds 90th percentile), which enforces constraints reactively rather than proactively through the reward signal.

B. State Representation: Informing the Agent's Decisions

The state representation defines the information available to the agent. Its design involves balancing information against complexity, noise, and the challenges of non-stationarity and partial observability inherent in financial markets.

Core Financial Data and Agent's Internal State: Historical price and volume (OHLCV) over a lookback window are standard inputs. Representation choices vary: raw prices, returns, normalized price tensors [1], or log returns [2]. The lookback length (e.g., $T = 60$ days [2], $n = 50$ periods [1]) dictates the temporal patterns learnable by the agent. Critically, the agent's internal state, such as previous portfolio weights $\mathbf{w}_{t-1}$ [1] or current holdings h_t and cash balance b_t [12], [13], is usually included to provide context for decisions involving transaction costs and constraints.

Technical Indicators and Market Context: Technical indicators (e.g., MACD, RSI, CCI, ADX used by Zou et al. [13], Acero et al. [4]) are often added to summarize trends and momentum, despite increasing dimensionality and debated predictive power. Market context features, like volatility metrics (rolling volatility, VIX index [2]) or market condition scores [22], aim to help the agent adapt to different market regimes.

Alternative Data Sources and ESG Integration: To enrich the state space, alternative data is explored. Ye et al. [5] used processed news embeddings to capture sentiment, facing challenges in reliability (reporting 61% test accuracy for their sentiment classifier) and correlation with price movements. They also tested auxiliary model predictions (asset up/down movement) as state features. Demonstrating multi-objective integration, Acero et al. [4] directly included ESG scores in the state to solve for responsible investing strategies. The accuracy of these scores from the given source still poses a challenge.

Neural Architecture Selection for State Processing: Neural network architecture choice is key to processing complex states. Temporal dependencies are often handled by LSTMs [13] or RNNs [1]. Other structural aspects are captured using CNNs for price tensors [1], attention mechanisms to focus on relevant features [14], and Graph Convolutional Networks (GCN) for inter-asset relationships [22]. Architecture selection requires domain knowledge, and increased complexity raises overfitting risk, a major concern in noisy financial data.

Creating effective state representations for financial reinforcement learning remains challenging [3]. Balancing information richness with tractability, managing noise and non-stationarity, and mitigating the effects of partial observability continue to be critical research challenges for advancing DRL applications in portfolio management.

C. Evaluation Methods: Ensuring Meaningful Performance Assessment

Robust evaluation is indispensable for validating DRL-based portfolio management strategies and understanding their potential practical value. Flawed backtesting can yield misleadingly optimistic results.

Comprehensive Performance Metrics: An effective assessment requires multiple metrics addressing different dimensions of strategy quality. While returns (cumulative or annualized) measure basic profitability [2], [13], risk metrics like volatility [2] and, crucially for practitioners, maximum drawdown [2], [13] capture stability and potential losses. Risk-adjusted performance is gauged by Sharpe Ratio [2], [4], [12], [13], Sortino Ratio [4], or Calmar Ratio [2].

Trading behavior, impacting real-world costs, is assessed via portfolio turnover [2] or number of trades (used to calculate Average Profitability Per Trade in Zou et al. [13]). Depending on the evaluation objectives, additional evaluation criteria may include distributional properties of returns—such as skewness, kurtosis, and Value-at-Risk (VaR) [2]—as well as non-financial metrics like ESG scores [4].

Methodological Rigor in Backtesting: The backtesting methodology requires scrutiny. Strict chronological data splitting (train/validation/test) is essential to prevent lookahead bias, as practiced by Zou et al. [13] and Yang et al. [12] (using 2009-2015 train, 2016-2020 test). [2] used a 10x7 year rolling window (5 for testing, 1 for validation, and 1 for testing), while Yang et al. Sood et al. [12] and Zou et al. [13] employed periodic retraining within their testing phase.

Realistic Transaction Cost Modeling: Accurate modeling of transaction costs (e.g., 0.1% per trade in Zou et al. [13],

0.2% of the invested capital Neuneier [8] 0.25% in Jiang et al. [1]) and potentially market impact/slippage is crucial. Neglecting these factors leads to unrealistic performance claims based on simplified testing scenarios.

