ANNUAL REVIEWS

Annual Review of Statistics and Its Application

A Review of Reinforcement Learning in Financial Applications

Yahui Bai,[1,*] Yuhe Gao,[1,*] Runzhe Wan,[2,*]
Sheng Zhang,[2,*] and Rui Song[2]

[1] Department of Statistics, North Carolina State University, Raleigh, North Carolina, USA

[2] Amazon, Seattle, Washington, USA; email: songray@gmail.com

Annu. Rev. Stat. Appl. 2025. 12:209–32

First published as a Review in Advance on November 15, 2024

The *Annual Review of Statistics and Its Application* is online at statistics.annualreviews.org

https://doi.org/10.1146/annurev-statistics-112723-034423

*These authors contributed equally and are listed in alphabetical order.

Keywords

finance, reinforcement learning, statistics, deep learning, machine learning, meta-analysis

Abstract

In recent years, there has been a growing trend of applying reinforcement learning (RL) in financial applications. This approach has shown great potential for decision-making tasks in finance. In this review, we present a comprehensive study of the applications of RL in finance and conduct a series of meta-analyses to investigate the common themes in the literature, such as the factors that most significantly affect RL's performance compared with traditional methods. Moreover, we identify challenges, including explainability, Markov decision process modeling, and robustness, that hinder the broader utilization of RL in the financial industry and discuss recent advancements in overcoming these challenges. Finally, we propose future research directions, such as benchmarking, contextual RL, multi-agent RL, and model-based RL, to address these challenges and to further enhance the implementation of RL in finance.

1. INTRODUCTION

A financial market is a marketplace where financial instruments such as stocks and bonds are bought and sold (Fama 1970). Individuals and organizations can play crucial roles in financial markets to facilitate the allocation of capital. Market participants face diverse challenges, such as portfolio management, which aims to maximize investment returns over time, and market making, which seeks to profit from the bid–ask spread while managing inventory risk. As the volume of financial data has increased dramatically over time, new opportunities and challenges have arisen in the analysis process, leading to the increased adoption of advanced machine learning (ML) models.

Reinforcement learning (RL) (Sutton & Barto 2018), as one of the main categories of ML, has revolutionized the field of artificial intelligence by empowering agents to interact with the environment and allowing them to learn and improve their performance. The success of RL has been demonstrated in various fields, including games, robots, and mobile health (Nash 1950, Kalman 1960, Murphy 2003). In finance, applications such as market making, portfolio management, and order execution can benefit from the ability of RL algorithms to learn and adapt to changing environments. Compared with traditional models that rely on statistical techniques and econometric methods such as time series models [autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA)], factor models, and panel models, the RL framework empowers agents to learn decision-making by interacting with an environment and deducing the consequences of past actions to maximize cumulative rewards (Charpentier et al. 2021). RL also facilitates online learning, allowing iterative refinement of investment strategies with new information. Moreover, through their deep network structure, RL models can better capture complex patterns in highly nonlinear and nonstationary financial data. These advantages give RL the potential to enhance both the efficiency and effectiveness of financial applications.

Despite numerous studies that demonstrate performance enhancements compared with conventional methods, it is crucial to thoroughly assess the challenges associated with applying RL in the financial domain. Financial data can be noisy and nonstationary, while often following a heavy-tailed distribution, and may involve a mix of frequencies. These features can present challenges to RL algorithms both in theory and in practice. This article aims to provide an overview of recent studies on the use of RL in three critical finance applications: market making, portfolio management, and optimal execution (OE). The main focus is to highlight challenging areas where further exploration would be potentially valuable. For all three areas mentioned above, we select recently published papers with a focus on those with high-quality and relevant experiments that can support our meta-analysis.

1.1. Related Work

Several papers have reviewed the use of RL in finance. Fischer (2018) and Pricope (2021) categorize model-free RL approaches into critic-only, actor-only, and actor–critic, and discuss various RL settings. Specifically, Fischer (2018) summarizes the challenges in financial data and suggests the actor-only approach may be the best suited for RL in financial markets. Pricope (2021) identifies limitations in current studies on deep reinforcement learning (DRL) in quantitative algorithmic trading and suggests that more research is needed to determine its ability to outperform human traders. Charpentier et al. (2021) and Huang et al. (2020) delve deep into the realm of RL applications, with a primary focus on their widespread use in economics, operations research, and banking. Other works focus on application in subdomains of finance. Hambly et al. (2023) conduct a survey of recent studies in RL applied to finance, concentrating on critical topics like OE and portfolio optimization, while Gašperov et al. (2021) focus on the review of RL in market making. Meng & Khushi (2019) diligently review research papers related to trading and forecasting in stock and foreign currency markets, shedding light on the prevalence of unrealistic assumptions within many of these studies, and Millea (2021) reviews recent developments in DRL for trading in the cryptocurrency market. Mosavi et al. (2020) compare ML and DRL methods in different financial applications, suggesting DRL may outperform traditional approaches.

Traditionally, RL methods have been applied to healthcare, such as precision medicine (Chakraborty & Murphy 2014, Kosorok & Laber 2019). Key differences between RL in finance and medical research include the following: Many RL agents in precision medicine are trained offline using observational or clinical trial data, while most RL policies in finance are trained online with simulators or real market data; traditional precision medicine settings typically involve finite horizons, whereas RL in financial markets often deals with infinite horizon decision-making; and financial environments can involve multiple agents, such as in market making, while precision medicine usually involves a single RL agent. It is worth mentioning that RL applications in mobile health devices are similar to those in finance, as both involve online RL with infinite decision stages.

1.2. Contribution

In this article, we conduct a comprehensive survey of RL methods in the financial domain. Our contributions are summarized as follows.

- We provide a few taxonomies to establish a unified view of the recent literature about RL in financial applications, including market making with a single agent and multiple agents, portfolio management, and OE.
- We conduct a series of meta-analyses based on the reviewed papers to investigate the common patterns in the literature, such as which factor affects the performance of RL the most compared with traditional methods.
- We outline a few key challenges that emerge when using RL in financial applications. In addition, we dive into a comprehensive discussion on the recent advances and progress made in addressing these challenges. We also suggest four future directions for these problems.

The rest of the article is organized as follows. Section 2 summarizes the concepts of RL and commonly used RL algorithms in financial applications. Section 3 provides a collection of RL papers in financial domains. Section 4 presents meta-analyses in different papers to better understand the performance of RL in financial applications. Section 5 discusses the current status of environments, benchmarking and open-source packages in finance RL. Section 6 discusses challenges for the application of RL in the financial domain. Section 7 concludes with a few future research directions.

2. PRELIMINARY

To begin our financial literature survey, we introduce RL algorithms, shown in **Figure 1** categorized by frequency of use and into model-free and model-based algorithms. In model-free RL algorithms, the agent learns to make decisions without a model of the environment's dynamics, while in model-based RL algorithms, the environment's dynamics is modeled.

There are three categories of model-free RL algorithms, including actor-only, critic-only, and actor–critic methods. Actor-only algorithms update only the policy, or the actor, while keeping the value function fixed. One popular actor-only algorithm is called the policy gradient (PG) (Sutton et al. 1999), which directly optimizes the policy parameters to maximize the expected cumulative reward. Another commonly used actor-only algorithm is proximal policy optimization (PPO). One

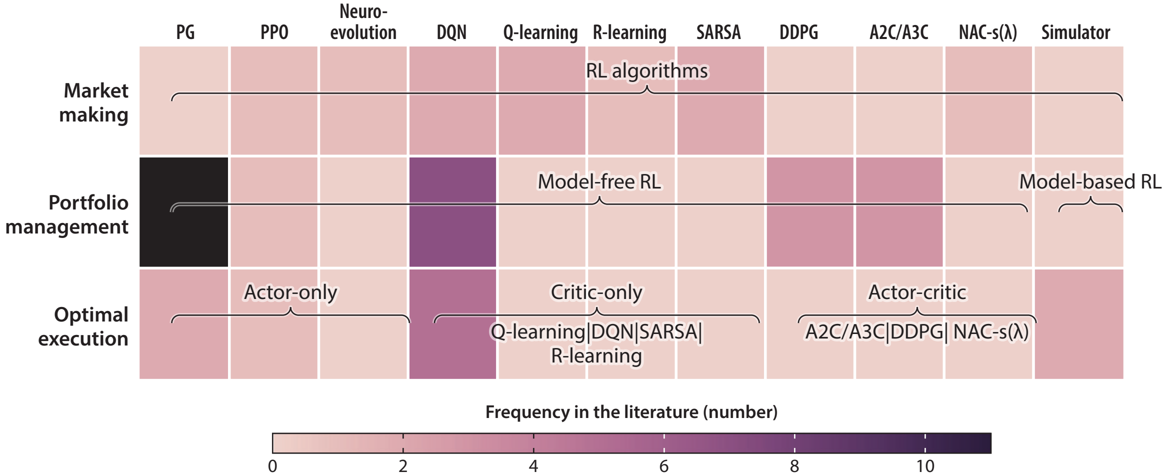


Figure 1

RL algorithms used in the literature and the frequency of their usage. Abbreviations: A2C, advantage actor–critic; A3C, asynchronous advantage actor–critic; DDPG, deep deterministic policy gradient; DQN, deep Q-network; NAC, natural actor–critic; PG, policy gradient; PPO, proximal policy optimization; RL, reinforcement learning; SARSA, state–action–reward–state–action.

of the key features of PPO is that it employs a proximal update rule, which limits the size of the policy update at each iteration. Neuroevolution (Such et al. 2017) differs from the aforementioned algorithms as it adapts a gradient-free genetic algorithm to optimize the policy.

