Artificial Intelligence for Engineering

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY | IET The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

FinRL Contests: Data-Driven Financial Reinforcement Learning Agents for Stock and Crypto Trading

Keyi Wang[1] | Nikolaus Holzer[2] | Ziyi Xia[2] | Yupeng Cao[3] | Jiechao Gao[1,4] | Anwar Walid[1,5] | Kairong Xiao[6] | Xiao-Yang Liu Yanglet[1,5]

[1]SecureFinAI Lab, Columbia University, New York, New York, USA | [2]Department of Computer Science, Columbia University, New York, New York, USA | [3]Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, New Jersey, USA | [4]Center for SDGC, Stanford University, Stanford, California, USA | [5]Department of Electrical Engineering, Columbia University, New York, New York, USA | [6]Columbia Business School, Columbia University, New York, New York, USA

Correspondence: Xiao-Yang Liu Yanglet (XL2427@columbia.edu)

Received: 7 July 2025 | **Revised:** 30 July 2025 | **Accepted:** 7 August 2025

Funding: Keyi Wang, Nikolaus Holzer, Jiechao Gao, Anwar Walid, and Xiao-Yang Liu Yanglet acknowledge the support from Columbia's SIRS and STAR Programme, as well as The Tang Family Fund for Research Innovations in FinTech, Engineering, and Business Operations. Xiao-Yang Liu Yanglet also acknowledges the support from a NSF IUCRC CRAFT centre research grant (CRAFT Grant 22017) for this research.

Keywords: benchmark | FinAI | financial engineering | financial reinforcement learning | FinRL | LLM

ABSTRACT

Financial reinforcement learning (FinRL) is now a practical paradigm for financial engineering. However, applying RL strategies to real-world trading tasks remains a challenge for individuals, as it is error-prone and engineering-heavy. The stock and crypto trading tasks pose unique challenges due to the non-stationarity of financial data, low signal-to-noise ratios and various market frictions. Although numerous FinRL methods have been developed for stock and crypto trading tasks, the lack of standardised task definitions, real-time high-quality datasets, close-to-real market environments and robust baselines has hindered consistent reproduction in both the open-source community and the FinTech industry. To bridge this gap, we organised a series of FinRL Contests from 2023 to 2025, covering a diverse range of financial tasks such as stock trading, order execution, crypto trading, and the use of large language model (LLM)-engineered signals. These contests attracted 230+ participants from 100+ institutions in 20+ countries. To encourage participation, we provided starter kits that feature GPU-optimised parallel market environments, ensemble learning methods, and comprehensive instructions. In this paper, we summarise these benchmarking efforts, detailing trading task formulations, data curation pipelines, environment implementations, evaluation protocols, participant performance and organisational insights. It guides our follow-up FinRL Contests and also provides a reference pipeline for FinAI contests alike.

1 | Introduction

Financial reinforcement learning (FinRL) [1–6] is an interdisciplinary field that applies reinforcement learning algorithms to financial tasks, such as algorithmic trading, portfolio management, option pricing, hedging and market making [7–11]. FinRL aims to train trading agents to interact with dynamic financial environments and to optimise their financial decisions autonomously. As FinRL methods continue to advance, two practical challenges have become increasingly apparent. One is policy instability, where small changes in training settings or market environments can lead to large variations in performance. The other is the sampling bottleneck, as collecting high-quality trajectories is often expensive and limited by access to financial data. Additionally, it remains difficult to compare the performance of these RL strategies due to inconsistencies in task

definitions, dataset choices, environment implementations and baselines. These challenges make it difficult to obtain reliable and consistent trading agents, which should be supported by systematic evaluation and reproducible benchmarks in both traditional stock markets and emerging crypto markets.

FinRL-Meta [5, 6] partially addressed this issue by creating standard gymnasium-style [12] market environments for different financial tasks. This framework builds an automated data curation pipeline that processes and loads dynamic market data into market environments. It also allows for direct plugging in RL agents to evaluate various trading tasks. However, there is no widely accepted benchmark for evaluating RL agents in trading scenarios [10], which hinders reproducibility and makes it difficult for the community to compare results and build on previous work. There is a strong need for an open and collaborative platform where researchers can systematically evaluate their trading agents under realistic financial conditions. To fill this gap, we initiated the FinRL Contests as a community-driven effort to promote reproducibility, transparency and benchmarking in financial reinforcement learning, particularly in both stock and crypto markets.

In this paper, we summarise the FinRL Contests 2023–2025, which are our benchmarking efforts for financial reinforcement learning. We describe the formulation of stock and crypto trading tasks as Markov decision processes (MDPs), the design of an automated data curation pipeline, the implementation of GPU-optimised parallel market environments and the establishment of standardised evaluation protocols. In addition, we provide a summary of the performance of the participants and the organisational experience gained from running these contests.

From 2023 to 2025, we have organised a series of FinRL Contests to support reproducible experimentation and systematic benchmarking of FinRL trading agents. These contests cover diverse tasks such as stock trading, crypto trading and LLM-engineered signals. The FinRL Contests attracted 230+ participants from 100+ institutions across 20+ countries. Figure 1 shows the participation of FinRL Contests. The map on the left is the geographical distribution of participants by country, with darker colours indicating more participants from that country. The map shows that most of the participants are from Asia, North America and Europe. For the 230+ participants, 70.5% come from academic institutions, 19.1% are individuals, and 10.6% come from industry. To ensure transparency and reproducibility, all participant teams were encouraged to open-source the solutions. To support the development of solutions for the FinRL Contest tasks, we provide starter kits for participants, including tutorials, market data, GPU-optimised parallel market environments and baseline methods. The full list of starter kits, contest websites and contest participation from 2023 to 2025 is summarised in Table 1. In addition, we continuously update the documentation, serving as a detailed guide and resource hub for participants. This documentation includes a detailed introduction, task descriptions and explanations of baseline solutions. As the contests continue, the FinRL community continues to attract students, researchers and practitioners who actively discuss and contribute on multiple platforms, including GitHub (15,000+ stars), Discord (550+ members), WeChat (600+ members), Google email groups (450+ members), and the

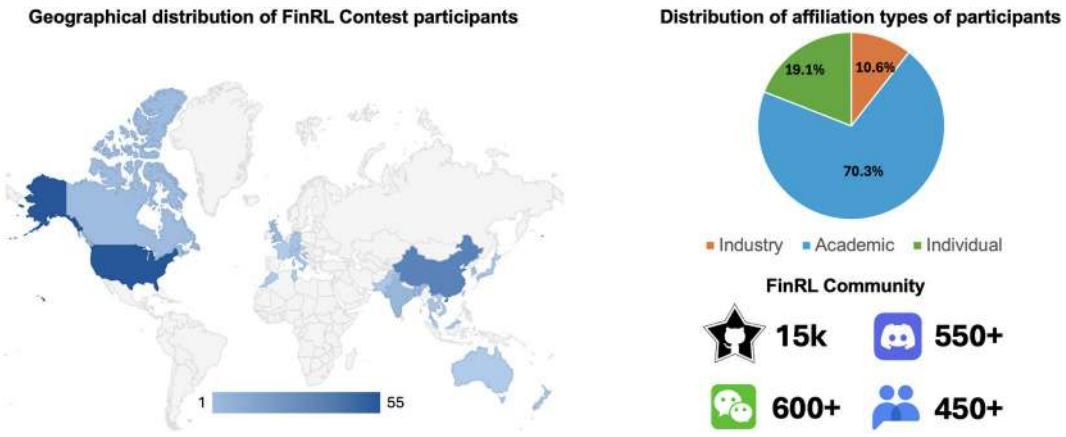


FIGURE 1 | Overview of the FinRL Contest participation and the FinRL community.

TABLE 1 | Links and participation of FinRL Contests 2023–2025.

FinRL Contests	Starter kit	Website	#Teams	#Participants
FinAI 2025 @ IEEE CSCloud 2025[a]	GitHub Repo 2025	Website 2025	30	36
FinRL 2025 @ IEEE IDS 2025	GitHub Repo 2025	Website 2025	85	120
FinRL 2024 @ ACM ICAIF 2024	GitHub Repo 2024	Website 2024	21	27
FinRL 2023 @ ACM ICAIF 2023	GitHub Repo 2023	Website 2023	46	53
Documentation		FinAI Contests documentation		

Note: FinAI has a broader scope than FinRL.
[a]The FinAI Contest 2025 @ IEEE CSCloud is ongoing and the statistics are reported on 27 July 2025.

FinRL Contest mailing list (300+ members), as shown in Figure 1. Through the repeated testing in contests, FinRL tasks have been verified and are being explored for real-world deployment. The FinRL market environments have become more stable and robust.

The remainder of this paper is organised as follows. Section 2 reviews related works. Section 3 describes tasks. Section 4 describes dynamic market data in FinRL. Section 5 describes the benchmarking pipeline. Section 6 shows performance. Section 7 summarises organisational aspects. Finally, we conclude this paper in Section 8.

2 | Related Work

2.1 | Applications of Financial Reinforcement Learning

Financial markets are inherently complex, dynamic and high-dimensional, making deep reinforcement learning (DRL) a compelling approach for sequential decision-making [7]. DRL has been successfully applied to a variety of financial tasks [9], including algorithmic trading, portfolio optimisation, and option pricing, where traditional rule-based or supervised methods often fall short in capturing long-term rewards under uncertainty. Beyond conventional reward-driven learning, recent advances in preference-based reinforcement learning offer alternative supervision signals. For example, direct preference optimisation (DPO) [13] formulates learning objectives based on pairwise comparisons between trajectories, enabling policy improvement in settings where explicit reward design is difficult or noisy. Such methods hold promise for financial applications that involve subjective goals, human feedback or implicit utility functions, and may further enhance the robustness and flexibility of DRL in real-world markets.

2.2 | Methods of Financial Reinforcement Learning

A wide range of reinforcement learning methods [8, 9] have been explored for financial decision-making. Among these, algorithms based on policy gradients and actor–critic architectures [14] have received particular attention due to their effectiveness in handling continuous control, high-dimensional action spaces and non-stationary dynamics. Notable examples include proximal policy optimisation (PPO), deep deterministic policy gradient (DDPG), soft actor-critic (SAC) and twin-delayed DDPG (TD3), which are frequently adopted for their robustness and sample efficiency. Ying et al. [15] proposed the CVaR proximal policy optimisation (CPPO) algorithm, which integrates conditional value-at-risk (CVaR) into its objective based on PPO.