Meaningful Baselines and Standardization Challenges: Performance requires context from strong baselines. Comparing against traditional methods like Mean-Variance Optimization (as done by Sood et al. [2], Acero et al. [4]) demonstrates practical relevance. Outperforming market indices (S&P 500 [12], DJI [13]) and simple strategies (Buy-and-Hold [1], [5], [13], Equally Weighted [13]) is a minimum requirement.

However, a significant challenge is the lack of standardization in evaluation practices. Diverse financial datasets ranging from crypto to equity, time-frames, costs, metrics, and baselines make it difficult to determine if the reported performance differences arise from algorithmic superiority or experimental setup variations. Initiatives promoting standardized benchmarks, e.g., FinRL, and reporting protocols are needed [3].

Rigorous and realistic evaluation is non-trivial but essential for advancing financial RL research. Addressing common pitfalls in backtesting, using strong baselines, and adopting standardized protocols are necessary steps for developing reliable and usable portfolio management solutions.

V. Conclusion

This survey provides a structured overview of the application of Deep Reinforcement Learning to stock portfolio management, synthesizing insights from key research contributions and the provided course materials. We have traced the field's trajectory from early adaptations of basic RL algorithms to the current SOTA, which leverages complex deep learning architectures, sophisticated state representations incorporating diverse data sources, and carefully engineered reward functions aimed at optimizing risk-adjusted returns and other objectives like ESG responsibility. Key advancements include the handling of continuous action spaces [1], the incorporation of diverse data sources through state augmentation [5] and attention mechanisms [14], [22], the use of graph networks [22], temporal modeling [13], ensemble strategies for robustness [12], and the integration of financial domain knowledge [4].

Despite promising results demonstrated in numerous backtesting studies [13], significant challenges remain for reliable real-world deployment. The inherent non-stationarity and noise of financial markets [3], [30], coupled with DRL's sample inefficiency and sensitivity [31], pose major challenges. The 'black-box' nature hinders interpretability and trust [32], while the gap between simplified backtesting assumptions and live market complexities (market impact, liquidity) limits the direct applicability of many research findings [30].

Future research might fix these fundamental limitations by several interconnected approaches. Hybrid models that inherit the learning power of DRL and the explainability of traditional financial models offer the promise of a trade-off between performance and explainability. The development of durable and flexible agents that can identify and react to regime shifts is yet another essential research direction. These systems should embed uncertainty quantification and risk-sensitive learning procedures in order to uphold performance under various market regimes. Sophisticated frameworks like Hierarchical Reinforcement Learning (HRL) and Multi-Agent Reinforcement Learning (MARL) propose solutions for modeling the intricate, multi-scale financial market dynamics, allowing models to acquire short-term tactical choices and long-term strategic placements. The creation of high-fidelity simulators and standardized benchmarks [3], [30]. Addressing these challenges will be key to unlocking the full potential of DRL as a transformative tool for automated, adaptive, and potentially superior portfolio management in the complex financial landscape.

References

[1] Z. Jiang, D. Xu, and J. Liang, "Deep reinforcement learning for portfolio management," *arXiv preprint arXiv:1706.10059*, 2017.

[2] S. Sood, K. Papasotiriou, M. Vaiciulis, and T. Balch, "Deep reinforcement learning for optimal portfolio allocation: A comparative study with mean-variance optimization," in *Planning and Scheduling for Financial Services Workshop, ICAPS*, 2023.

[3] Y. Bai, Y. Gao, R. Wan, S. Zhang, and R. Song, "A review of reinforcement learning in financial applications," *arXiv preprint arXiv:2411.12746*, 2024.

[4] F. Acero, P. Zehtabi, N. Marchesotti, M. Cashmore, D. Magazzeni, and M. Veloso, "Deep reinforcement learning and mean-variance strategies for responsible portfolio optimization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 1112–1119.

[5] Y. Ye, H. Pei, B. Wang, *et al.*, "Reinforcement-learning based portfolio management with augmented asset movement prediction states," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 1112–1119.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd. Cambridge, MA: MIT Press, 2018.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[8] R. Neuneier, "Enhancing q-learning for optimal asset allocation," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 10, 1997, pp. 936–942.

[9] J. Moody and M. Saffell, "Performance functions and reinforcement learning for trading systems and portfolios," *Journal of Forecasting*, vol. 17, no. 5-6, pp. 441–470, 1998.

[10] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 653–664, 2016.