Critic-only methods focus solely on learning the value function, without explicitly learning a policy function. The actions are obtained based on the approximate value function (Sutton & Barto 2018). Q-learning estimates the optimal action-value function, which represents the expected total reward for taking a given action in a given state and following the optimal policy thereafter (Watkins & Dayan 1992). A deep Q-network (DQN) (Mnih et al. 2013) uses a deep neural network to estimate the Q-value function. Another type of critic-only algorithm is called SARSA (state–action–reward–state–action) (Sutton & Barto 2018), which updates the Q-value function using the observed reward and the next state-action pair. Additionally, R-learning (Schwartz 1993) is also value-based, except that its objective is to maximize the average reward rather than the discounted reward in Q-learning and SARSA.

Actor–critic methods combine the benefits of both policy-based (actor) and value-based (critic) approaches. These methods leverage two separate components to learn both the policy and value functions, with the actor generating actions based on the learned policy and the critic network evaluating the quality of the actions taken by the actor. Advantage actor–critic (A2C) and asynchronous advantage actor–critic (A3C) are popular and effective actor–critic methods with deep learning techniques. A2C, proposed by Mnih et al. (2016), combines the actor–critic algorithm with parallelization techniques to improve the learning speed and stability. As an extension of A2C, A3C employs asynchronous updating of the actor–critic model by running multiple threads in parallel. Deep deterministic policy gradient (DDPG) (Lillicrap et al. 2019) combines ideas from the deterministic PG algorithm and the DQN algorithm, which is particularly effective in continuous action spaces. The NAC-s(λ) algorithm, proposed by Thomas (2014), is a natural actor–critic method that uses semigradient SARSA for policy evaluation and is specifically designed for stochastic policies.

Model-based methods involve learning a model of the environment's dynamics and using this model to simulate the future to select actions. In financial applications like market making and

OE, the model-based RL approach entails constructing and leveraging a simulator that mimics the dynamics of the financial market. This simulator acts as a predictive model, providing agents with valuable insights about how the market may behave under different conditions and scenarios. By employing this model as a proxy of the real market, agents can execute a multitude of trades and test various trading strategies. While these methods can be computationally intensive, they are shown to produce better long-term behavior when the environment is complex and uncertain (Sutton & Barto 2018).

3. APPLICATIONS OF REINFORCEMENT LEARNING IN FINANCE

3.1. Summary of Markov Decision Process Formulations in the Literature

Supplemental Material >

We first present a taxonomy of the state, action, reward, and algorithm used in the surveyed papers, to provide a unified view of the literature, in **Supplemental Tables 1** and **2**. We organize the state, action, and reward variables into several categories: (*a*) The state variables include the price information (historical prices, returns, change of price, etc.), inventory information (the on-hand units for one stock or every stock in the portfolio), market indicators (Citigroup Economic Surprise Index, etc.), technical indicators (relative strength index, moving average convergence divergence, etc., denoted as “tech-indicators” in **Supplemental Table 1**), company information (price-to-earnings ratio, dividend, market capitalization, etc.), predictive representations of state, trading volume (including volume of past trades, change of volume, market share of different agents, etc.), bid–ask information (bid–ask spread, imbalance of bid–ask order in the limit order book, etc.), volatility, end-to-end information (E2E, where we directly use a large number of observable variables), and time index. (*b*) The action variables are summarized as long/short/hold, the target portfolio weight, the bid price, and the trading quantity. (*c*) The reward variables are summarized as profit and loss (PnL) (including cash income and outflow, and some other customized function of profit), return, volatility-adjusted return (which includes the Sharp ratio and other variants), implementation shortfall, slippage, various shaped rewards, inventory penalty, and market quality [containing bid–ask spread, some measures of price change, etc., as discussed by Chan & Shelton (2001)]. Here, a shaped reward (Lin & Beling 2020a,b) refers to a reward definition that is not the same as any existing metrics but instead has a relatively complicated (but carefully designed) structure to achieve a special goal.

3.2. Applications in Market Making

Market makers increase the liquidity of the market by continuously offering buy and sell orders. In general, market makers profit from order execution (i.e., the spread profit) as well as inventory speculation resulting from changes in the market prices of the assets they hold. However, maintaining a nonzero inventory also exposes market makers to the inventory risk, which is the potential risk of adverse price change in the market (O’Hara & Oldfield 1986). Thus, market making algorithms aim at maximizing profit while controlling this inventory risk. RL methods have been introduced to market making since 2001 (see Chan & Shelton 2001), with most works emerging in the past five years (post-2018). Most researchers use a single RL agent to handle the problem of market making while incorporating techniques to mitigate various risks, a representative selection of which is reviewed in Section 3.2.1. Alternatively, some works investigate a multi-agent framework to achieve various objectives, such as increasing the robustness of the RL agent from some adversaries or deriving a nearly optimal strategy in the presence of competing market makers. These approaches are reviewed in depth in Section 3.2.2.

3.2.1. Single-agent reinforcement learning in market making.

The application of reinforcement learning (RL) in market making was first introduced by Chan & Shelton (2001). This pioneering work integrates the problem of market making into the RL framework, considering both the agent's information (such as inventory) and market characteristics (like the imbalance of buy and sell orders, bid–ask spread, and price changes between trades) as state variables. The RL agent's actions range from setting bid–ask prices, setting bid–ask order sizes, and executing buy/sell orders. The rewards of RL agents come from profit maximization, inventory minimization, and improvement of market quality. In the simulation model of Chan & Shelton (2001), a single security is traded and state variables are influenced by the actions of the market maker, informed traders (with inside information distribution), and uninformed traders.

Subsequent research focuses on mitigating inventory and other risks for the RL market maker agent in various settings. Mani et al. (2019) introduce a novel approach using the double SARSA algorithm and a customized temporal difference update for the Q-function. This approach leads to variance reduction in rewards, enabling the agent to achieve higher profits while maintaining lower inventory levels compared with the standard SARSA policy from Chan & Shelton (2001). Another line of research involves modifying the reward function to manage risks. For instance, Spooner et al. (2018) design an asymmetrically dampened reward function that encourages the agent to receive rewards by providing more liquidity rather than holding a high inventory, thereby reducing the inventory risk. Selser et al. (2021) propose to modify the reward function by incorporating the estimated variance of the agent's wealth as a penalty term. In the simulation results of Selser et al. (2021), it is demonstrated that a DQN model with this reward function can outperform a non-RL baseline that solves partial differential equations (Avellaneda & Stoikov 2008). Guéant & Manziuk (2019) extend this risk-averse reward function strategy to a multi-asset market making setting. They propose an actor–critic algorithm to approximate the optimal quote for multiple bonds. Their RL model includes a penalty term for the portfolio's PnL variance in the reward function, and this approach is proved to be scalable in high-dimensional cases involving up to 100 bonds—a task challenging for non-RL methods.

3.2.2. Multi-agent reinforcement learning in market making.

The multi-agent RL framework is an extension to the single-agent RL framework for market making, and the research directions can be divided into three branches. The first branch introduces adversarial agents in simulations, such as these agents profit from the losses of RL market making agents. This approach enhances the RL agent's robustness to market conditions or increases risk aversion. For example, Spooner & Savani (2020) explore an adversarial RL setting in which an adversarial agent during the RL agent's training alters the environmental parameters in simulators. This discrepancy in perceived versus actual market conditions trains the market maker to be more robust against market condition misspecifications. Extending this idea, Gašperov & Kostanjčar (2021) introduce an adversary that directly disrupts the actions of the market making agent by displacing quotes. They show that this technique reduces inventory risk and improves profits compared with training the RL agent without an adversary.

The second branch focuses on competition between market makers, where a professional market making agent competes with the RL agent. Ganesh et al. (2019) first demonstrate theoretically that the RL market making agent's optimal pricing policy to maximize spread PnL depends on its competitors' pricing distributions. Ganesh et al. (2019) show in their experiments that the RL agent can learn the pricing distribution of competing market makers from pricing and trading observations and derive a nearly optimal strategy accordingly, despite the parameters of the competitor agents being unknown to the RL agent. In the third branch, the focus is mainly on adapting the original market making decision process into a hierarchical one. Patel (2018) transforms the RL framework by introducing a macroagent that determines the overall trading strategy (buy, sell, or hold the security), while a microagent sets specific quote prices based on the macroagent's direction. This hierarchical approach results in less volatile profits, as shown in their simulations.

3.3. Applications in Portfolio Management

In portfolio management, the agent aims to distribute a fixed sum across diverse assets, aiming to maximize the portfolio's expected return while prudently managing risk. Although classical methods such as the Markowitz model (Markowitz 1952) have been proven to be effective in the past, they rely on a static approach that may not be suitable for a rapidly changing market. Hence, introducing a dynamic portfolio optimization algorithm capable of adapting to changing market conditions is crucial for better risk management and return improvement. We review bandit algorithms in Section 3.3.1 and deep RL algorithms in Section 3.3.2.

3.3.1. Bandit algorithms for portfolio management. The bandit problem (Lattimore & Szepesvári 2020) is a special case of RL, in that there does not exist a state transition, and hence, there is no long-term dependency. Bandit algorithms have been studied in a few papers on portfolio optimization applications (Shen et al. 2015, Shen & Wang 2016, Zhu et al. 2019, Huo & Fu 2017), where the investor dynamically updates the portfolio based on historical observations.

Shen et al. (2015), Shen & Wang (2016) , and Zhu et al. (2019) all adopt a standard bandit algorithm called Thompson sampling (Russo et al. 2018) in a noncontextual multi-armed bandit setting, with the major difference of how to define the arms. Shen et al. (2015) did the first work on bandits for portfolio optimization. This work focuses on the challenge that it is not appropriate to regard stocks as independent arms (due to the correlation between them). With the goal of risk diversification, the authors propose to integrate bandits with a popular practice in portfolio optimization, which is first applying principal component analysis to the stock returns and then regarding the eigenvectors as arms. Therefore, the article addresses the limitation of a naive application of bandit algorithms in portfolio optimization and provides a nice integration with finance theory, which is further supported by the performance of the backtest when compared with several other online portfolio optimization algorithms. Shen & Wang (2016) similarly apply Thompson sampling to portfolio optimization. The authors study portfolio blending—i.e., they regard the portfolios given by different selection strategies as arms and aim to find the best combinations of these strategies. The algorithm presented by Shen & Wang (2016) is designed to mix two base strategies, which is further extended by Zhu et al. (2019) to allow mixing multiple base strategies. Superior performance over several online portfolio optimization algorithms as well as the base strategies is reported in backtesting.