Building on these algorithmic foundations, many studies have explored DRL applications in quantitative finance. For example, Deng et al. [16] propose a recurrent deep reinforcement learning framework that jointly learns market representations and trading policies. Their model demonstrates robust performance across stock and commodity markets by integrating deep feature extraction with sequential decision-making. Avramelou et al. [17] propose a multi-modal embedding approach to integrate both price and sentiment data in DRL-based trading agents. Their method improves trading performance while enabling dynamic adjustment of modality importance without retraining. To address the practical challenges of applying DRL in quantitative trading, Li et al. [4] propose FinRL-Podracer, a scalable cloud-based framework for efficient training and deployment of DRL agents. The system integrates high-performance GPU optimisation with automated training workflows, significantly improving both trading performance and development efficiency. Although recent studies have advanced DRL applications across various financial domains, systematic evaluation and comparison of different approaches remain an important direction, motivating further efforts towards benchmarking financial reinforcement learning.

2.3 | Benchmarks of Financial Reinforcement Learning

Recent research works have applied deep reinforcement learning (DRL) to quantitative finance [18–22] by developing customised market environments. Although these open-source libraries provide useful environments and datasets from sources such as Yahoo Finance and WRDS, there is still a lack of standardised benchmarks for systematic evaluation. FinRL [1–3] offers a full pipeline with environments for stock trading, portfolio allocation, and crypto trading; however, these environments are increasingly insufficient to meet the growing demands of the community. To address the challenges of data quality, survivorship bias and model overfitting, Liu et al. [5, 6] further introduced FinRL-Meta. FinRL-Meta follows a DataOps paradigm to automatically collect dynamic datasets and construct hundreds of gym-style market environments, reproduces popular DRL trading papers to facilitate research reproducibility, and supports cloud-based deployment and visualisation for community-driven performance evaluation. Despite these advances, establishing a unified, large-scale, and competitive benchmark for data-driven financial reinforcement learning remains an open direction, motivating the development of FinRL Contests.

3 | Tasks

Based on open-source projects FinRL [1–4] and FinRL-Meta [5, 6], and feedback from our open-source community, we select several stable and representative trading tasks for the FinRL Contests. In this section, we describe the tasks in the language of Markov Decision Processes (MDPs), discuss the parallelism in the estimate of policy gradients, and describe the specific tasks featured in the FinRL Contests.

3.1 | Tasks in FinRL Contests 2023–2025

We summarise trading tasks of FinRL Contests 2023–2025 in Figure 2, including daily stock trading with OHLCV data and

Key Components	Attributes	FinRL Contest 2023	FinRL Contest 2024		FinRL Contest 2025		FinAI Contest 2025
		Task 1 Data-Centric Stock Trading	Task 1 Crypto Trading with Ensemble Learning	Task 2 LLM-Engineered Signals with RLMF	Task 1 FinRL-DeepSeek for Stock Trading	Task 2 FinRL-AlphaSeek for Crypto Trading	Task 1 FinRL-DeepSeek for Crypto Trading
State	Balance; Shares	✓	✓	✓	✓	✓	✓
	Open/high/low/close/volume (OHLCV) data	✓		✓	✓		
	Limit order book (LOB) data		✓			✓	✓
	Technical indicators	✓	✓	✓	✓	✓	✓
	Fundamental indicators						
	Social data; Sentiment data			✓	✓		✓
	Smart beta indexes, etc.						
Action	Buy/Sell/Hold	✓	✓		✓	✓	✓
	Short/Long						
	Portfolio weights						
	Sentiment score			✓			
Rewards	Change of portfolio value	✓	✓		✓	✓	✓
	Portfolio log-return						
	Sharpe ratio						
	Penalize for high risk				✓		
	Market feedback			✓			

FIGURE 2 | Tasks in FinRL Contests 2023–2025.

second-level crypto trading with limit order book (LOB) data. The stock trading tasks include the following:

- **Data-Centric Stock Trading @ FinRL Contest 2023** [1–3]. It is to train a stock trading agent by applying novel data and feature engineering based on market data. It is a multi-asset trading task for 30 constituent stocks in the Dow Jones Index. The task has been clearly defined and well-studied in FinRL [1–3] and FinRL-Meta [5, 6]. The construction of the task is stable and reproducible, making it ideal for benchmarking in a competition setting. In addition, this task focuses on data processing and feature engineering, encouraging participants to tackle the challenge of dynamic market data. This aligns with the community's ongoing interest in exploring dynamic financial data and developing stable multi-asset environments.
- **LLM-Engineered Signals with RLMF @ FinRL Contest 2024** [23, 24]. It is to develop LLMs that can generate actionable trading signals from financial news by using reinforcement learning from market feedback (RLMF). LLMs have been applied in many financial tasks [24], but general-purpose LLMs trained on Internet data cannot capture the intrinsic dynamics of financial markets. To align LLMs with financial markets, this task encourages adapting LLMs using reinforcement learning from market feedback, as a counterpart to reinforcement learning from human feedback (RLHF) [25]. RLMF utilises feedback from the financial market as reward signals, enabling LLMs to learn from financial market behaviours. This task was selected for its novelty and potential to push the frontier of LLM-enhanced financial decision-making. It aligns with the growing interest in the FinRL and FinLLM communities in combining FinRL and LLMs.
- **FinRL-DeepSeek for Stock Trading @ FinRL Contest 2025** [23, 26]. It is to develop trading agents by combining FinRL and LLM-engineered signals. Unstructured financial texts, such as news and SEC filings, contain valuable information about market sentiment, risk and events. LLMs, such as DeepSeek-V3, have been applied to extract signals from financial documents [24], which are then integrated into FinRL. This task was selected because it reflects real-world decision-making, where financial analysts and traders rely on both structured market data and unstructured financial text. It also aligns with the growing interest in the FinRL and FinLLM communities in LLM-engineered signals and multimodal data exploration.

Different from stock markets, crypto trading faces greater volatility due to drastic price fluctuations and market sentiment shifts. Crypto markets operate 24/7, demanding adaptable strategies in a continuous trading environment without traditional market open and close cycles. It is more challenging to develop robust and profitable trading strategies. To reflect these unique challenges, we include crypto trading tasks in the FinRL Contests. To differentiate from the multi-asset stock trading tasks above, the crypto trading tasks are designed to trade single-asset Bitcoin at the second level. The crypto trading tasks include:

- **Crypto Trading With Ensemble Learning @ FinRL Contest 2024** [27]. It is to train a trading agent for Bitcoin by using ensemble methods. RL policies are unstable, whose performance is sensitive to hyperparameters, market noise and random seeds. Policy instability can also come from value function approximation errors. These issues are amplified in crypto markets with high volatility. Ensemble methods have shown effectiveness in mitigating policy instability [18]. This task extends these ideas to the crypto trading tasks. In addition, collecting high-quality financial trajectories is often expensive and limited by data access. It causes the sampling bottleneck, especially during training multiple FinRL agents for an ensemble. Therefore, this task encourages participants to employ novel ensemble methods and utilise the provided GPU-optimised parallel environments to improve sampling speed. It also aligns with the need in the financial industry for robust and efficient trading strategies in crypto markets.
- **FinRL-AlphaSeek for Crypto Trading @ FinRL Contest 2025** [27] extends the previous task of crypto trading by introducing an additional focus on factor mining. Factors such as momentum play a critical role in driving trading decisions, enabling traders to design efficient, data-driven strategies. This task encourages participants to develop a two-stage pipeline: 1) factor engineering and selection process, and 2) ensemble learning to build robust trading agents. It aligns with the industry need for effective alpha and beta signal extraction in crypto markets.
- **FinRL-DeepSeek for Crypto Trading @ FinAI Contest 2025** [27] extends the previous two tasks of crypto trading

by introducing LLM-engineered signals. Crypto markets are highly sensitive to news headlines, tweets, regulatory shifts and viral narratives. The timely interpretation of market sentiment is critical for crypto trading. LLMs can extract actionable signals from financial news, tweets and filings. In this task, we encourage participants to explore LLM-engineered signals and integrate them into a FinRL trading agent for crypto trading.

To address the sampling bottleneck of the training stage, we develop massively parallel market environments on GPUs. We encourage contestants to use ensemble methods to enhance policy stability and improve models' robustness. These will be described in the following section.

3.2 | MDP Formulation

We formulate stock and crypto trading tasks as Markov decision processes (MDPs) [6] and the diagram is shown in Figure 3.

- **State** $\mathbf{s_t} = \left[b_t, \mathbf{p_t}, \mathbf{h_t}, \mathbf{f_t}\right] \in \mathbb{R}^{K(I+2)+1}$ represents market conditions at time t. $b_t \in \mathbb{R}_+$ is the cash balance. $\mathbf{p_t} \in \mathbb{R}_+^K$ is the prices of stocks or cryptocurrencies, where K is the number of assets. $\mathbf{h_t} \in \mathbb{R}_+^K$ is the holding positions. $\mathbf{f_t} \in \mathbb{R}^{KI}$ is the feature vector, where there are I features for each asset.
- **Action** $\mathbf{a_t} \in \mathbb{R}^K$ represents trading actions at time t, such that $\mathbf{h}_{t+1} = \max(\mathbf{h_t} + \mathbf{a_t}, 0)$ (making $\mathbf{h}_{t+1}$ nonnegative). An entry $a_t^i > 0, i = 1, ..., K$ indicates buying a_t^i shares of assets i, whereas $a_t^i < 0$ indicates selling and $a_t^i = 0$ indicates holding the current position. Each entry of $\mathbf{h_{t+1}}$ is nonnegative, which means the agent cannot sell more shares than it currently holds.
- **State-transition function** $\delta(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ defines the probability distribution over the next state $\mathbf{s}_{t+1}$ given the current state $\mathbf{s}_t$ and action $\mathbf{a}_t$. Transitions are governed by real-world asset price movements and trading outcomes. In simulation with historical market data, the transitions may be deterministic during training. In real-time trading, the state transition function is stochastic.
- **Reward** $r_{t+1} = R(\mathbf{s_t}, \mathbf{a_t}, \mathbf{s_{t+1}})$ is an incentive signal to encourage or discourage the trading agent to perform action $\mathbf{a_t}$ at state $\mathbf{s_t}$. We use r_{t+1} instead of r_t to reflect the delayed reward in financial markets. It emphasises that the reward r_{t+1} is determined at the next state $\mathbf{s}_{t+1}$ after taking action $\mathbf{a}_t$. The reward can be defined as the change in the total asset value, that is, $R(\mathbf{s_t}, \mathbf{a_t}, \mathbf{s_{t+1}}) = v_{t+1} - v_t$, where $v_t = b_t + \mathbf{p_t^T h_t}$ is the total asset value at time t. The reward can also be penalised for high-risk signals.
- **Discount factor** $\gamma \in (0, 1)$ is used to evaluate discounted expected return at time t. The discount factor γ determines the present value of future rewards.