[11] G. Lucarelli and M. Borrotti, "A deep q-learning portfolio management framework for the cryptocurrency market," *Neural Computing and Applications*, vol. 32, no. 23, pp. 17 229–17 244, 2020.

[12] H. Yang, X. Liu, S. Zhong, and A. Walid, "Deep reinforcement learning for automated stock trading: An ensemble strategy," in *Proceedings of the ACM International Conference on AI in Finance*, 2020.

[13] J. Zou, J. Lou, B. Wang, and S. Liu, "A novel deep reinforcement learning based automated stock trading system using cascaded lstm networks," *Expert Systems with Applications*, 2023.

[14] J. Wang, Y. Zhang, K. Tang, J. Wu, and Z. Xiong, "Alphastock: A buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*, 2019, pp. 1900–1908.

[15] H. Ma, F. Li, J. Y. Lim, Z. Luo, T. V. Vo, and T.-Y. Leong, "Catching two birds with one stone: Reward shaping with dual random networks for balancing exploration and exploitation," in *Forty-second International Conference on Machine Learning*, 2025.

[16] H. Ma, Z. Luo, T. V. Vo, K. Sima, and T.-Y. Leong, "Highly efficient self-adaptive reward shaping for reinforcement learning," in *Thirteenth International Conference on Learning Representations*, 2025.

[17] H. Ma, K. Sima, T. V. Vo, D. Fu, and T.-Y. Leong, "Reward shaping for reinforcement learning with an assistant reward agent," in *Forty-first International Conference on Machine Learning*, 2024.

[18] H. Ma, Z. Luo, T. V. Vo, K. Sima, and T.-Y. Leong, *Centralized reward agent for knowledge sharing and transfer in multi-task reinforcement learning*, 2025. arXiv: 2408.10858.

[19] Z. Luo, H. Ma, D. Shi, and W.-S. Gan, "Gfanc-rl: Reinforcement learning-based generative fixed-filter active noise control," *Neural Networks*, vol. 180, p. 106 687, 2024.

[20] H. Ma, T. V. Vo, and T.-Y. Leong, "Mixed-initiative bayesian sub-goal optimization in hierarchical reinforcement learning," in *Proceedings of the 23rd international conference on autonomous agents and multiagent systems*, 2024, pp. 1328–1336.

[21] H. Ma, T. V. Vo, and T.-Y. Leong, "Hierarchical reinforcement learning with human-ai collaborative subgoals optimization," in *Proceedings of the 22nd international conference on autonomous agents and multiagent systems*, 2023, pp. 2310–2312.

[22] Z. Wang, B. Huang, S. Tu, K. Zhang, and L. Xu, "Deeptrader: A deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 643–650.

[23] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 32, Beijing, China: PMLR, 2014, pp. 387–395.

[24] Z. Luo, J. Li, and Y. Zhu, "A deep feature fusion network based on multiple attention mechanisms for joint iris-periocular biometric recognition," *IEEE Signal Processing Letters*, vol. 28, pp. 1060–1064, 2021.

[25] T. P. Lillicrap, J. J. Hunt, A. Pritzel, *et al.*, "Continuous control with deep reinforcement learning," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[26] V. Mnih, A. P. Badia, M. Mirza, *et al.*, "Asynchronous methods for deep reinforcement learning," in *Proceedings of the 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., ser. Proceedings of Machine Learning Research, vol. 48, 2016, pp. 1928–1937.

[27] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor–critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, J. G. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, 2018, pp. 1856–1865.

[28] J. Lee, R. Kim, S.-W. Yi, and J. Kang, "MAPS: Multi-agent reinforcement learning-based portfolio management system," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, 2020, pp. 4520–4526.

[29] R. Wang, H. Wei, B. An, Z. Feng, and J. Yao, "Commission fee is not enough: A hierarchical reinforced framework for portfolio management," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, vol. 35, AAAI Press, 2021, pp. 626–633.

[30] S. Sun, M. Qin, X. Wang, and B. An, "Prudex-compass: Towards systematic evaluation of reinforcement learning in financial markets," *arXiv preprint arXiv:2302.00586*, 2023.

[31] A. Millea, "Deep reinforcement learning for trading—a critical survey," *Data*, vol. 6, no. 11, p. 119, 2021.

[32] R. Israel, B. Kelly, and T. Moskowitz, "Can machines "learn" finance?" *Journal of Investment Management*, vol. 18, no. 2, pp. 23–36, 2020.