The aforementioned works focus on maximizing the cumulative expected return while ignoring another important metric in finance, risk. To fill this gap, Huo & Fu (2017) propose to consider a linear combination between a stock selected using a bandit algorithm and another portfolio solved by (greedy) online portfolio optimization. The former targets the expected return (in the long run), and the latter is selected by minimizing the risk. By choosing the weights of the two parts carefully and manually, the authors aim to achieve a balance between the expected return and the risk.

In closing, we raise a fundamental question: Is the bandit problem ideal for portfolio optimization? In other words, is active exploration necessary in this application? As highlighted by Russo et al. (2018, section 8.2.1), this question warrants further discussion, which we delve into in Section 6.

3.3.2. Reinforcement learning algorithms for portfolio management. In contrast to bandit algorithms, which focus solely on immediate rewards, RL algorithms approach the sequential portfolio management problem by formulating it as a Markov decision process (MDP). By doing so, these algorithms take into account the consequences of current portfolio decisions on future states, making them more comprehensive and forward-looking. In this subsection, we review papers that use the RL algorithms for training trading agents in portfolio management.

To begin with, Jiang et al. (2017) propose the Ensemble of Identical Independent Evaluators, which is an ensemble of neural networks with history price as input, to solve the asset allocation problems in the cryptocurrency market. The study explores three backbone networks [recurrent neural networks (RNN), long short-term memory (LSTM), and convolutional neural networks] as RL policy networks and demonstrates that RL models can outperform traditional approaches, indicating their potential for portfolio management. Subsequent studies in RL algorithm trading agents consider this paper as a benchmark.

3.3.2.1. Complicated policy networks. Liu et al. (2022b) build upon Jiang et al. (2017) by adopting DDPG over PG algorithms for trading agent optimization, resulting in remarkable performance improvement. This paper sheds light on how deep policy networks can improve the RL agents for portfolio management tasks. More complicated policy networks (such as DQN, hierarchical agents, and parallel agents) are explored in other studies. For example, Gao et al. (2020) utilize DQN for portfolio management with discrete action space, where duel Q-net (Wang et al. 2016) is used as the backbone framework of the DQN. To improve sampling efficiency, they also employ prioritized experience replay with the SumTree structure (Schaul et al. 2016) to prioritize valuable experiences over noise during training. R. Wang et al. (2021) propose a hierarchical reinforced portfolio management approach. Specifically, the hierarchical structure includes two agents: a high-level RL agent maximizing long-term profits and a low-level RL agent minimizing trading costs while achieving the wealth redistribution targets set by the high-level model within a limited time window. Similarly, Ma et al. (2021) propose an approach with two parallel agents: one capturing current asset information and the other using LSTM layers to detect long-term market trends.

3.3.2.2. Empirical financial strategy–oriented methods. Some studies have recognized the value of leveraging preexisting domain knowledge to enhance the efficiency and effectiveness of RL models. Traditional financial strategies, like buying winners and selling losers (BWSL) (Jegadeesh & Titman 1993), the Rescorla–Wanger model (Rescorla 1972), modern portfolio theory (MPT) (Menezes & Hanson 1970, Pratt 1978), and the dual thrust strategy, are employed in work discussed below.

AlphaStock (Wang et al. 2019) adopts the BWSL strategy, buying assets with high price-rising rates and selling those with low rates. The authors design two optimizing agents: a long agent purchasing selected winner assets and a short agent selling selected loser assets, optimizing for the Sharpe ratio using a cross-assets attention network. Sensitivity analysis on stock features shows that AlphaStock tends to select stocks with high long-term growth, low volatility, and high intrinsic value, and that are undervalued. Li et al. (2019) note that a bear market, marked by pessimism, aligns with economic downturns, while a bull market, marked by optimism, occurs when security prices outpace interest rates. This paper incorporates market sentiment by modifying the Rescorla–Wanger model to adapt differently under positive and negative environments. MPT (Menezes & Hanson 1970, Pratt 1978) is widely used to construct optimal investment portfolios by balancing the trade-off between risk and return. Zhang et al. (2020) construct the reward function of the RL framework by leveraging MPT, and they show that the constructed reward function is equivalent to MPT's utility function subject to a proposed condition. Another strategy, Michael Chalek's Dual (Liu et al. 2020), is an investment approach combining the true range (a measure of volatility) with the dual thrust model, which sets buy and sell thresholds. Specifically, a buy signal is generated when the market price exceeds the buy threshold, and a sell signal is triggered when it falls below the sell threshold. Liu et al. (2020) utilize the dual thrust strategy to initiate the demonstration buffer, enhancing the deterministic PG method to balance exploration and exploitation effectively.

3.3.2.3. Methods aimed at enhancing robustness.

The volatility of the financial market presents significant challenges when applying RL to real-world investment processes. To ensure consistent performance, some studies prioritize enhancing robustness, i.e., the stability of the model's performance under different market conditions, as a key objective. More discussion on the challenges of robustness can be found in Section 6.2.3.

Yang et al. (2020) propose an ensemble strategy that can dynamically select from three RL algorithms (PPO, A2C, and DDPG) based on market indicators. Their experimental results show that this approach effectively preserves robustness under different market conditions. Lee et al. (2020) also focus on dealing with the continuously changing market conditions. Multiple agents are designed, and each agent aims to optimize their portfolio while considering the potential negative impact on the overall system's risk-adjusted return if portfolios are similar. Another robustness-related RL method is DeepPocket (Soleymani & Paquet 2021), in which the interaction between assets is represented by a graph with financial assets as nodes and the correlation function between assets as edges. The graph is encoded as the state in the actor–critic RL algorithm. The paper shows that DeepPocket can handle unanticipated changes generated by exogenous factors such as COVID-19 in online learning. Benhamou et al. (2021) suggest a multi-network approach that combines financial contextual information with asset states to predict the economy's health, which enables the model to outperform baseline methods, especially during recessions.

3.3.2.4. Market prediction–based methods.

An emerging trend observed among many papers is to include predicted contextual information, such as asset price movements, data augmentation, market trends, market sentiment, or economic conditions.

Market-prediction-based deep RL in the portfolio management domain was first applied by Yu et al. (2019). They implement a prediction module to forecast the next time-step price and a market indicator aiming to augment state space. The authors also propose a behavior cloning module to reduce volatility by preventing a large change in portfolio weights based on imitation learning. Similarly, the data augmentation idea is explored by Théate & Ernst (2021) with a DQN policy network. Other than prices, more complicated state augmentations have been explored. Lei et al. (2020) enhance the RL states by incorporating a prediction-based auto-encoder using the attention mechanism, to capture the market trend. Ye et al. (2020) incorporate the market sentiment feature extracted from relevant news as an external stock movement indicator within the RL framework. Koratamaddi et al. (2021) also take market sentiment into account, by augmenting the state space with a calculated market confidence score derived from company-related news and tweets. DeepTrader (Z. Wang et al. 2021) is another market-prediction-oriented work. The authors design an asset scoring unit to capture asset-specific growth and a market scoring unit to adjust the long/short proportion based on the overall economic situation, which enables the model to balance risk and return and to adapt to changing market conditions.

3.4. Applications in Optimal Execution

OE is another important problem in finance (Bertsimas & Lo 1998). Compared with portfolio optimization, OE is finer-grained: It focuses on the optimal way to buy or sell some prespecified units of a single stock in a given time frame. Therefore, OE becomes an indispensable component of the continuous updating of portfolios. Since the naive strategy that places the entire order instantly may have a significant impact on the market and hence profitability, a well-designed algorithm is required.

In the literature, various metrics are considered to evaluate OE algorithms, including the PnL (Nevmyvaka et al. 2006, Shen et al. 2014, Deng et al. 2016, Wei et al. 2019), a normalized version of the PnL called the implementation shortfall (Hendricks & Wilcox 2014, Lin & Beling 2020b), the Sharpe ratio (Deng et al. 2016), and the shaped reward (Lin & Beling 2020a, Fang et al. 2021).

3.4.1. Model-free reinforcement learning. Nevmyvaka et al. (2006) are the first to attempt to apply model-free RL to OE. Using a modified Q-learning algorithm, they conduct numerical experiments on large-scale NASDAQ market microstructure datasets and report significant improvements over baselines on trading costs. Lin & Beling (2020a) similarly apply a variant of DQN to OE. The main novelty is to propose a shaped reward structure and incorporate the zero-end inventory constraint into DQN. Backtesting shows the DQN-based approach outperforms a few baselines, and the constraint improves the stability. Lin & Beling (2020b) further design an E2E framework utilizing LSTM to automatically extract features from the time-dependent limit-order book data, demonstrating improvements over previous methods. Similar ideas are studied by Deng et al. (2016), by integrating RNN and PG. In financial problems, the data are typically noisy with imperfect market information. To overcome this bottleneck, Fang et al. (2021) propose an oracle policy distillation approach: During training, a teacher policy is learned with access to the future price and volume, which is then distilled to learn the student policy, which has no access to future information and is the one to be deployed. Such a procedure reduces the training variance and leads to a better policy in testing, as demonstrated via experiments over a few baselines.

As discussed in previous sections, one prominent limitation of standard RL formulations is that the objective ignores the risk, which is of particular importance in OE and is also one of the main concerns of algorithmic trading. To fill the gap, Shen et al. (2014) adopt the risk-sensitive MDP formulation of Shen et al. (2013) to design a Q-learning-type algorithm. Specifically, the expected cumulative reward is replaced by a utility-based shortfall metric proposed in the mathematical finance literature. The backtesting results demonstrate improved robustness, especially during the 2010 flash crash.