Note that a policy $\pi(\cdot|\mathbf{s}_t)$ is a probability distribution over actions at state $\mathbf{s}_t$. It determines the likelihood of each possible trading action given the current market conditions. The objective of the trading agent is to learn a policy π that maximises cumulative positive rewards while minimising losses over time.

In the state $\mathbf{s}_t$, the holding positions $\mathbf{h}_t$ are typically constrained to be nonnegative in long-only trading settings, where the agent is not allowed to short assets. The feature vector $\mathbf{f}_t$ can include market indicators (e.g., moving average convergence divergence), factors learnt by neural networks, and LLM-engineered signals from financial news or regulatory filings. The action $\mathbf{a}_t$ can be discrete or continuous, depending on the task. It may support long-only or long-short strategies. The reward function R can use the Sharpe ratio in addition to the change in asset value. It can be further adjusted by risk or market sentiment to better reflect real-world trading scenarios.

For example, a trading task is to develop a trading agent for all 30 constituent stocks in the Dow Jones Index with an initial investment of 1 million. In this setting, the number of assets is $K = 30$. The state includes the cash balance of $b_0 = 100,000$, prices of the 30 stocks, holding positions of the 30 stocks, and $I = 4$ technical indicators for each stock. The state space has a

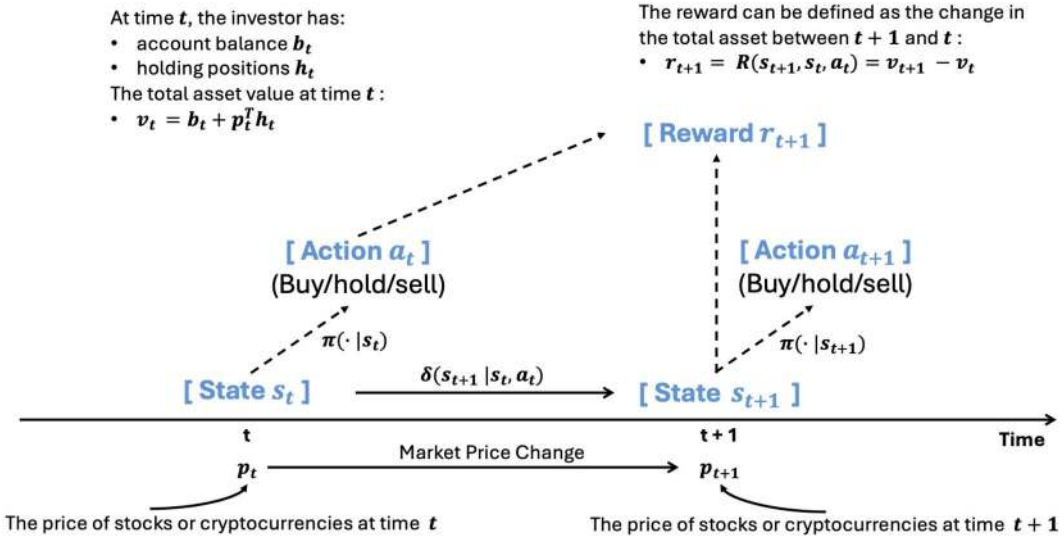


FIGURE 3 | The stock and crypto trading tasks in FinRL are modeled as Markov Decision Processes.

dimension $K(I + 2) + 1 = 181$. The action space has dimension $K = 30$. An entry, for example, $a_t^1 = 5$ means buying 5 shares of the first stock in the portfolio. The reward function is the change in the total asset value, with $v_0 = 100,000$. Taking the first stock as an example, assume at time t, the cash balance is $b_t = 100,000$, the price of the first stock is $p_t^1 = 200$, and the shares of the first stock are $h_t^1 = 10$. The agent buys 5 shares of the first stock $\left(a_t^1 = 5\right)$ and the price increases to $p_{t+1}^1 = 201$ at time $t + 1$. Assuming holding all other 29 stocks, the state is updated where $b_{t+1} = 99,000$, $p_{t+1}^1 = 201$, and $h_{t+1} = 15$. The return $r_t = v_{t+1} - v_t = \left(b_{t+1} + \mathbf{p}_{t+1}^T \mathbf{h}_{t+1}\right) - \left(b_t + \mathbf{p}_t^T \mathbf{h}_t\right) = b_{t+1} - b_t + p_{t+1}^1 h_{t+1}^1 - p_t^1 h_t^1 = -1000 + 201 \cdot 15 - 200 \cdot 10 = 15$. In this example, the agent only operates on the first stock.

In each specific FinRL Contest task, the definitions and constraints associated with the environment and state space vary. Detailed descriptions for each task are provided in Sections 4, 5 and 6.

3.3 | Offline Policy Learning and Its Parallelism

In the FinRL's trading tasks, the policy π_θ is parameterised by a neural network (e.g., transformer network or diffusion model) with learnable parameters θ. Such a policy network is trained through offline policy learning algorithms. A trajectory, $\tau = (\mathbf{s}_0, \mathbf{a}_0, r_1, \mathbf{s}_1, \mathbf{a}_1, r_2, \mathbf{s}_2, \mathbf{a}_2, ..., \mathbf{s}_T)$, is a sequence of trading actions and states observed over a period, where $\mathbf{s}_0$ is the initial state and T is the number of steps. The probability of the trajectory τ is

$$P(\tau|\pi_\theta) = \rho_0(\mathbf{s}_0) \prod_{t=0}^{T-1} \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)\delta(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t), \tag{1}$$

where $\rho_0(s_0)$ is the initial state distribution. The financial results are quantified as rewards along τ to calculate the cumulative return $R(\tau) = \sum_{t=1}^{T} \gamma^{t-1} r_t$. The goal is to learn a policy π_θ that maximises the following expected return

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)] = \int_\tau P(\tau|\pi_\theta)R(\tau), \tag{2}$$

The gradient of $J(\theta)$ with respect to θ is [28]

$$\nabla J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[(R(\tau) - b)\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)\right], \tag{3}$$

where $b \in \mathbb{R}$ is a constant, and subtracting b can reduce the variance in practice.

In the FinRL tasks, the trading strategy is governed by π_θ to maximise $J(\theta)$. When dealing with complex and noisy financial datasets, achieving a low-variance gradient estimation $\nabla J(\theta)$ is crucial for stable and reliable policy updates, reducing the risk of suboptimal trading strategies.

To estimate $\nabla J(\theta)$ in Equation (3), we can use the Monte Carlo method [29]

$$\nabla J(\theta) = \frac{1}{n}\sum_{j=1}^{n}\left(\left(R\left(\tau^{(j)}\right) - b\right)\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta\left(\mathbf{a}_t^{(j)} \mid \mathbf{s}_t^{(j)}\right)\right), \tag{4}$$

where n trajectories are used. The Law of Large Numbers guarantees that as the sample size n increases, the estimation of $\nabla J(\theta)$ will converge to its expected value. According to the Central Limit Theorem, increasing n leads to a reduction in the variance of the estimate.

The overall process of offline policy learning is shown in Figure 4. In offline policy learning, the agent interacts with the environment over time and generates trajectories τ's. The trajectories are stored in the replay buffer as trading experiences. In the learning phase, a batch of trajectories is sampled from the replay buffer and used to update the policy through policy gradient methods. The policy is updated to the direction of $\nabla J(\theta)$, as estimated in Equation (4), and α is the learning rate. Then the agent uses the updated policy to interact with the environment and continues the offline policy learning process.

For Equation (4), a large number n of trajectories sampled during the simulation phase is required to reduce the variance of $\nabla J(\theta)$. In Equation (4), each j in the outer sum from 1 to n corresponds to a separate trajectory $\tau^{(j)}$, which can be considered as a complete and independent simulation of the policy π_θ in the environment. Therefore, each trajectory $\tau^{(j)}$ can be

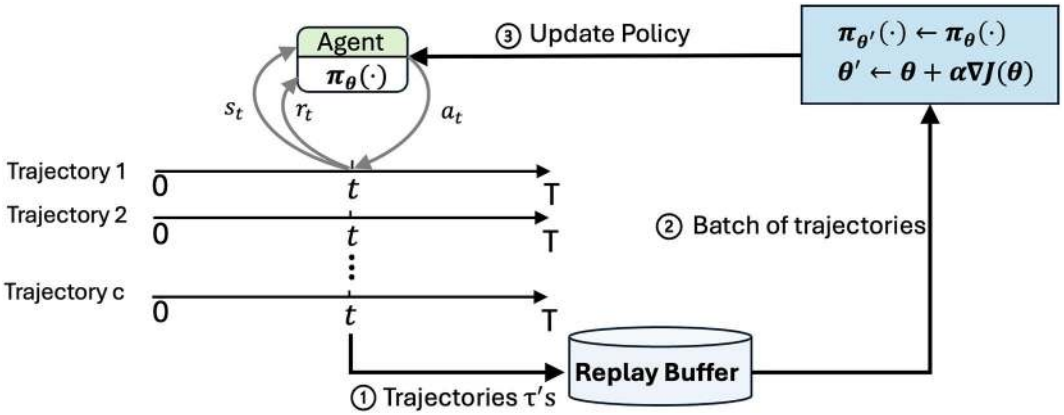


FIGURE 4 | Illustration of offline policy learning.

simulated in parallel, allowing for a high degree of parallelism. In FinRL, parallel simulation involves executing the trading strategy in multiple market scenarios simultaneously. The parallelism accelerates the simulation phase, allowing for more rapid updates and iterations of the policy π_θ. Therefore, the trading strategy governed by π_θ can be quickly updated and adapted to changing market conditions.

The application of deep generative models, such as transformers and sequence models, in offline policy learning has shown great potential [30]. Decision transformers formulate offline policy learning as a sequence modelling task and are trained on the datasets of trajectories to autoregressively predict actions. The training relies on large datasets of trajectories. Parallel simulation can more efficiently generate massive, high-quality trajectories to train the deep generative models for offline policy learning. In addition, offline policy learning can also start with behavioural cloning by learning expert demonstrations. Wu et al. [31] proposed financial curriculum learning, which is a two-stage imitation-and-reinforcement learning method. The agent is first trained through imitation learning to align with retail traders' strategies and then trained through DRL. This approach enables the agent to leverage expert knowledge while also adapting to dynamic market environments. Parallelism in offline policy learning can be leveraged in both stages to accelerate the whole pipeline.

4 | FinRL Environments

Market data and the simulation environment are two critical components of FinRL that define trading tasks. In this section, we provide detailed information on market data and GPU-optimised parallel market environments.

4.1 | Automated Data Curation Pipeline for Dynamic Market Data

We first describe different sources of dynamic market data used in FinRL Contests, along with feature engineering, market data split strategies and the automated data curation pipeline.