3.4.2. Model-based reinforcement learning. The papers above rely on model-free RL and may suffer from low sample efficiency. Wei et al. (2019) first apply model-based DRL to OE: It first learns an environment model (simulator) using RNN and auto-encoder, and then trains a policy by running model-free DRL (including double DQN, PG, and A2C) within the simulator. One main contribution of the paper is to show the algorithm's five-day real performance in the real market, where the agent is profitable and the model-predicted trajectory is close to the observed trajectory. In contrast, Hambly et al. (2021) consider a structural model assumption, the linear–quadratic regulator problem. They use the OE problem as an example to demonstrate the robustness of running model-free methods in a model-based environment over directly solving the model-based controller, when there is model misspecification.

The model fidelity is typically central to the performance of model-based RL. Karpe et al. (2020) utilize a flexible agent-based market simulator, ABIDES (agent-based interactive discrete event simulation) (Byrd et al. 2020), to configure a multi-agent environment for training RL execution agents and show it converges to a time-weighted average price strategy.

3.4.2.1. Leveraging mathematical finance models. The RL methods surveyed above are largely disconnected from the mathematical finance literature and therefore may not utilize some domain-specific structure for more efficient learning. Meanwhile, Hendricks & Wilcox (2014) focus on optimal liquidation and leverage the Almgren–Chriss solution (Almgren & Chriss 2001) as a base algorithm: at every decision point, the authors first compute the recommended action based on the Almgren–Chriss model, then allow the RL agent to learn an action as a multiplicative factor on this base action based on the market microstructure state vector. With the additional flexibility, the RL agent successfully improves the shortfall by 10.3% compared with the Almgren–Chriss solution. One limitation of this analysis is that it misses comparisons with baseline RL agents.

3.4.2.2. Inverse reinforcement learning. The reward definition quality is also critical to the empirical performance of RL agents. The Roa-Vicens et al. (2019) study reveals the reward function of expert agents by inverse RL. They apply three inverse RL algorithms to simulated datasets to study their performance, and identify the failure of linear model-based inverse RL algorithms in recovering nonlinear reward functions.

4. META-ANALYSIS OF EXPERIMENTAL RESULTS

In this section, we conduct several meta-analyses on the effectiveness of RL in financial domains. Our objective is to explore four key research questions, which we believe can help analyze the effectiveness of RL methods in finance in current literature; discover challenges in RL applications (Section 6); and provide insights for future research directions (Section 7). In this section, we explore (*a*) the relationship between MDP design and model performance, (*b*) the influence of training duration on model performance, (*c*) the most prevalent RL algorithms and their relative performance, and (*d*) the suitability of current studies to the financial domain. To answer these questions, we collect a comprehensive set of literature summarized in **Supplemental Table 1** and analyze the experimental results reported in these papers.

Supplemental Material >

The varying use of test data and baselines in different research papers presents a significant challenge in conducting a meta-analysis. To address this, we propose the RL premium metric to benchmark model performance consistently across studies, providing insights into performance-influencing factors and guiding future research. Although it has broad applications across the financial field, it is particularly useful in portfolio management, where the Sharpe ratio is a standard evaluation metric. While alternative metrics like PnL exist for comparing model performance, the proposed RL premium metric offers distinct advantages in ensuring fair comparisons. By mitigating variations in time periods, markets, and transaction costs across different studies, our metric provides a more standardized basis for evaluation. The RL premium is calculated by subtracting the Sharpe ratio of the strongest non-RL baseline $\mathrm{SR}_{\mathrm{Baseline}}$ from the RL method's Sharpe ratio $\mathrm{SR}_{\mathrm{RL}}$, then normalizing this difference by $\mathrm{SR}_{\mathrm{Baseline}}$. Formally, the RL premium observed in a paper is defined as[1]

$$\text{RL premium} = (\mathrm{SR}_{\mathrm{RL}} - \mathrm{SR}_{\mathrm{Baseline}})/\mathrm{SR}_{\mathrm{Baseline}}.$$

For example, in the work of Jiang et al. (2017), the strongest non-RL baseline has a Sharpe ratio of 0.012, while the RL method has a Sharpe ratio of 0.087. The RL premium in this case would be calculated as $(0.087 - 0.012)/0.012 = 6.25$, indicating a significant performance gain from the use of the RL method.

4.1. Question 1: Does Markov Decision Process Design Affect Reinforcement Learning Performance?

We examine the relationship between the RL premium and the design of MDPs, specifically the state, action, and reward components.

[1] In portfolio management analysis, RL premiums that fall outside of the range between the 5% quantile and 95% quantile of the whole sample are treated as outliers and hence removed.

4.1.1. State. We observe a growing trend of integrating supplementary information beyond asset prices into the state space. For instance, Zhang et al. (2020) include technical indicators as part of the state for the RL agent, and Koratamaddi et al. (2021) consider market sentiment. To study whether including more features can help with model performance, we regress the RL premium on state feature dimension in **Figure 2*a***, with each experiment setting treated as separate data points. Findings suggest that most papers use two to three features, with only a few incorporating more. While the upward slope demonstrates that additional information may slightly improve portfolio management performance, the effect is not significant based on *p*-value.

4.1.2. Action. In multi-asset portfolio management tasks, the number of assets is related to the size of action space. As illustrated in **Figure 2*b***, RL models perform well when the action space is extensive. While optimizing numerous assets manually by human analysts can be a daunting task, RL models face a less arduous challenge in this regard.[2]

4.1.3. Reward. Designing a suitable reward is essential in RL modeling as it can greatly increase decision-making efficiency. In portfolio management, the most commonly used reward is the portfolio return. Innovative rewards such as risk-adjusted or sentiment-adjusted returns have been proposed in some studies. To understand the impact of rewards, we compare the RL premium using the original return and the shaped return. **Figure 2*c*** shows that the use of shaped rewards leads to a noticeable improvement in the performance of the model.

4.2. Question 2: How Does the Training Period Affect Model Performance?

The volatility of financial market data highlights the importance of carefully choosing the training period. Our analysis reveals that the length of the training period varies by study, indicating the absence of an established standard. We are interested in understanding whether a longer training period will improve or harm model performance. However, from the regression shown in **Figure 2*d***, we cannot conclude a relationship between the two. One possible explanation is that financial information is rapidly changing, so a lengthy training period may introduce noise rather than offering useful information for recent decisions.

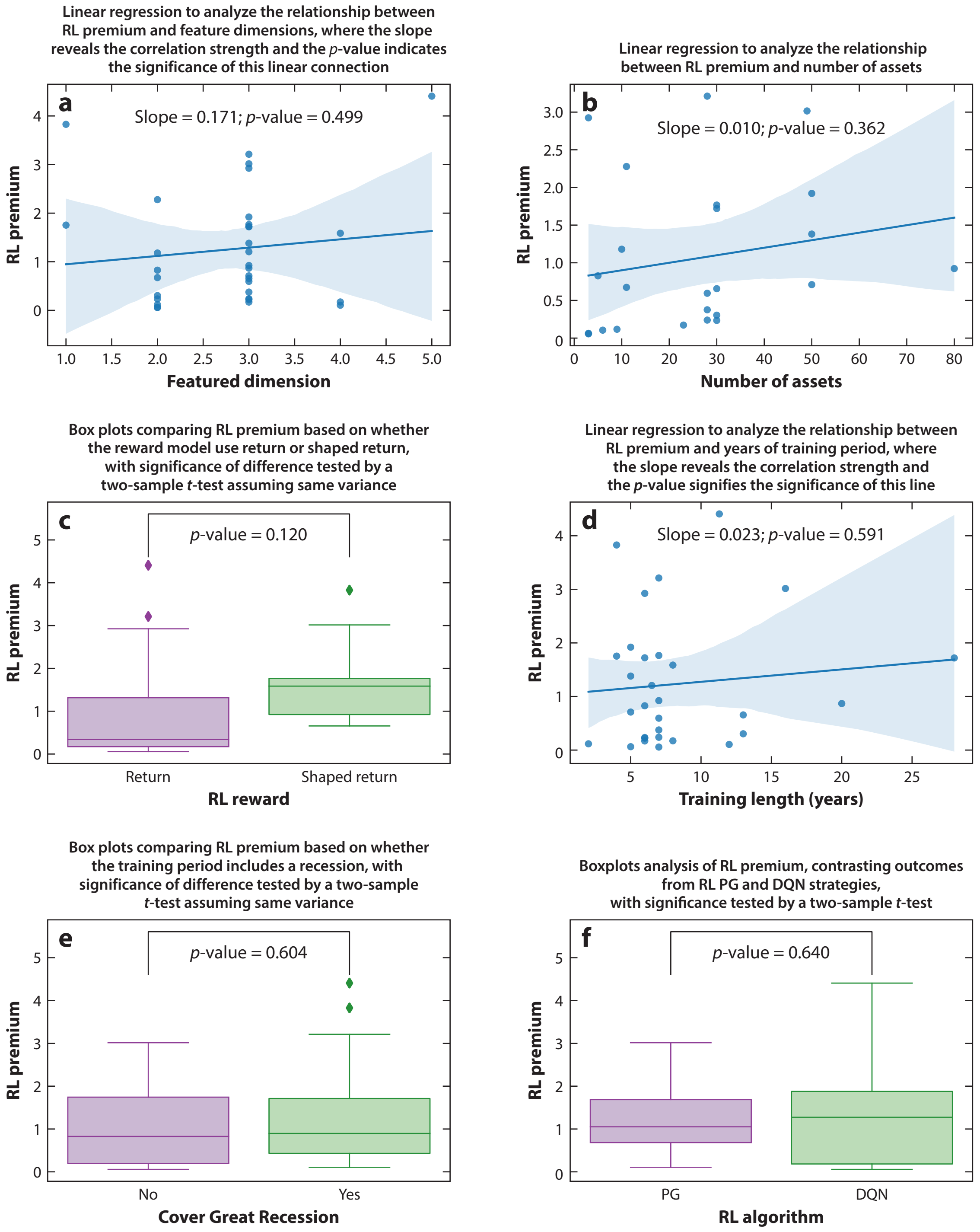
Including a recession period in training data poses challenges for RL models, as recession can lead to sudden and unpredictable market changes, rendering nonstationary conditions and hindering precise decision-making. However, statistical analysis leveraging a two-sample *t*-test to evaluate differences in the RL premium for periods covering a recession in **Figure 2*e*** shows that inclusion of a recession period does not appear to negatively affect the RL premium.

4.3. Question 3: Does the Choice of RL Algorithm Have Any Impact on Model Performance?

Supplemental Material >

Among all the articles we have reviewed, PG and DQN are the most popular RL algorithms in portfolio management, as shown in **Supplemental Table 2**. To compare the difference, we perform a two-sample *t*-test on the average RL premium between the RL methods PG and DQN. From **Figure 2*f***, no significant differences are detected between the two algorithms. In the literature on market making, the RL premium of three critic-only methods (Q-learning, SARSA, and R-learning) can be calculated. Further performance comparisons of these methods can be based on the RL premium calculation.

[2] To prevent skewness from affecting the analysis, studies with over 100 assets were excluded.



Linear regression to analyze the relationship between RL premium and feature dimensions, where the slope reveals the correlation strength and the *p*-value indicates the significance of this linear connection

Linear regression to analyze the relationship between RL premium and number of assets

Box plots comparing RL premium based on whether the reward model use return or shaped return, with significance of difference tested by a two-sample *t*-test assuming same variance

Linear regression to analyze the relationship between RL premium and years of training period, where the slope reveals the correlation strength and the *p*-value signifies the significance of this line

Box plots comparing RL premium based on whether the training period includes a recession, with significance of difference tested by a two-sample *t*-test assuming same variance

Boxplots analysis of RL premium, contrasting outcomes from RL PG and DQN strategies, with significance tested by a two-sample *t*-test

(Caption appears on following page)

Figure 2 (*Figure appears on preceding page*)

RL premium analysis. (*a*) Linear regression to analyze the relationship between the RL premium and feature dimensions. (*b*) Linear regression to analyze the relationship between the RL premium and number of assets. (*c*) Box plots comparing the RL premium based on whether the reward model uses return or shaped return. (*d*) Linear regression to analyze the relationship between the RL premium and years of training period. (*e*) Box plots comparing the RL premium based on whether the training period includes a recession. (*f*) Box plot analysis of the RL premium, contrasting outcomes from the RL PG and DQN strategies. Abbreviations: DQN, deep Q-network; PG, policy gradient; RL, reinforcement learning.

4.4. Question 4: Do the Assumptions Made in the Literature Accurately Reflect Reality?

In practical portfolio management, certain constraints, such as slippage and transaction costs, can negatively impact investment returns. Neglecting these real-world constraints may lead to an unrealistic evaluation of the performance of the RL model. We notice that most papers either assume zero slippage or do not discuss the treatment in their models (as shown in **Figure 3*a***). Regarding the transaction cost, the most common range used in the literature is 0.2–0.3% (as shown in **Figure 3*b***).

5. DISCUSSIONS ON ENVIRONMENTS, PACKAGES, AND BENCHMARKING

High-quality datasets/environments, rich open-source packages, and fair benchmarking are critical for the advance of an RL area. In Section 5.1, we give a detailed discussion of the training and evaluation environments used in RL for finance. Section 5.2 reviews major open-source packages and platforms, and discusses the benchmarking status.

5.1. Discussion on Training and Evaluation Environments

Interactive data collection is crucial for training or evaluating RL agents. This section covers various environments and their uses. Among all the papers we survey, the environments fall into three categories: (*a*) the real financial market, (*b*) a historical trajectories-based simulator with minimal assumptions, and (*c*) a simulator with a manually designed data generation mechanism. While real market data carry high requirements, we primarily focus on the latter two types. Nevertheless, incorporating more real-world evidence could significantly advance the field.

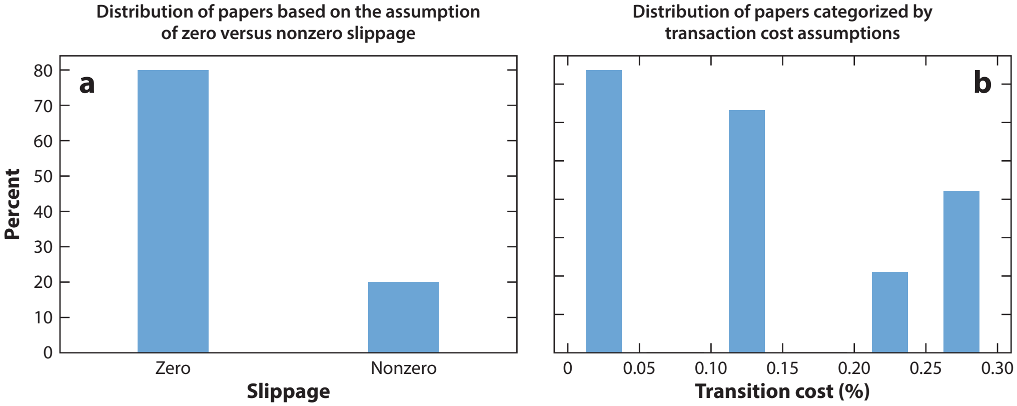


Figure 3

Analysis of realistic assumptions in surveyed papers. (*a*) Distribution of papers based on the assumption of zero versus nonzero slippage. (*b*) Distribution of papers categorized by transaction cost assumptions.

5.1.1. Historical trajectories-based simulator and exogenous Markov decision processes.

This section examines the assumptions behind replaying historical trajectories in RL and its technical validity. The exploration–exploitation trade-off, a core RL challenge, arises because actions not taken provide no outcomes (i.e., counterfactuals), making direct simulation from historical data without assumptions impossible. Fortunately, in MDPs with exogenous inputs (exo-MDP; Chitnis & Lozano-Pérez 2020), simulating by replaying historical trajectories turns out to be feasible. Specifically, suppose that the state variables can be decomposed into two parts as $s_t = (x_t, z_t)$, where (*a*) $\mathrm{P}(x_{t+1}|s_t, a_t) = \mathrm{P}(x_{t+1}|x_t)$ and (*b*) $z_{t+1} = f(s_t, a_t)$ with f as a known function. Then, at time t, even if we take an action a'_t in state s'_t and both are different from the recorded ones, setting $s'_{t+1} = (x_{t+1}, z'_{t+1})$ with $z'_{t+1} = f(x_t, z'_t, a'_t)$ is a valid way to sample the next state.

In the literature we reviewed, state variables in finance are often grouped into external information (like news or the day of the week), public market data (such as price or volume), and private agent states (like inventory levels). The market data are typically considered exogenous. The effect of an agent's actions on external information is termed the market impact. In many finance applications, exo-MDP is a reasonable assumption, hence providing the validity of the widely used backtesting-type protocol. For example, most portfolio optimization papers surveyed in Section 3.3 presume that only stock prices and side information are random, the agent's actions minimally affect the market, and state components related to the agent's own portfolio can be updated by accounting in a predetermined and deterministic manner. Therefore, the exo-MDP assumptions are satisfied, and we can use the historical trajectories to evaluate any policy, for either learning or evaluation purposes. There is minimal bias since this method does not require additional model assumptions like those needed in designed simulators.

However, exo-MDP may not be a suitable model in some other applications, such as OE (see Section 3.4), where the market impact is a central topic. Interestingly, many papers on OE overlook the market impact when using historical data for training or backtesting. We recommend a more thorough discussion and justification of using historical data in contexts like OE as a critical next step. Additionally, developing a high-fidelity simulator, similar to ABIDES (Byrd et al. 2020), would be highly beneficial for these studies.

To conclude, we refocus on the bandit algorithms for portfolio management discussed earlier in Section 3.3.1. In the surveyed literature, experiments utilize historical trajectories assuming an exogenous price process. Consequently, active exploration is unnecessary, suggesting that traditional online portfolio selection (Li & Hoi 2014) algorithms might suffice. However, the rationale for employing bandit algorithms in this context is not well articulated, and further in-depth discussions could clarify their relevance and utility.

5.1.2. Simulator with a manually designed data generation mechanism.

In applications where the assumption in Section 5.1.1 cannot be satisfied, or when high-quality data are not available, a simulator with a manually designed data generation mechanism is typically used. As widely acknowledged in the RL field, the simulator quality is often critical to the learning performance: A high-fidelity simulator can make the exploration much easier by generating more data when needed, while a simulator far from the real environment may cause a significant sim-to-real gap (Zhao et al. 2020), i.e., a learned policy that performs well in the simulation may fail once deployed in the real world.

In the finance domain, we observe two major types of data generation model: probabilistic model–based and agent-based. The former is developed on the basis of the rich mathematical finance literature (Horst 2018), e.g., by modeling the financial time series by some stochastic processes or other models from partial differential equations; the latter focuses on the fact that the finance market is composed of many trading agents who jointly determine the dynamic, and hence this approach models the trading behavior of these agents (Byrd et al. 2020). Both approaches are popular and sometimes even used together.

Despite their importance, high-fidelity simulators have not been publicly available in previous years, possibly due to the sensitivity and privacy issues in the finance domain. The lack of a fair evaluation and comparison benchmark has largely impeded the development of the area. Fortunately, in recent years, there has been increasing attention to this issue, with several benchmark environments published (Byrd et al. 2020, Ardon et al. 2021, Liu et al. 2022a). We do believe that these advances are as valuable and critical as those in methodology innovation.