4.1.1 | Data Sources

We leverage both structured market data and unstructured financial documents. Structured market data captures quantitative market behaviour, such as historical price information and market indicator data. Unstructured financial documents provide information about economic trends, regulations, and events that drive market movements. The structured market data are processed into price features $\mathbf{p}_t$, which are included in state $\mathbf{s}_t$. We outline the financial data utilised in FinRL as follows:

- **OHLCV data.** The OHLCV (open, high, low, close and volume) data are typical historical volume-price data in finance. We provide daily OHLCV data for stocks from 1999 to 2023. This dataset is prepared through yfinance[1] (an open-source Python library) and FinRL-Meta [5, 6], released under the CDLA-permissive-2.0 licence. The missing values are removed. We also provide data APIs to download market data, such as Alpaca,[2] which are held on the open-source repository FinRL[3] [5, 6].
- **Limit order book (LOB)** data at second level for cryptocurrency trading.[4] LOB data offers a detailed view of market depth and liquidity, capturing the behaviour of market participants and providing valuable insights into market trends. Given their high granularity and extensive size, these data are ideally suited for participants to leverage massively parallel simulations, enabling more effective development of cryptocurrency trading strategies.
- **Financial news.** The Financial News and Stock Price Integration Dataset (FNSPID) [32] contains 15 million time-aligned financial news articles from 1999 to 2023 for constituent companies in S&P 500. It is released under CC-BY-NA 4.0 for academic purposes. From the FNSPID dataset, we randomly select one news article per day per stock. The news is aggregated with OHLCV data to ensure temporal alignment and create a multimodal dataset. For cryptocurrency news, we aggregate BTC news collected from multiple sources. The aggregated BTC news dataset: https://huggingface.co/datasets/SecureFinAI-Lab/FinRL_BTC_news_signals. Data sources: https://huggingface.co/datasets/edaschau/bitcoin_news [5]; and ensure the temporal alignment with the LOB data. We also provide data APIs to access financial news, such as Yahoo Finance and Finnhub. These data APIs are held on the open-source repository[6] [23].

4.1.2 | Feature Engineering

We conduct feature engineering before making the dataset available for contests.

- **Market indicators.** Ten market indicators listed in Table 2 are computed based on OHLCV data. These indicators are used to enrich the feature vector $\mathbf{f}_t$ in state $\mathbf{s}_t$.
- **ML-learnt factors.** For the LOB data, bid and ask prices at different market depths are extracted, and 101 adapted alpha factors [33] are constructed. An RNN is trained to distil 8 strong factors from the 101 weak alphas, modelling and predicting the prices. These factors are used to enrich the feature vector $\mathbf{f}_t$ in state $\mathbf{s}_t$.

The feature selection process involves two steps: (1) compute the Pearson correlation matrix of the features, and (2) select one representative feature from each group of highly correlated features, following the method proposed by Gort et al. [27].

4.1.3 | LLM-Engineered Signals

Recent studies have demonstrated the feasibility of employing LLMs to generate trading signals from financial data [34]. For instance, RiskLabs [35] utilises LLM to assess market risk levels across multiple financial data sources. ECC analyser [36]

TABLE 2 | Market indicators and their descriptions.

Indicator	Name	Description
macd	Moving average convergence divergence	A trend-following momentum indicator that shows the relationship between two exponential moving averages (EMAs) of a stock's price. Traders use the MACD to identify potential trend changes, divergence between price and momentum, and overbought or oversold conditions
boll_ub	Bollinger bands upper band	Bollinger bands are used to visualise the volatility and potential price levels of a stock. The upper band represents the upper volatility boundary, showing where the price might find resistance
boll_lb	Bollinger bands lower band	Similarly, the lower band represents the lower volatility boundary and shows where the price might find support
rsi_30	Relative strength index for 30 periods	A momentum oscillator that measures the speed and change of price movements. RSI oscillates between zero and 100
cci_30	Commodity channel index for 30 periods	A versatile indicator that can be used to identify a new trend or warn of extreme conditions. It measures the current price level relative to an average price level over a given period of time
dx_30	Directional movement index for 30 periods	An indicator that assesses the strength and direction of a trend of a stock. It does this by comparing highs and lows over time
close_30	Thirty-period simple moving average of closing prices	Represents the average closing price over the last 30 periods. This moving average provides a smoothed representation of the asset's price over the 30 periods, making it easier to identify trends and potential support/resistance levels
close_60	Sixty-period simple moving average of closing prices	Represents the average closing price over the last 60 periods. This moving average provides a smoothed representation of the asset's price over the 60 periods, making it easier to identify trends and potential support/resistance levels
vix	Volatility index	Often referred to as the 'fear index', it represents the market's expectation of 30-day forward-looking volatility. It is calculated from the prices of selected stock option contracts on the S&P 500 index
Turbulence	Turbulence	To control the risk in a worst-case scenario, such as the financial crisis of 2007–2008, FinRL employs the financial turbulence index that measures extreme asset price fluctuation

employs a retrieval-augmented generation (RAG) framework built upon LLMs, enabling users to extract trading insights from earnings conference call transcripts through customised queries. These textual insights are integrated with audio signals to support downstream tasks, such as predicting market volatility. Moreover, several studies have applied LLMs or FinGPTs [23, 37, 38] to extract trading signals, including volatility and sentiment analysis [39], from monetary policy conference [40], financial reports [41] and news [42]. LLMs also have shown considerable promise for generating alpha signals across multiple financial data sources [24, 43, 44]. We encourage participants to leverage LLMs and train a trading agent for making informed decisions. Here we show the sentiment and risk scores generated by DeepSeek models from financial news [26]:

- **Sentiment score.** An LLM assigns a sentiment score u of 1–5 according to the news, with 1 for strongly negative and 5 for highly positive. For example, we can use DeepSeek-V3 [45] to generate signals. It is included in the feature vector $\mathbf{f}_t$ and is used to adjust actions via the sentiment factor $l_t^i = 1 + 0.05(u - 3)\text{sign}(a_t^i)$. The factor is close to 1 for the stability of the algorithm. The adjusted action is $a_t^{i'} = l_t^i a_t^i$, which is amplified under positive sentiment and dampened under negative sentiment.
- **Risk level.** An LLM assigns a risk level q of 1–5 from the news, with 1 for low risk and 5 for high risk. It is included in the feature vector $\mathbf{f}_t$ and is used to penalise rewards through the risk factor $m_t^i = 1 + 0.05(q_t^i - 3)$. The aggregated risk factor is $M_t = \sum_i^K w_t^i m_t^i$, where w_t^i is the portfolio weight of the stock i and $\sum w^i = 1$. $M_t > 1$ penalises the reward for high risk; $M_t < 1$ increases the reward for low risk.

4.1.4 | Automated Data Curation Pipeline

After preparing the datasets, we split them into two parts, one released to participants for training and the other held by organisers for evaluation purposes. We use the temporal partitioning approach to maintain the temporal integrity of financial data and prevent future information leakage.

- **Dataset released to participants.** We provide long-span historical datasets for participants to develop their models. It covers different market patterns, such as financial crises, bull markets and bear markets. Participants are not restricted to the provided datasets and APIs. They are encouraged to use external data sources, such as alternative market indicators, financial news feeds, and tweets.
- **Dataset withheld for evaluation.** Depending on the task and data availability, we use two methods to obtain the evaluation dataset: (1) we extract the most recent ~15% of the original dataset, withhold it as the evaluation dataset, and encrypt all timestamps; (2) we collect out-of-sample data for the period after the model submission deadline. It involves downloading market data and financial news via `yfinance` or other APIs. These two methods will avoid future data leakage and ensure a fair assessment. This setup ensures that the evaluation simulates real-world, time-forward deployment, where models should generalise to new, unseen market data.

To better reflect real-world forward-moving financial markets, we present an automated data curation pipeline. We illustrate this pipeline using the example of the paper trading task. The conventional backtesting approach splits historical market data into in-sample and out-of-sample time periods. Backtesting may be performed multiple times on the out-of-sample data. However, stock paper trading is strictly required to be carried out **once** in real-time on the real-world market.

Figure 5 shows our 'training-validation-trading' pipeline. In a window, there are X days' data for training and Y days' data for validation. At the end of a window, we perform paper trading for 1 day. Note that we always retrain the agent using $X + Y$ days of training and validation data together. Then, we roll the window forward by 1 day ahead and perform the above steps for a new window. Paper trading is always carried out for 1 day. Therefore, Z windows correspond to Z trading days.

Algorithm 1 summarises the pipeline of paper trading. For Z trading days ($z = 0, 1, ..., Z - 1$), we keep doing the following three steps:

- **Step (1).** Download and process X-day data, from day $z - Y - X$ to day $z - Y - 1$. Then build the data into a gym-style environment and train the agent. Then download and process Y-day data, from day $z - Y$ to day $z - 1$. Then build the data into a gym-style environment and validate how the agent performs. According to the agent's performance on the validation environment, adjust the hyperparameters.
- **Step (2).** Build the training and validation data, totally $X + Y$ days from day $z - Y - X$ to day $z - 1$, into a gym-style environment. Update hyperparameters to the values chosen from **Step (1).** Then retrain the agent on this $X + Y$-day environment.
- **Step (3).** Deploy the trained agent to the paper trading market.

Before each trading day z, we use the historical data period from $z - Y - X$ to $z - Y - 1$ to train the FinRL agent. Then, data periods from $z - Y$ to $z - 1$ are used to validate the trained agent, adjusting hyperparameters accordingly. We retrain the agent by combining all the data from the training and validation period, that is, the data period from $z - Y - X$ to $z - 1$. Finally, we perform 1 day paper trading on the z-th day. We continue this process for $z = 0, 1, ..., Z - 1$.

ALGORITHM 1 | Pseudo code for stock paper trading.

```
Initialise a set of hyperparameters;
for z = 0 to Z − 1 do
    # Step 1). Train an agent
    Train the agent on data period [z−Y−X, z−Y−1]
    Validate trained agent and tune hyperparameters on
    data period [z − Y, z − 1]
    # Step 2). Retrain the agent
    Retrain the agent on data period [z−Y−X, z−1] using
    tuned hyperparameters
    # Step 3). Perform paper trading for one day
    Trade on day z with the trained agent
end for
```

4.2 | Market Environments

The financial data are processed into standard gym-style market environments to build a standard practical environment. This part describes market environments with near-real market constraints. To address the sampling bottleneck of the training stage, we develop massively parallel market environments on GPUs.

4.2.1 | Market Environment With Trading Constraints

In Section 3.2, we have defined the MDP process in FinRL trading tasks with **State** $\mathbf{s_t} = [b_t, \mathbf{p_t}, \mathbf{h_t}, \mathbf{f_t}] \in \mathbb{R}^{K(I+2)+1}$, **Action** $\mathbf{a_t} \in \mathbb{R}^K$, **State-transition function** $\delta(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$, **Reward** $r_{t+1} = R(\mathbf{s_t}, \mathbf{a_t}, \mathbf{s_{t+1}})$ and **Discount factor** $\gamma \in (0, 1)$. The financial data are processed into standard gym-style market environments. The technical indicators, ML-learnt factors and LLM-engineered signals are included in the $\mathbf{f}_t$. We define the key operations in the FinRL market environment, including:

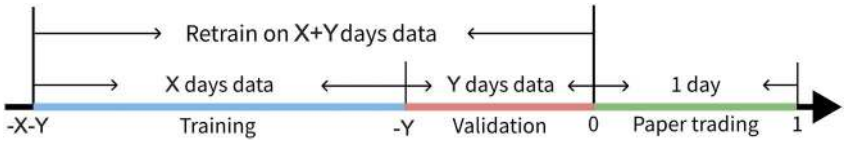


FIGURE 5 | Data split for a window with training, validation and trading.

- `reset`: $\mathbf{s}_t = [b_t, \mathbf{p_t}, \mathbf{h_t}, \mathbf{f_t}] \rightarrow \mathbf{s}_0 = [b_0, \mathbf{p_0}, \mathbf{h_0}, \mathbf{f_0}]$, resets the environment to its initial state. b_0 is the amount of initial investment and $\mathbf{h}_0 = 0$ since there are no shareholdings yet. As shown in Figure 4, `reset` sets the state back to time $t = 0$.

- `step`: $(\mathbf{s}_t, \mathbf{a}_t) \rightarrow \mathbf{s}_{t+1}$, executes $\mathbf{a}_t$ and updates $\mathbf{s}_t$ to $\mathbf{s}_{t+1}$. As shown in Figure 3, given $\mathbf{a}_t$ and $\mathbf{s}_t$, the environment is updated to $\mathbf{s}_{t+1}$ through state-transition function $\delta(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$. In $\mathbf{s}_{t+1}$, price $\mathbf{p}_{t+1}$ is updated to the new price at the next trading period, feature vector $\mathbf{f_{t+1}}$ is calculated based on the time series of market data, and b_{t+1} and h_{t+1} are updated as follows:

$$b_{t+1} = b_t - \mathbf{p}_t^T \mathbf{a_t}.$$

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \mathbf{a}_t.$$

The total asset value is changed from $v_t = b_t + \mathbf{p}_t^T \mathbf{h_t}$ to $v_{t+1} = b_{t+1} + \mathbf{p}_{t+1}^T \mathbf{h_{t+1}}$

- `reward`: $R(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) \rightarrow r_{t+1}$, computes the reward. In Section3.2, we defined $r_{t+1} = v_{t+1} - v_t$ and r_{t+1} is calculated as follows:

$$r_{t+1} = v_{t+1} - v_t = \left(b_{t+1} + \mathbf{p}_{t+1}^T \mathbf{h}_{t+1}\right) - \left(b_t + \mathbf{p}_t^T \mathbf{h}_t\right).$$

The `reward` function is called by the `step` function when the agent interacts with the environment.

To reflect the real-world trading scenario, we incorporate near-real trading constraints:

- **Transaction costs.** Each trade has transaction fees and different brokers charge varying commissions. In FinRL, we set a cost of 0.1% for each action {buy, sell}, accounting for commission and slippage. Therefore, the reward becomes:

$$r_{t+1} = v_{t+1} - v_t = \left(b_{t+1} + \mathbf{p}_{t+1}^T \mathbf{h}_{t+1}\right) - \left(b_t + \mathbf{p}_t^T \mathbf{h}_t\right) - \left(\mathbf{p}_t^T |a_t| \times 0.1\%\right).$$

  where $|a_t|$ means taking the entry-wise absolute value of a_t.

- **Market volatility.** The turbulence index and VIX index are risk indicators. A large value signals heightened volatility from factors such as investor fear and increased uncertainty, while a small value signals increased stability in markets.

4.2.2 | Massively Parallel Simulations on GPUs

Stable policy updates require a low-variance gradient estimate $\nabla J(\theta)$ from a large n independent trading trajectories τ, where there is high parallelism, as shown in Equation (4). PyTorch's `vmap` can map operations over some dimension onto parallel GPU cores, exploiting the parallelism of Equation (3). Therefore, we construct vectorised environments and provide GPU optimisation via `vmap`. As shown in Figure 6, a vectorised environment manages parallel sub-environments (SubEnvs).

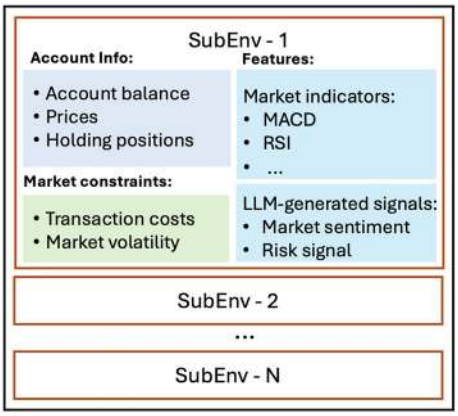


FIGURE 6 | Vectorised environment.

Using `vmap`, the `step` and `reward` functions are vectorised to operate in massively parallel environments. For example, the `reward` function, vectorised by `vmap`, computes the reward on $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$ for each SubEnv simultaneously. This computation is dispatched to available GPU cores, each responsible for calculating its assigned data.

Data samples are stored as tensors in GPU memory. They have the shape $N \times T \times D$:

- N is the number of parallel SubEnvs.
- T is the number of steps in a trajectory.
- D is the dimension as in Section 3.2, where $D = K(I + 2) + 1$ for state, $D = K$ for action, and $D = 1$ for reward.

The tensors for states $\left(\mathbf{s} \in \mathbb{R}^{K(I+2)+1}\right)$, actions $\left(\mathbf{a} \in \mathbb{R}^K\right)$ and rewards $(r \in \mathbb{R})$ are as follows:

$$\begin{bmatrix} \mathbf{s}_0^1 & \mathbf{s}_1^1 & \cdots & \mathbf{s}_{T-1}^1 \\ \mathbf{s}_0^2 & \mathbf{s}_1^2 & \cdots & \mathbf{s}_{T-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_0^N & \mathbf{s}_1^N & \cdots & \mathbf{s}_{T-1}^N \end{bmatrix}, \begin{bmatrix} \mathbf{a}_0^1 & \mathbf{a}_1^1 & \cdots & \mathbf{a}_{T-1}^1 \\ \mathbf{a}_0^2 & \mathbf{a}_1^2 & \cdots & \mathbf{a}_{T-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_0^N & \mathbf{a}_1^N & \cdots & \mathbf{a}_{T-1}^N \end{bmatrix}, \begin{bmatrix} r_1^1 & r_2^1 & \cdots & r_T^1 \\ r_1^2 & r_2^2 & \cdots & r_T^2 \\ \vdots & \vdots & \ddots & \vdots \\ r_1^N & r_2^N & \cdots & r_T^N \end{bmatrix}.$$

Storing data samples as tensors in GPU memory bypasses the CPU-GPU bandwidth bottleneck.

4.2.2.1 | Improved Sampling Speed With Massively Parallel Environments on GPU. We evaluated the sampling speed measured in samples per second using vectorised environments for stock trading. We used the PPO agent and the OHLCV data of 30 constituent stocks in the Dow Jones index, from 1 January 2020 to 1 January 2022. The NVIDIA A100 GPU is used. The numbers of parallel environments vary from 1, 2, 4, ..., to 2048. As shown in Figure 7, the average sampling speed with 2048 parallel environments is 227,212.54 samples per second. The sampling speed is improved by $1,649.93\times$ compared with a single environment. The sampling speed scales approximately linearly with the number of parallel environments. The results show the effectiveness of massively parallel simulation in improving sampling speed in FinRL Contest tasks.

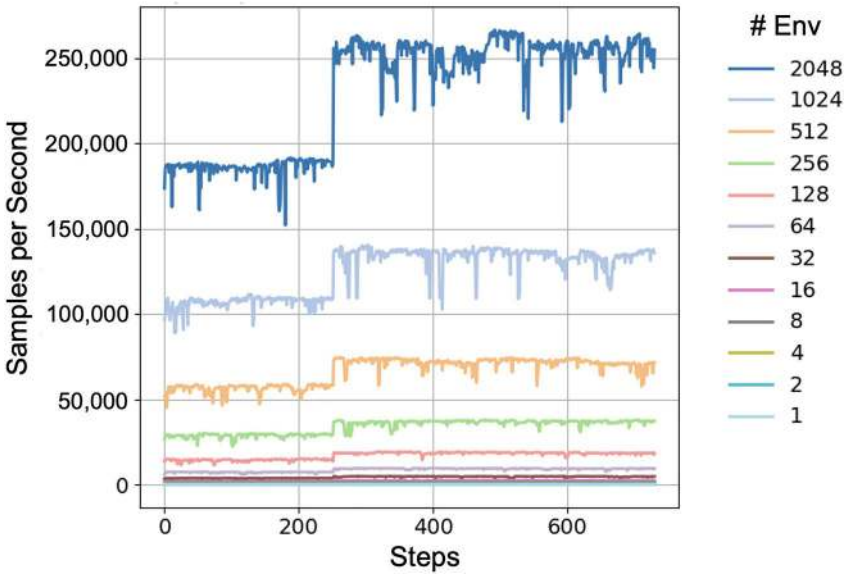


FIGURE 7 | Samples per second for the stock trading task (using NVIDIA A100 GPU).

5 | Benchmarking Protocol

To assist participants in becoming familiar with FinRL Contests and to ensure fair comparisons, we establish a benchmarking protocol and provide several baseline methods.

5.1 | Baseline Methods

5.1.1 | Market Index and Baseline Methods

We provide the following baseline methods to participants that are widely used in the financial industry.