5.2. Open-Source Packages and Benchmarking

In **Figure 1**, we summarize the frequency of different RL algorithms used in finance. Although there are works comparing different RL algorithms under the same settings (e.g., Spooner et al. 2018), there are still no comprehensive comparisons of common model-free algorithms (actor-only, critic-only, and actor–critic). Among the papers we have reviewed, different works adapt different MDP processes (either generated by a simulator or from real data) and RL approaches. Thus, to obtain a benchmark of different RL algorithms, a unified simulator is required. In the FinRL tool developed by Liu et al. (2022a), many different RL algorithms are implemented in the same environment for portfolio management, and such benchmarking can be achieved. However, in other areas such as market making, different tasks still adapt different price-generating processes and metrics. Specifically, most market making papers report their PnL, but the scale of that is not unified, as discussed in the definition of the RL premium. Thus, a unified benchmarking study is needed to compare different RL algorithms proposed by different papers.

To implement popular RL algorithms, open-source packages such as RLlib (Liang et al. 2018), Stable-baselines (Raffin et al. 2021), and TensorFlow agents (Hafner et al. 2018) can be used. However, to implement and adapt them to the finance domain, an E2E pipeline and platform are key, but could also be time-consuming and a major factor contributing to the inconsistency in settings between research papers. Several open-source packages have been contributed to this purpose, such as FinRL (Liu et al. 2021, 2022a,c) and TensorTrader (King 2024).

6. CHALLENGES

Despite recent advances in RL methods in many finance areas (Hambly et al. 2023), there are still many challenges to designing a practical RL system in the real world. Some of these challenges come from the nature of financial data, and others are due to the limitations of RL algorithms. We discuss these two aspects in Sections 6.1 and 6.2, respectively.

6.1. Challenges Due to Peculiar Features of Financial Data

In finance, RL tasks require special consideration due to the distinct characteristics of financial data, as highlighted by Fan & Yao (2017): high volatility, heavy-tailed distributions, nonstationarity, and asymmetry.

6.1.1. High volatility. In finance, high volatility refers to significant fluctuations in asset prices. It is often measured by the standard deviation of an asset's price over time or by its beta value, as described by Lakonishok & Shapiro (1984), which compares the stock's volatility to the overall market.

One approach to dealing with stock volatility in RL is to proceed with low- and high-volatility stocks separately, as suggested by Jin & El-Saawy (2016). The authors assume prior knowledge of which stocks are high and low volatility, and their approach prevents the agent from trading both types simultaneously, potentially boosting Sharpe ratio and reducing variance. However, solely relying on known volatility levels may be inadequate, necessitating the prediction of future volatility. While ML offers methods for volatility forecasting (for an overview see, e.g., Ge et al. 2022), integrating them into RL in finance remains largely unexplored.

6.1.2. Heavy-tailed distributions. The tail of the probability distribution of stock returns (defined as the ratio of price change over the initial price) is typically heavier than that of a normal distribution (Bradley & Taqqu 2003). For instance, Fan & Yao (2017) provide examples of some stock returns in the S&P 500 whose distribution tails are heavier than a t-distribution with degree 5, indicating that events far from the distribution mean may occur more frequently than expected under a normal distribution. In the context of RL, this phenomenon corresponds to cases where the reward distribution has a heavy tail, as the reward is calculated based on stock returns.

Although the heavy-tail issue has long been observed in financial data, in the RL literature, researchers still typically assume a light-tailed distribution, such as the Gaussian distribution, for rewards. To our knowledge, only Zhuang & Sui (2021) have designed RL algorithms for the heavy-tailed reward distribution setting. They propose a Q-learning algorithm that can achieve near-optimal regret bounds in heavy-tailed settings. Moreover, they have shown that learning the reward distribution is even more challenging than learning the state transition in such a setting. Since this paper is relatively recent, RL algorithms that are robust to heavy-tailed distributions have not yet been widely applied in finance applications.

6.1.3. Non-stationarity. The empirical validity of most state-of-the-art RL algorithms heavily depends on the stationarity of the environment—in other words, the assumption that the system transition and the reward function are stable (Li et al. 2024). However, the nonstationarity of the finance market has been widely acknowledged (Fan & Yao 2017). For example, the market dynamic during the financial recession in 2008 was found to be very different from that in the normal period (Guharay et al. 2013); therefore, including this period in training may affect the performance of the tests in normal periods, as discussed in our meta-analysis (see Section 4).

To the best of our knowledge, no work on finance RL has studied the nonstationarity issue formally. The training period is typically picked based on experience or data availability, without much justification. We believe that careful handling of this issue, for example, by testing stationarity and detecting change points (Li et al. 2024), can further improve the performance of RL in finance. A stronger connection to the vast literature on nonstationarity in financial time series would also be valuable (Schmitt et al. 2013).

6.1.4. Long-range dependency and latent information. Another critical assumption of most RL algorithms is the Markovian assumption, or roughly speaking, that our state variable contains sufficient historical information to predict the state transition. Without such an assumption, the foundation of these RL algorithms is shaken, which can lead to deterioration of their performance (Shi et al. 2020). Validating or satisfying this condition is particularly challenging in finance due to the market's complex dynamics and the lack of access to critical but unobservable information. Moreover, it is widely observed that the financial time series has long-range dependency (Fan & Yao 2017).

Based on our observation, the selection of state variables in financial RL works is largely based on the researchers' personal experience alone, without much explanation and validation. One common practice is to include a long time range of past data points, with the motivation to satisfy the Markovian assumption to a certain extent. For example, Lin & Beling (2020a,b) and Deng et al. (2016) use LSTM or RNN to learn from past time points. However, including a high-dimensional vector will make learning harder due to the introduced noise. Among the previously cited works, Fang et al. (2021) propose an oracle policy distillation approach and Liu et al. (2020) adopt partially observable Markov decision process algorithms to partially remedy information imperfectness. We suggest that one valuable future direction is to investigate which variables should be included in the state space. The Markovian assumption testing procedure and the variable selection procedure developed for RL in Shi et al. (2020) would be very useful.

6.2. Challenges Due to Limitations of Reinforcement Learning Algorithms

Another set of challenges arises from the inherent constraints of RL methods. These intrinsic limitations include the lack of interpretability, issues with MDP modeling, and the models' lack of robustness and generalizability.

6.2.1. Explainability. Obtaining insightful interpretations from DRL methods presents significant challenges. As Israel et al. (2020) point out, asset managers, driven by fiduciary duty, prefer interpretable and transparent models to communicate risks to clients. Therefore, the choice between interpretability and predictive power becomes a business decision for them. Recently, interpretable RL methods (Milani et al. 2024) have gained traction, suggesting a promising avenue for research in finance.

Supplemental Material >

6.2.2. Markov decision process modeling. The challenge of modeling a finance application as an MDP comes from two major aspects: first, how to select the variables considered in the states $\mathcal{S}$, and second, how to decide the reward function $\mathcal{R}$. Potential variables in financial data may include inventory, executed orders, historical prices, high-frequency data, and even news from the real world (**Supplemental Table 1**). It is not realistic to fit all factors into the state due to the curse of dimensionality (Bellman 1966). However, enough information should be included in the MDP to make sure that RL can accurately predict expected rewards or/and next states given an action. The reward functions vary in different financial applications. Even in the same financial application, no universal reward function is used in different papers. One reason is that in real-world financial applications, many metrics should be considered, including but not limited to profit, market impact, and regulators' requirements. Therefore, it is challenging to combine the different types of outcomes into a single reward function.

6.2.3. Robustness. Robustness, which indicates how stable the model performance could be under different conditions, is one of the main focuses of modeling. As discussed by Israel et al. (2020), the difficulties of using advanced models in financial markets, particularly in portfolio management, are largely due to the small data size and low signal-to-noise ratio. However, very limited attention has been paid to robustness. R. Wang et al. (2021) demonstrate the robustness of a model by assessing its performance across various markets and time periods. In general, we could not find a well-structured framework to measure and improve robustness in existing financial RL works.

7. FUTURE DIRECTIONS

Application of RL in finance has been an active research area in recent years. In this section, we discuss some future directions in this area.

7.1. Multi-Agent Reinforcement Learning

Most reviewed articles use a single RL agent in financial applications. However, Spooner & Savani (2020) and Gašperov & Kostanjčar (2021) suggest that incorporating adversarial agents can enhance the robustness of the RL agent. This adversarial setting has also been explored in robotics and card games (Heinrich & Silver 2016, Pinto et al. 2017). Patel (2018) introduces a multi-agent framework with hierarchical policies, which is also applicable in complex RL problems beyond finance, such as navigation and locomotion (Nachum et al. 2018). Despite these insights, multi-agent RL papers are a small portion of the reviewed works, indicating a need for further exploration.

7.2. Model-Based Reinforcement Learning

As is pointed out by Wei et al. (2019), there is an inherent connection between the model-based RL method and electronic trading systems, since electronic trading systems usually include a component to simulate the market. Therefore, it is very natural to use model-based RL algorithms that learn a simulator for the environment. However, as shown in **Figure 1**, few articles on OE use model-based RL. More research is needed to apply model-based RL algorithms across various finance fields.

7.3. Offline Reinforcement Learning

All surveyed works use online RL algorithms, which rely on interactions with an environment to generate new trajectories and improve the agent. In contrast, offline RL (Levine et al. 2020) learns directly from historical trajectories, without the need for new interactions. Offline RL has gained attention recently due to its safety and practicality in applications where online interaction is risky or infeasible. Moreover, Section 5.1.1 suggests that using historical data as a simulator for online RL may be incorrect in settings with market impacts; in contrast, offline RL is designed to handle these counterfactual issues. Therefore, offline RL is a promising direction for further research.

7.4. Risk-Sensitive Reinforcement Learning

Risk in investment refers to the uncertainty of capital loss. In portfolio management, many of the papers we reviewed incorporate the Sharpe ratio in their reward functions to account for the volatility of return. In market making, inventory penalties are added to the reward functions for reducing risks. In summary, most reviewed papers of RL in finance still use traditional RL algorithms and address risk through reward function design. However, in some RL algorithms, risk is managed via the design of the RL algorithms themselves, such as the risk-averse Q-learning method by Shen et al. (2014). Future works could incorporate risk-sensitive RL algorithms such as those of Mihatsch & Neuneier (2002) and Shen et al. (2014).

8. DISCLAIMER

It is important to note that the results of the quantitative meta-analysis presented in this article are intended to provide some general insight into the field of RL in finance. However, the results obtained are limited by the availability of data and variations in the settings used in each study. Furthermore, measurement of the relationship between model performance and factors only captures the impact of each factor independently and does not take into account potential confounders that may have influenced the outcome. Although the findings may be helpful in understanding the current stage of this research topic, the conclusions drawn from this study should be interpreted with caution and should not be considered definitive or conclusive.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review. This work does not relate to the positions of R.S., S.Z., and R.W. at Amazon.

LITERATURE CITED

Almgren R, Chriss N. 2001. Optimal execution of portfolio transactions. *J. Risk* 3:5–40

Ardon L, Vadori N, Spooner T, Xu M, Vann J, Ganesh S. 2021. Towards a fully RL-based market simulator. In *ICAIF '21: Proceedings of the Second ACM International Conference on AI in Finance*, pp. 1–9. New York: ACM

Avellaneda M, Stoikov S. 2008. High-frequency trading in a limit order book. *Quant. Finance* 8(3):217–24

Bellman R. 1966. Dynamic programming. *Science* 153(3731):34–37

Benhamou E, Saltiel D, Ohana JJ, Atif J. 2021. Detecting and adapting to crisis pattern with context based deep reinforcement learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 10050–57. Piscataway, NJ: IEEE

Bertsimas D, Lo AW. 1998. Optimal control of execution costs. *J. Financ. Markets* 1(1):1–50

Bradley BO, Taqqu MS. 2003. Financial risk and heavy tails. In *Handbook of Heavy Tailed Distributions in Finance*, ed. ST Rachev, pp. 35–103. Amsterdam: Elsevier

Byrd D, Hybinette M, Balch TH. 2020. ABIDES: towards high-fidelity multi-agent market simulation. In *Proceedings of the 2020 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, pp. 11–22. New York: ACM

Chakraborty B, Murphy SA. 2014. Dynamic treatment regimes. *Annu. Rev. Stat. Appl.* 1:447–64

Chan NT, Shelton C. 2001. *An electronic market-maker*. Rep., Artif. Intell. Lab., MIT, Cambridge, MA

Charpentier A, Elie R, Remlinger C. 2021. Reinforcement learning in economics and finance. *Comput. Econ.* 62:425–62

Chitnis R, Lozano-Pérez T. 2020. Learning compact models for planning with exogenous processes. *Proc. Mach. Learn. Res.* 100:813–22

Deng Y, Bao F, Kong Y, Ren Z, Dai Q. 2016. Deep direct reinforcement learning for financial signal representation and trading. *IEEE Trans. Neural Netw. Learn. Syst.* 28(3):653–64

Fama EF. 1970. Efficient capital markets: a review of theory and empirical work. *J. Finance* 25(2):383–417

Fan J, Yao Q. 2017. *The Elements of Financial Econometrics*. Cambridge, UK: Cambridge Univ. Press

Fang Y, Ren K, Liu W, Zhou D, Zhang W, et al. 2021. Universal trading for order execution with oracle policy distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 107–15. Washington, DC: AAAI

Fischer TG. 2018. *Reinforcement learning in financial markets–a survey*. Tech. Rep., FAU Discussion Papers in Economics, Friedrich-Alexander-Univ., Nürnberg, Ger.

Ganesh S, Vadori N, Xu M, Zheng H, Reddy P, Veloso M. 2019. Reinforcement learning for market making in a multi-agent dealer market. arXiv:1911.05892 [q-fin.TR]

Gao Z, Gao Y, Hu Y, Jiang Z, Su J. 2020. Application of deep Q-network in portfolio management. In *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*, pp. 268–75. Piscataway, NJ: IEEE

Gašperov B, Begušić S, Posedel Šimović P, Kostanjčar Z. 2021. Reinforcement learning approaches to optimal market making. *Mathematics* 9(21):2689

Gašperov B, Kostanjčar Z. 2021. Market making with signals through deep reinforcement learning. *IEEE Access* 9:61611–22

Ge W, Lalbakhsh P, Isai L, Lenskiy A, Suominen H. 2022. Neural network–based financial volatility forecasting: a systematic review. *ACM Comput. Surv.* 55(1):14

Guéant O, Manziuk I. 2019. Deep reinforcement learning for market making in corporate bonds: beating the curse of dimensionality. *Appl. Math. Finance* 26(5):387–452

Guharay SK, Thakur GS, Goodman FJ, Rosen SL, Houser D. 2013. Analysis of non-stationary dynamics in the financial system. *Econ. Lett.* 121(3):454–57

Hafner D, Davidson J, Vanhoucke V. 2018. TensorFlow agents: Efficient batched reinforcement learning in TensorFlow. arXiv:1709.02878 [cs.LG]

Hambly B, Xu R, Yang H. 2021. Policy gradient methods for the noisy linear quadratic regulator over a finite horizon. *SIAM J. Control Optim.* 59(5):3359–91

Hambly B, Xu R, Yang H. 2023. Recent advances in reinforcement learning in finance. *Math. Finance* 33(3):437–503

Heinrich J, Silver D. 2016. Deep reinforcement learning from self-play in imperfect-information games. arXiv:1603.01121 [cs.LG]

Hendricks D, Wilcox D. 2014. A reinforcement learning extension to the Almgren-Chriss framework for optimal trade execution. In *2014 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pp. 457–64. Piscataway, NJ: IEEE

Horst U. 2018. *Introduction to Mathematical Finance*. Berlin: Humboldt Univ.

Huang J, Chai J, Cho S. 2020. Deep learning in finance and banking: a literature review and classification. *Front. Bus. Res. China* 14:13

Huo X, Fu F. 2017. Risk-aware multi-armed bandit problem with application to portfolio selection. *R. Soc. Open Sci.* 4(11):171377

Israel R, Kelly BT, Moskowitz TJ. 2020. Can machines "learn" finance? *J. Invest. Manag.* 18(2):23–36

Jegadeesh N, Titman S. 1993. Returns to buying winners and selling losers: implications for stock market efficiency. *J. Finance* 48:65–91

Jiang Z, Xu D, Liang J. 2017. A deep reinforcement learning framework for the financial portfolio management problem. arXiv:1706.10059 [q-fin.CP]

Jin O, El-Saawy H. 2016. *Portfolio management using reinforcement learning*. Work. Pap., Dep. Stat., Stanford Univ., Stanford, CA

Kalman RE. 1960. A new approach to linear filtering and prediction problems. *J. Basic Eng.* 82(1):35–45

Karpe M, Fang J, Ma Z, Wang C. 2020. Multi-agent reinforcement learning in a realistic limit order book market simulation. In *ICAIF '20: Proceedings of the First ACM International Conference on AI in Finance*, art. 30. New York: ACM

King A. 2024. TensorTrade: trade efficiently with reinforcement learning. *Statistical Software*, version 1.0.3. **https://github.com/tensortrade-org/tensortrade?tab=readme-ov-file**

Koratamaddi P, Wadhwani K, Gupta M, Sanjeevi SG. 2021. Market sentiment-aware deep reinforcement learning approach for stock portfolio allocation. *Eng. Sci. Technol. Int. J.* 24(4):848–59

Kosorok MR, Laber EB. 2019. Precision medicine. *Annu. Rev. Stat. Appl.* 6:263–86

Lakonishok J, Shapiro AC. 1984. Stock returns, beta, variance and size: an empirical analysis. *Financ. Anal. J.* 40(4):36–41

Lattimore T, Szepesvári C. 2020. *Bandit Algorithms*. Cambridge, UK: Cambridge Univ. Press

Lee J, Kim R, Yi SW, Kang J. 2020. MAPS: multi-agent reinforcement learning-based portfolio management system. arXiv:2007.05402 [cs.AI]

Lei K, Zhang B, Li Y, Yang M, Shen Y. 2020. Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading. *Expert Syst. Appl.* 140:112872

Levine S, Kumar A, Tucker G, Fu J. 2020. Offline reinforcement learning: tutorial, review, and perspectives on open problems. arXiv:2005.01643 [cs.LG]

Li B, Hoi SC. 2014. Online portfolio selection: a survey. *ACM Comput. Surv.* 46(3):35

Li M, Shi C, Wu Z, Fryzlewicz P. 2024. Testing stationarity and change point detection in reinforcement learning. arXiv:2203.01707 [stat.ML]

Li X, Li Y, Zhan Y, Liu XY. 2019. Optimistic bull or pessimistic bear: adaptive deep reinforcement learning for stock portfolio allocation. arXiv:1907.01503 [q-fin.ST]

Liang E, Liaw R, Nishihara R, Moritz P, Fox R, et al. 2018. RLlib: Abstractions for distributed reinforcement learning. *Proc. Mach. Learn. Res.* 80:3053–62

Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, et al. 2019. Continuous control with deep reinforcement learning. arXiv:1509.02971 [cs.LG]

Lin S, Beling PA. 2020a. A deep reinforcement learning framework for optimal trade execution. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, ed. Y Dong, G Ifrim, D Mladenić, C Saunders, S Van Hoecke, pp. 223–40. New York: Springer

Lin S, Beling PA. 2020b. An end-to-end optimal trade execution framework based on proximal policy optimization. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, ed. C Bessiere, pp. 4548–54. Palo Alto, CA: AAAI

Liu XY, Rui J, Gao J, Yang L, Yang H, et al. 2022a. FinRL-Meta: a universe of near-real market environments for data-driven deep reinforcement learning in quantitative finance. arXiv:2112.06753 [q-fin.TR]