- **DJIA.** The Dow Jones Industrial Average index is a price-weighted index for 30 blue-chip US companies. It is calculated as the sum of constituent stock prices divided by the Dow Divisor. The divisor is a constant adjusted for stock splits and structural changes. The data can be downloaded by `yfinance`. DJIA is one of the oldest and most recognised market indexes. It provides a real-world baseline to evaluate whether a FinRL agent outperforms a passive investment strategy.
- **S&P 500.** The Standard and Poor's 500 is a capitalisation-weighted index tracking the stock performance of 500 leading companies. The data can be downloaded by `yfinance`. Covering approximately 80% of the total market capitalisation, it offers a broader and more diversified baseline.
- **Mean-variance optimisation.** Mean-variance optimisation strategy, as part of modern portfolio theory (MPT) [46], constructs portfolios that maximise the expected return for a given level of risk. It uses expected asset returns and covariances to solve an optimisation problem, where risk is quantified using the covariance matrix of asset returns. We typically use the past 1 year's daily price data to calculate expected returns and the covariance matrix. We limit individual stock weights to a maximum of 5%. Mean-variance optimisation is a foundational technique in portfolio management and can serve as a classical financial optimisation strategy baseline.
- **Buy-and-hold strategy**: With the buy-and-hold strategy, the investor purchases assets and holds them for a long period without frequent reactions to short-term market fluctuations. This strategy is based on the assumption that, over the long term, markets tend to rise and short-term volatility will not significantly impact the overall performance. The buy-and-hold strategy is a passive investment strategy, often used as a benchmark to compare the performance of active trading strategies.
- **Equal-weighted portfolio.** In an equal-weighted portfolio, the investor allocates an equal percentage of their total capital to each asset of the portfolio. This approach is straightforward and does not require complex calculations for capital allocation. Since ML models are often trained to seek optimal weights in portfolio management, the equal-weighted investment strategy is often used as a benchmark for evaluating the performance of ML methods.

5.1.2 | Ensemble Agent

Ensemble methods have shown effectiveness in enhancing overall performance by combining selected actions or action probabilities from component RL algorithms [47]. We train ensemble trading agents to mitigate policy instability.

5.1.2.1 | Agent Diversity. The diversity of component agents is essential for risk mitigation by leveraging various trading agents. Achieving high diversity requires training multiple agents across environments that simulate different market scenarios.

5.1.2.1.1 | Using KL Divergence in Objective Functions. To enforce diversity among the component agents, we introduce a Kullback–Leibler (KL) divergence term into the

agent's objective function. The KL divergence measures how one probability distribution diverges from another [48]. The KL divergence term penalises similarities in policies between different agents, encouraging them to converge on different trading strategies. The new objective function for a component agent is as follows:

$$\max L_{\text{new}}(\theta_A) = L_{\text{original}}(\theta_A) + \lambda \sum_{A \neq B} \text{KL}(\pi_{\theta_B} || \pi_{\theta_A}), \tag{5}$$

where θ_A are the policy parameters for agent A, $L(\theta_A)$ is the objective function, $\text{KL}(\pi_{\theta_B} || \pi_{\theta_A})$ is the KL divergence between agent A's and agent B's policies, and $\lambda > 0$ is a regularisation constant. After obtaining the trained agents, one can ensemble them using a majority voting method or the weighted sum method [18].

5.1.2.1.2 | Using Various Datasets. The financial datasets used for training component agents are varied. For each stock or crypto, a random percentage change ranging from -1% to 1% is generated and applied to its prices, which shifts the price scale while preserving the original price trends. Agents are also trained on different stocks from the test set for the stock trading task. It enables agents to learn various strategies for a broader range of stocks rather than reacting to a limited number of stocks.

5.1.2.2 | Stock Trading Task. As shown in Figure 8a, the ensemble for stock trading includes PPO, SAC, and DDPG agents. These agents act in a continuous action space, allowing for fine-grained portfolio weights. The ensemble's final trading action is determined by weighted averaging over the agents' action probabilities. The process is as follows:

- **Training.** Each component agent is independently trained in a VecEnv on a 30-day training rolling window, using massively parallel simulation in Section 4.2.2 and agent diversity methods in Section 5.1.2.1.
- **Validation.** After training, agents are validated on a 5-day rolling window. Sharpe ratios are calculated to evaluate their ability to balance returns with associated risks.

$$\text{Sharpe Ratio} = \frac{\bar{r}_p - r_f}{\sigma_p}, \tag{6}$$

where $\bar{r}_p$ is the portfolio return, r_f is a chosen risk-free rate, and σ_p is the standard deviation of the portfolio return.

- **Weights calculation.** Agents with very low Sharpe ratios during validation are discarded. Weights for the remaining agents are calculated using a softmax function applied to their Sharpe ratios.
- **Trading.** The ensemble acts based on a weighted average of agent action probabilities during a 5-day trading window.

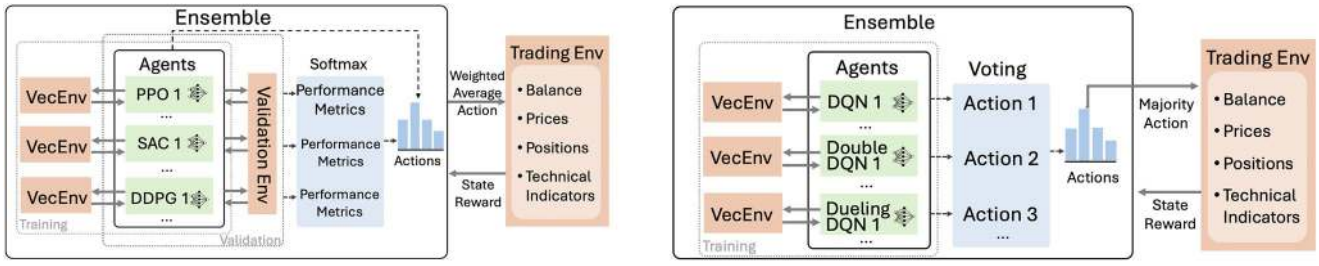
This rolling window approach ensures that the ensemble method remains adaptive to the continuously changing market.

5.1.2.3 | Crypto Trading Task. The ensemble method for the crypto trading task is shown in Figure 8b, which adopts different agents and ensemble methods from stock trading.

5.1.2.3.1 | Agent Selection. For crypto trading at a relatively high frequency, market movements can be modelled as discrete events, which require a discrete action space. Therefore, DQN [49], Double DQN [50] and Duelling DQN [51] are used to handle this discrete action space. In addition, the dataset for a single crypto is relatively small. DQN and its variants, with fewer parameters and simpler architectures, can be trained faster to avoid overfitting. Moreover, trading at a relatively high frequency requires fast responses, and DQN agents can offer lower latency in decision-making compared to more complex models.

5.1.2.3.2 | Ensemble Methods. The ensemble model for crypto trading uses majority voting rather than weighted averaging to combine the actions of component agents. The approach of validating agents and calculating the Sharpe ratio to assign weights is time-consuming and is acceptable in daily stock trading. However, fast decision-making is crucial in relatively high-frequency trading. Majority voting is faster and well-suited for a discrete action space. It ensures that the chosen action reflects consensus among agents, mitigating biases from any single agent's actions [52]. The process is as follows:

- **Training.** Each component agent is independently trained with a VecEnv, using massively parallel simulation in Section 4.2.2 and the agent diversity methods in Section 5.1.2.1.
- **Majority voting.** During the testing phase, each agent processes the same market state and determines an action based on its policy. The majority action is selected as the final ensemble action.
- **Trading.** The ensemble takes the selected action to interact with the trading environment.



(a) Ensemble method for the stock trading task

(b) Ensemble method for the cryptocurrency trading task

FIGURE 8 | Ensemble methods.

5.2 | Evaluation Protocol

To evaluate the models, we adopt two frameworks, backtesting and rolling windows, as shown in Figure 9:

- **Backtesting.** We used backtesting to evaluate models for trading tasks in FinRL Contests 2024 and 2025, as shown in Figure 9a. The datasets released to participants are used for their training and validation. The withheld out-of-sample dataset is used for evaluation. We did not use the rolling window framework because the evaluation datasets for stock trading have long time spans, and the crypto data are of high frequency. In both cases, retraining models frequently would be too time-consuming and difficult to manage.
- **Rolling windows.** We use the rolling window framework to evaluate models for FinRL Contest 2023 data-centric stock trading and paper trading, as discussed in Section 4.1. Figure 9b presents the mechanism of each rolling window for the stock trading task. There are in total 9 days in a rolling window. We divide it into three stages, namely, 6 days of training, 2 days of validation, and 1 day of trading using the trained DRL agent. The models are retrained for 8 days before each trading day. The validation stage allows adjusting the hyperparameters or selecting an agent from the ensemble. Finally, the well-trained agent performs stock trading on the last day of a rolling window.

To ensure fair and reproducible benchmarking for all models, the evaluation process follows the principles below:

- **Uniform evaluation platform.** All models are evaluated on the same infrastructure to eliminate differences caused by hardware or software variations. Specifically, we conduct evaluations using Google Cloud Platform (GCP) virtual machines with identical hardware configurations. The software environment is Ubuntu 22.04.2 LTS and Python 3.10.12.
- **Controlled evaluation setting.** All models are evaluated in the same environment settings, including testing data, initial investment amount, and transaction costs. No additional training or model adjustment is allowed during the evaluation phase.

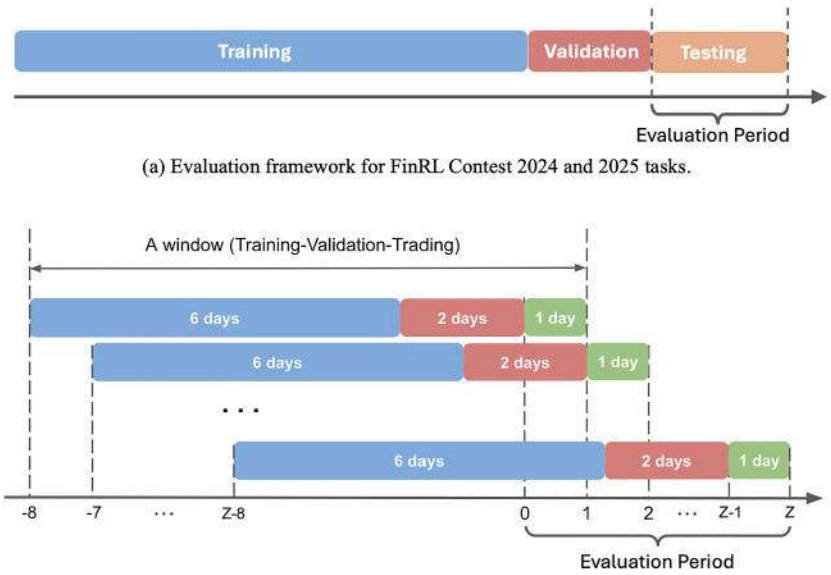
To ensure the correctness and reproducibility of the results, we implement a standard evaluation process based on the uniform evaluation platform and controlled settings:

- **Complete repository submission requirements.** Participant teams are required to submit a well-organised GitHub repository, which contains all code, scripts, model weights, and any custom libraries used to implement the solution. Participant teams also need to provide a detailed README for step-by-step instructions. This allows evaluators to verify and reproduce results independently using the submitted codebase.
- **Validation.** Each solution will be tested by at least two evaluators. The results are checked to ensure the discrepancies fall within an acceptable range. Significant discrepancies will be further reviewed.
- **Result Scrutiny.** When discovering abnormal or possibly incorrect results, we review the evaluation process and the submitted code. We contact the participant team for clarification if necessary. If we cannot finish the scrutiny before the deadline of the leaderboard announcement, we will annotate the entry and inform participants for future scrutiny.