Liu XY, Xiong Z, Zhong S, Yang H, Walid A. 2022b. Practical deep reinforcement learning approach for stock trading. arXiv:1811.07522 [cs.LG]

Liu XY, Yang H, Chen Q, Zhang R, Yang L, et al. 2022c. FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance. arXiv:2011.09607 [q-fin.TR]

Liu XY, Yang H, Gao J, Wang CD. 2021. FinRL: deep reinforcement learning framework to automate trading in quantitative finance. In *ICAIF '21: Proceedings of the Second ACM International Conference on AI in Finance*, art. 1. New York: ACM

Liu Y, Liu Q, Zhao H, Pan Z, Liu C. 2020. Adaptive quantitative trading: an imitative deep reinforcement learning approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2128–35. Washington, DC: AAAI

Ma C, Zhang J, Liu J, Ji L, Gao F. 2021. A parallel multi-module deep reinforcement learning algorithm for stock trading. *Neurocomputing* 449:290–302

Mani M, Phelps S, Parsons S. 2019. Applications of reinforcement learning in automated market-making. In *Proceedings of the GAIW: Games, Agents and Incentives Workshops, Montreal, Canada*, pp. 13–14. N.p.: IFAAMS

Markowitz H. 1952. Portfolio selection. *J. Finance* 7(1):77–91

Menezes CF, Hanson DL. 1970. On the theory of risk aversion. *Int. Econ. Rev.* 11(3):481–87

Meng TL, Khushi M. 2019. Reinforcement learning in financial markets. *Data* 4(3):110

Mihatsch O, Neuneier R. 2002. Risk-sensitive reinforcement learning. *Mach. Learn.* 49(2):267–90

Milani S, Topin N, Veloso M, Fang F. 2024. Explainable reinforcement learning: a survey and comparative review. *ACM Comput. Surv.* 56(7):168

Millea A. 2021. Deep reinforcement learning for trading—a critical survey. *Data* 6(11):119

Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, et al. 2016. Asynchronous methods for deep reinforcement learning. *Proc. Mach. Learn. Res.* 48:1928–37

Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, et al. 2013. Playing Atari with deep reinforcement learning. arXiv:1312.5602 [cs.LG]

Mosavi A, Faghan Y, Ghamisi P, Duan P, Ardabili SF, et al. 2020. Comprehensive review of deep reinforcement learning methods and applications in economics. *Mathematics* 8(10):1640

Murphy SA. 2003. Optimal dynamic treatment regimes. *J. R. Stat. Soc. Ser. B* 65(2):331–66

Nachum O, Gu SS, Lee H, Levine S. 2018. Data-efficient hierarchical reinforcement learning. In *NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ed. S Bengio, HM Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, pp. 3307–17. Red Hook, NY: Curran

Nash JF Jr. 1950. Equilibrium points in *n*-person games. *PNAS* 36(1):48–49

Nevmyvaka Y, Feng Y, Kearns M. 2006. Reinforcement learning for optimized trade execution. In *ICML '06: Proceedings of the 23rd International Conference on Machine Learning*, pp. 673–80. New York: ACM

O'Hara M, Oldfield GS. 1986. The microeconomics of market making. *J. Financ. Quant. Anal.* 21(4):361–76

Patel Y. 2018. Optimizing market making using multi-agent reinforcement learning. arXiv:1812.10252 [q-fin.TR]

Pinto L, Davidson J, Sukthankar R, Gupta A. 2017. Robust adversarial reinforcement learning. *Proc. Mach. Learn. Res.* 70:2817–26

Pratt JW. 1978. Risk aversion in the small and in the large. In *Uncertainty in Economics*, ed. P Diamond, M Rothschild, pp. 59–79. Amsterdam: Elsevier

Pricope TV. 2021. Deep reinforcement learning in quantitative algorithmic trading: a review. arXiv:2106.00123 [cs.LG]

Raffin A, Hill A, Gleave A, Kanervisto A, Ernestus M, Dormann N. 2021. Stable-Baselines3: reliable reinforcement learning implementations. *J. Mach. Learn. Res.* 22(268):1–8

Rescorla RA. 1972. A theory of Pavlovian conditioning: variations in the effectiveness of reinforcement and non-reinforcement. In *Classical Conditioning II: Current Research and Theory*, ed. AH Black, WF Prokasy, pp. 64–69. New York: Appleton-Century-Crofts

Roa-Vicens J, Chtourou C, Filos A, Rullan F, Gal Y, Silva R. 2019. Towards inverse reinforcement learning for limit order book dynamics. arXiv:1906.04813 [cs.LG]

Russo DJ, Van Roy B, Kazerouni A, Osband I, Wen Z, et al. 2018. A tutorial on Thompson sampling. *Found. Trends Mach. Learn.* 11(1):1–96

Schaul T, Quan J, Antonoglou I, Silver D. 2016. Prioritized experience replay. arXiv:1511.05952 [cs.LG]

Schmitt TA, Chetalova D, Schäfer R, Guhr T. 2013. Non-stationarity in financial time series: generic features and tail behavior. *Europhys. Lett.* 103(5):58003

Schwartz A. 1993. A reinforcement learning method for maximizing undiscounted rewards. In *ICML'93: Proceedings of the Tenth International Conference on International Conference on Machine Learning*, pp. 298–305. San Francisco, CA: Morgan Kaufmann

Selser M, Kreiner J, Maurette M. 2021. Optimal market making by reinforcement learning. arXiv:2104.04036 [cs.LG]

Shen W, Wang J. 2016. Portfolio blending via Thompson sampling. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, ed. S Kambhampati, pp. 1983–89. Palo Alto, CA: AAAI

Shen W, Wang J, Jiang YG, Zha H. 2015. Portfolio choices with orthogonal bandit learning. In *IJCAI'15: Proceedings of the 24th International Conference on Artificial Intelligence*, ed. Q Yang, M Wooldridge, pp. 974–80. Palo Alto, CA: AAAI

Shen Y, Huang R, Yan C, Obermayer K. 2014. Risk-averse reinforcement learning for algorithmic trading. In *2014 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pp. 391–98. Piscataway, NJ: IEEE

Shen Y, Stannat W, Obermayer K. 2013. Risk-sensitive Markov control processes. *SIAM J. Control Optim.* 51(5):3652–72

Shi C, Wan R, Song R, Lu W, Leng L. 2020. Does the Markov decision process fit the data: testing for the Markov property in sequential decision making. *Proc. Mach. Learn. Res.* 119:8807–17

Soleymani F, Paquet E. 2021. Deep graph convolutional reinforcement learning for financial portfolio management—DeepPocket. *Expert Syst. Appl.* 182:115127

Spooner T, Fearnley J, Savani R, Koukorinis A. 2018. Market making via reinforcement learning. In *AAMAS '18: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 434–42. Richland, SC: Int. Found. Auton. Agents Multiagent Syst.

Spooner T, Savani R. 2020. Robust market making via adversarial reinforcement learning. arXiv:2003.01820 [q-fin.TR]

Such FP, Madhavan V, Conti E, Lehman J, Stanley KO, Clune J. 2017. Deep neuroevolution: genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. arXiv:1712.06567 [cs.NE]

Sutton RS, Barto AG. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press

Sutton RS, McAllester D, Singh S, Mansour Y. 1999. Policy gradient methods for reinforcement learning with function approximation. In *NIPS'99: Proceedings of the 12th International Conference on Neural Information Processing Systems*, ed. SA Solla, TK Leen, K Müller, pp. 1057–63. Cambridge, MA: MIT Press

Théate T, Ernst D. 2021. An application of deep reinforcement learning to algorithmic trading. *Expert Syst. Appl.* 173:114632

Thomas P. 2014. Bias in natural actor-critic algorithms. *Proc. Mach. Learn. Res.* 32(1):441–48

Wang J, Zhang Y, Tang K, Wu J, Xiong Z. 2019. AlphaStock: a buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks. In *KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1900–8. New York: ACM

Wang R, Wei H, An B, Feng Z, Yao J. 2021. Commission fee is not enough: a hierarchical reinforced framework for portfolio management. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, pp. 626–33. Palo Alto, CA: AAAI

Wang Z, Huang B, Tu S, Zhang K, Xu L. 2021. DeepTrader: a deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, pp. 643–50. Palo Alto, CA: AAAI

Wang Z, Schaul T, Hessel M, Hasselt H, Lanctot M, Freitas N. 2016. Dueling network architectures for deep reinforcement learning. *Proc. Mach. Learn. Res.* 48:1995–2003

Watkins CJ, Dayan P. 1992. Q-learning. *Mach. Learn.* 8:279–92

Wei H, Wang Y, Mangu L, Decker K. 2019. Model-based reinforcement learning for predictions and control for limit order books. arXiv:1910.03743 [cs.AI]

Yang H, Liu XY, Zhong S, Walid A. 2020. Deep reinforcement learning for automated stock trading: An ensemble strategy. In *ICAIF '20: Proceedings of the First ACM International Conference on AI in Finance*, art. 31. New York: ACM

Ye Y, Pei H, Wang B, Chen PY, Zhu Y, et al. 2020. Reinforcement-learning based portfolio management with augmented asset movement prediction states. In *Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pp. 1112–19. Palo Alto, CA: AAAI

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

Zhang Z, Zohren S, Roberts S. 2020. Deep reinforcement learning for trading. *J. Financ. Data Sci.* 2(2):25–40

Zhao W, Queralta JP, Westerlund T. 2020. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 737–44. Piscataway, NJ: IEEE

Zhu M, Zheng X, Wang Y, Li Y, Liang Q. 2019. Adaptive portfolio by solving multi-armed bandit via Thompson sampling. arXiv:1911.05309 [cs.LG]

Zhuang V, Sui Y. 2021. No-regret reinforcement learning with heavy-tailed rewards. *Proc. Mach. Learn. Res.* 130:3385–93