6 | Performance

6.1 | Performance Metrics

We use the well-established quantitative metrics [2] in finance to evaluate models, as shown in Table 3. For each metric, we



(a) Evaluation framework for FinRL Contest 2024 and 2025 tasks.

(b) Rolling windows for FinRL Contest 2023 data-centric stock trading task and paper trading task.

FIGURE 9 | The rolling windows for the evaluation of the stock trading task.

TABLE 3 | Performance metrics.

Metric	Definition	Range	Direction
Cumulative return	Measure the total value change of an investment over time by summing daily logarithmic returns. Higher values indicate better strategy effectiveness	$(-\infty, +\infty)$	Higher is better
Annualised return	The geometric average amount of money earned by the agent each year over a given time period	$(-\infty, +\infty)$	Higher is better
Annualised volatility	The annualised standard deviation of daily returns. This metric highlights potential return deviations across the year	$[0, +\infty)$	Lower is better
Sharpe ratio	Assesses risk-adjusted returns by dividing the average excess return over the risk-free rate by its volatility. Higher ratios signify better performance	$(-\infty, +\infty)$	Higher is better
Maximum drawdown	Calculates the largest portfolio value drop from peak to trough. Lower values indicate a lesser risk and higher strategy robustness	$[-1, 0]$	Higher is better
Rachev ratio	A risk-adjusted return that measures the potential upside gain compared to potential downside risk, using the tails of the return distribution	$(-\infty, +\infty)$	Higher is better
Return over maximum drawdown (RoMaD)	It is calculated as the cumulative return divided by the absolute value of maximum drawdown	$(-\infty, +\infty)$	Higher is better
Sortino ratio	The excess return divided by the downside deviation	$(-\infty, +\infty)$	Higher is better
Calmar ratio	The annualised excess return divided by the maximum drawdown	$(-\infty, +\infty)$	Higher is better
Omega ratio	It compares the probability of achieving returns above a threshold to the probability of falling below it	$[0, +\infty)$	Higher is better
Win/loss ratio	It is calculated by dividing the number of winning trades by the number of losing trades	$[0, +\infty)$	Higher is better

report its definition, numerical range, and optimisation direction (i.e., whether higher or lower values indicate better performance).

6.1.1 | Returns

The most frequently used metrics to measure profitability include cumulative return and annualised return. The cumulative return reflects the overall growth of the investment over the entire trading period. The adjusted closing price is more suitable for calculating the cumulative return because it has considered the impact of interest, dividends, and stock splits. The annualised return is the geometric average return per year, typically assuming each year has 252 trading days. It is preferred when fairly comparing the performance of trading strategies with different trading periods.

6.1.2 | Risk

The most frequently used metrics to measure risk include annualised volatility and maximum drawdown. Annualised volatility is the standard deviation of daily returns scaled to the yearly horizon (252 trading days). It reflects the variability of returns. Maximum drawdown measures the largest drop of the portfolio value from peak to trough. It only indicates the largest loss the investor can experience, not reflecting the frequency of large losses.

6.1.3 | Risk-Adjusted Return

Risk-adjusted performance measures evaluate return relative to its risk. The Sharpe ratio is the most used risk-adjusted return, calculated using Equation (6). Different from the Sharpe ratio, the Sortino ratio only focuses on the downside risk by using the downside deviation as the denominator. It evaluates the profitability of a trading strategy for a given level of bad risk, where upside variability is usually not a concern for investment. Return over maximum drawdown (RoMaD) is often used as an alternative to the Sharpe Ratio or Sortino Ratio, calculated as the cumulative return divided by the absolute value of maximum drawdown. The Calmar ratio is an annualised alternative to RoMaD, calculated as annualised return divided by maximum drawdown. RoMaD and the Calmar ratio are useful for strategies that aim to minimise large losses. The Rachev ratio compares the right tail return to the left tail loss on a certain percentage (typically 5%) of the return distribution. It is useful when returns have a fat-tailed and skewed distribution. The Omega ratio is to select a threshold of the cumulative return distribution and compare the area above the threshold and the area below. By varying the threshold, it captures different investor risk preferences. The win/loss ratio indicates the

effectiveness of a trading strategy in making profitable trades. It is useful in high-frequency trading or short-term trading, where consistency in trading is critical.

6.2 | Stock Trading Tasks

6.2.1 | Data-Centric Stock Trading

The data-centric stock trading task in the FinRL Contest 2023 is to train a trading agent by using novel data and feature engineering strategies. The training dataset is the OHLCV dataset with 10 market indicators for all 30 constituent stocks in the Dow Jones Index from 1 July 2010 to 24 October 2023 (3352 trading days). The market indicators are shown in Table 2. The models are evaluated for two periods: 25 October 2023–14 November 2023 (15 trading days, pre-submission deadline) and 15 November 2023–21 November 2023 (5 trading days, post-submission deadline). Table 4 shows the cumulative return, Sharpe ratio, and maximum drawdown of the top three winners and the Dow Jones Index. The following teams used different approaches[7]:

- **SZU-Fin-621** employed PPO-Switch, which is an ensemble method combining multiple PPO agents. The agent pool was created by training different PPO agents with diverse online portfolio selection (OPS) [53] price features. The agent was selected based on its short- and long-term returns, and the action space was sparsified to enhance capital efficiency [54].
- **Nik-Elena** added new technical indicators in the state, including the Relative Strength Index for different periods, On-Balance Volume, and Moving Average for different periods. The team also enlarged the negative return when the turbulence exceeded a threshold.
- **WeCan** added 101 formulaic alpha factors [33] as new features in the state, in addition to the provided market indicators.

The results show that the teams' strategies achieved better risk-adjusted returns and risk management, but showed poorer profitability compared with the Dow Jones Index. In the first testing period, from 25 October 2023 to 14 November 2023, all teams outperformed the Dow Jones Index in the maximum drawdown. Nik-Elena achieved the best maximum drawdown of −0.40%, indicating effective downside risk management. In the second testing period from 15 November 2023 to 22 November 2023, SZU-FIN-621 and WeCan had negative cumulative returns and Sharpe ratios. Nik-Elena had a small positive return of 0.04% but still underperformed the market index. However, Nik-Elena still achieves a better maximum drawdown than the Dow Jones Index, indicating better risk control even in challenging market environments. Participants' models showed strong risk management compared to the DJIA in the first testing period, but their generalisation to new, unseen market conditions remains a challenge. For the anomalous Sharpe Ratio reported in Table 4, one possible reason is due to very limited testing data (only 1–3 weeks). It may introduce high variance in performance. The standard deviation may be underestimated due to insufficient variance being captured, leading to a high Sharpe ratio. After this contest, we adopted a longer period of testing data in subsequent contests.

6.2.1.1 | Take-Home Messages. Data-centric approaches, such as feature engineering, can have better risk management. However, maintaining consistent profitability across unseen market regimes remains a challenge. The evaluation methodology (e.g., short testing periods) can also distort standard metrics such as the Sharpe ratio, showing the importance of robust evaluation protocols in future contests.

6.2.2 | Stock Trading With Ensemble Methods

We performed the stock trading task for 30 stocks in the Dow Jones index by using three ensemble models, and individual PPO, SAC and DDPG agents.

6.2.2.1 | Stock Data. We use historical daily OHLCV data for all 30 stocks in the Dow Jones index from 1 January 2021 to 1 December 2023 (734 trading days). OHLCV data is a rich source for learning financial market behaviours and trends. We use the technical indicators listed in Table 2. These indicators enrich the data with more insights into market behaviours and trends. Therefore, $K = 30$ and $I = 10$ in the setting of Section 3.2.

TABLE 4 | The performance of top winner teams for FinRL Contest 2023 Task 1 data-centric stock trading.

Team	Testing period	Cumulative return (%)	Sharpe ratio[a]	Maximum drawdown (%)
SZU-FIN-621	10/25/2023–11/14/2023	1.05	2.26	−1.36
	11/15/2023–11/22/2023	−0.03	−10.25	−0.03
Nik-Elena	10/25/2023–11/14/2023	3.50	9.56	−0.40
	11/15/2023–11/22/2023	0.04	0.95	−0.15
WeCan	10/25/2023–11/14/2023	2.35	8.07	−0.55
	11/15/2023–11/22/2023	−0.05	−1.74	−0.11
DJIA	10/25/2023–11/14/2023	5.42	0.45	−1.87
	11/15/2023–11/22/2023	0.80	0.49	−0.18

Note: The stocks are constituents of the Dow Jones Index ($K = 30$).
[a]The results of the Sharpe ratio were reported wrongly.

6.2.2.2 | Agents for Stock Trading Task. We use PPO, SAC and DDPG agents. The policy network for each agent consists of a feed-forward network with two hidden layers, having 64 and 32 units, respectively. We set a learning rate of $3 \cdot 10^{-4}$ and a batch size of 64. All ensemble models and individual agents are trained, validated, and tested on **a rolling-window basis** with 30-day training, 5-day validation, and 5-day testing windows.

6.2.2.3 | Ensemble Methods. The first method (Ensemble 1) consists of 1 PPO, 1 SAC and 1 DDPG agents; the second method (Ensemble 2) consists of 5 PPO, 5 SAC and 5 DDPG agents; and the third method (Ensemble 3) consists of 10 agents for each type. As in Section 5.1.2.2, all three ensemble models use the weighted average approach to combine component agent action probabilities.

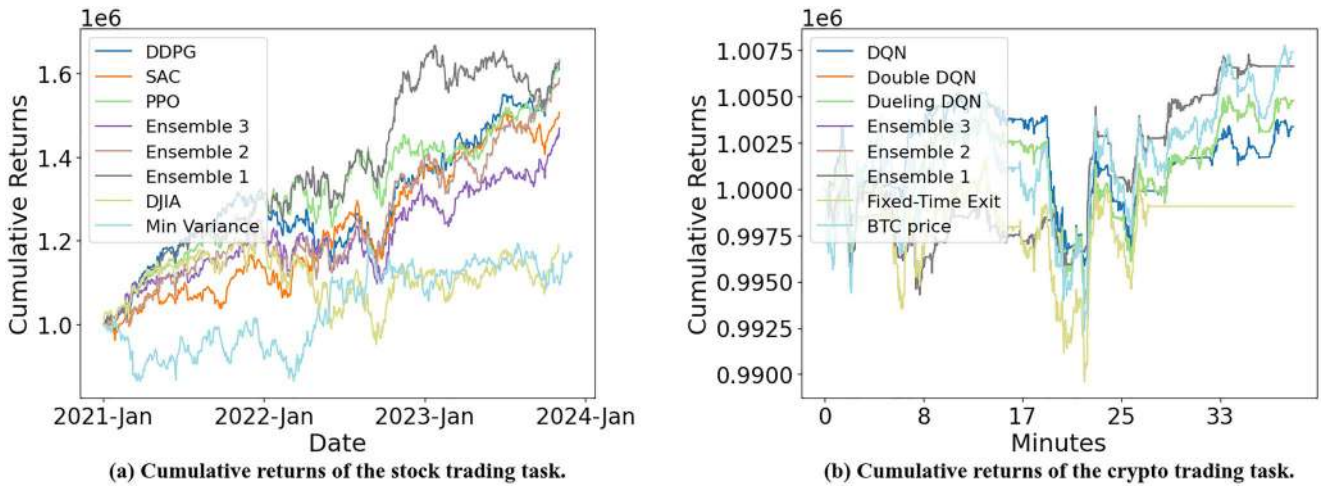
6.2.2.4 | Results. As can be seen in Table 5, the PPO agent achieves the highest cumulative returns of 63.37%, Sharpe ratio of 1.55 and Sortino ratio of 2.44, showing an ability to maintain high returns with controlled volatility and downside risk. Although DDPG's cumulative returns are comparable to PPO's, its higher maximum drawdown of −13.15% signals a greater risk of large value drops, which is a concern for risk management. SAC has a lower maximum drawdown than DDPG but underperforms in other metrics. All individual agents significantly outperform two traditional baselines across all metrics. The ensemble models also maintain profitability and risk management advantages over the baselines. Ensemble 1 has a high cumulative return of 62.60%, and as shown in Figure 10a, it shows superior performance from September 2022 to October 2023. Ensemble 1 also achieves the smallest maximum drawdown and a higher Sharpe ratio than SAC and DDPG. Ensembles 1 and 2 have high RoMaD and Calmar ratios, showing an ability to quickly recover from peak-to-trough losses and a potential for steady growth in market adversities.

6.2.2.5 | Take-Home Messages. FinRL agents show superior performance over traditional baselines, including the market index and min-variance strategy. Ensemble methods can effectively address the policy instability of individual RL agents by combining the strengths of multiple agents and reducing individual failures. These results support the use of ensemble methods in FinRL as a promising direction for robust financial decision-making.

TABLE 5 | Stock trading task performance.

Model	Ensemble-1	Ensemble-2	Ensemble-3	PPO	SAC	DDPG	Min-variance	DJIA
Cumulative return	62.60%	58.77%	46.89%	**63.37%**	50.62%	63.19%	13.9%	18.95%
Annual return	18.22%	17.25%	14.15%	**18.41%**	15.14%	18.36%	7.34%	6.15%
Annual volatility	11.76%	12.61%	12.70%	**11.35%**	11.67%	11.93%	18.16%	15.14%
Sharpe ratio	1.48	1.33	1.11	**1.55**	1.27	1.47	0.48	0.47
Sortino ratio	2.34	2.14	1.74	**2.44**	2.05	2.37	0.73	0.67
Max drawdown	**−8.98%**	−11.27%	−12.27%	−9.96%	−12.02%	−13.15%	−14.9%	−21.94%
RoMaD	**6.97**	5.22	3.82	6.36	4.21	4.81	1.10	0.86
Calmar ratio	**2.03**	1.53	1.15	1.85	1.26	1.40	0.49	0.28
Omega ratio	1.31	1.28	1.23	**1.33**	1.27	1.32	1.09	1.08

Note: Models are trained, validated and tested on a rolling window basis on OHLCV datasets for 30 Dow Jones stocks. Ensemble models use weighted averages on agent action probabilities. The best results are highlighted in bold.



(a) Cumulative returns of the stock trading task.

(b) Cumulative returns of the crypto trading task.

FIGURE 10 | Cumulative returns of different strategies for the stock trading task and crypto trading task.

6.2.3 | LLM-Engineered Signals for Stock Trading

We designed two tasks around LLM-engineered signals from news, including FinRL Contest 2024 LLM-Engineered Signals with RLMF and FinRL Contest 2025 FinRL-DeepSeek for Stock Trading.

6.2.3.1 | FinRL Contest 2024 LLM-Engineered Signals With RLMF. Participants fine-tuned LLMs to generate signals through reinforcement learning from market feedback. The training dataset was daily OHLCV data and financial news for $K = 7$ large-cap stocks from 1 January 2020 to 8 October 2022 (447 trading days). The effectiveness of LLM-engineered signals is evaluated on a dataset from 9 October 2022 to 15 December 2023 (229 trading days), using a long-short trading strategy, that is, long (buy) the three stocks with the highest sentiment scores and short (sell) the bottom three, closing all positions 3 days later. **Aethernet**[8] [55] fine-tuned the LLaMA-3.2-3B-Instruct model through RLMF to generate sentiment scores from news. The team designed a reward function to assess the sentiment score based on market data. The model had a cumulative return of 134.05%, whereas a simple buy-and-hold strategy yielded a 72.71% return. It indicates that the LLM-engineered signals were effective even using a simple long-short execution strategy. The win/loss ratio of 1.5 suggests that the model generated more profitable trades than losing ones.

6.2.3.2 | FinRL Contest 2025 FinRL-DeepSeek for Stock Trading. This task uses a different approach to combine FinRL and LLM-engineered signals, where the signals are integrated into the environment to train FinRL agents. The training dataset is daily OHLCV data with 10 market indicators (listed in Table 2) and financial news for Nasdaq 100 constituent stocks from 1 January 2013 to 31 December 2018 (1510 trading days). The models are evaluated on the testing dataset from 1 January 2019 to 31 December 2023 (1258 trading days). The winning teams used different approaches[9]:

- **Otago Alpha** [56] added the put-call ratio to the feature vector, which is an assessment of put and call option transaction volumes. The put-call ratio was sourced from the OptionMetrics via the Wharton Research Data Services (WRDS) database and Bloomberg Terminal.
- **Ruijian and Sally** [57] used the Group Relative Policy Optimisation (GRPO) [58] algorithm, which is refined by Decoupled Clipping and Dynamic sAmpling Policy Optimisation (DAPO). The team also designed a reward function adjusted by sentiment and risk.
- **Kuxlnw** [59] adopted a dynamic model selection mechanism that adapts to market conditions and macroeconomic indicators. The team added new features to the environment, including interest rates, gold prices and oil prices. They also used DeepSeek-generated sentiment scores and risk levels in the features. They trained specialised FinRL agents for each stock. Then they employed a mechanism to dynamically allocate the most suitable trading agents based on bull or bear markets. The algorithms they use include PPO with 100 training epochs (PPO 100), PPO with DeepSeek-generated signals and 100 training epochs (PPO-DeepSeek 100), CPPO with 100 training epochs (CPPO 100), CPPO with DeepSeek-generated signals and 100 training epochs (CPPO-DeepSeek 100) and Adaptive Portfolio applying the dynamic model selection mechanism.
- **Queen's Gambit** [60] proposed a two-phase framework, market-informed sentiment for trading (MIST), to extract signals from news. In the first phase, it used a structured prompt with few-shot examples for DeepSeek-R1 7B to evaluate financial news and assign a sentiment score (1–5). In the second stage, the team designed an advanced prompt incorporating market behaviour (i.e., short-term stock price change direction) to either reinforce or contradict the sentiment extracted in the first stage. The final trading signal is integrated into the environment to train FinRL agents. The team trained two PPO agents, one with only the first stage of MIST (PPO with MIST 1) and one with two stages of MIST (PPO with MIST 2).

The performance of the winning teams is shown in Table 6. Team Otago Alpha, Ruijian and Sally, and Queen's Gambit showed a superior performance over S&P 500 and Nasdaq-100

TABLE 6 | The performance of top winner teams for FinRL Contest 2025 Task 1 FinRL-DeepSeek for stock trading.

Team	Algorithm	Cumulative return	Sharpe ratio	Max drawdown	Rachev ratio
Otago alpha	PPO	191.14%	1.0800	−28.16%	1.0557
Ruijian and Sally	Refined GRPO	335.57%	0.9500	−50.24%	1.0300
Kuxlnw	Adaptive portfolio	163.88%	0.0574	−33.10%	0.9389
	PPO 100	204.51%	0.0671	−36.56%	1.0027
	CPPO 100	90.52%	0.0359	−34.78%	1.0194
	PPO-DeepSeek 100	94.43%	0.0433	−34.09%	0.9441
	CPPO-DeepSeek 100	91.26%	0.0383	−43.03%	0.8684
Queen's Gambit	PPO with MIST 1	238.70%	0.2859	−92.20%	6.7999[a]
	PPO with MIST 2	342.65%	0.2897	−92.47%	6.6723[a]
S&P 500	/	90.03%	0.0448	−33.93%	0.9047
Nasdaq-100	/	164.52%	0.0558	−35.56%	0.9434

[a]The Rachev ratio of Queen's Gambit is reported wrongly and needs further scrutiny.

in cumulative return and Sharpe ratio. Otago Alpha achieved the highest Sharpe ratio (1.08) and maximum drawdown (−28/16%), and the cumulative return outperformed two market indices, showing a good balance between profitability and risk management. Although Ruijian and Sally and Queen's Gambit showed high cumulative returns, they had a worse maximum drawdown of −50.24% and −92.47%, respectively, compared with two market indices, indicating a larger downside risk. A Rachev ratio larger than 1 indicates that their strategies' upside tail rewards outweighed their downside tail risks during extreme market movements. Kuxlnw's adaptive portfolio underperformed its PPO 100 model in cumulative return, Sharpe ratio, and Rachev ratio, suggesting that a multi-asset trading agent may have more stable performance in this setting.

6.2.3.3 | Take-Home Messages. Both tasks demonstrate the power of combining FinRL and LLM-engineered signals. Although LLM-generated signals have shown the ability to improve cumulative returns, downside risk management, as reflected by maximum drawdowns, remains a key challenge. In addition, the FinRL Contest 2025 attracted over 80 teams, with particularly strong interest in the task of FinRL-DeepSeek for stock trading. This shows the growing interest in applying LLMs to extract trading signals from financial news and documents.

6.3 | Crypto Trading Tasks

The FinRL Contest 2024 Crypto Trading with Ensemble Learning and the FinRL Contest 2025 FinRL-AlphaSeek for Crypto Trading are tasks designed for Bitcoin trading.

6.3.1 | Crypto Trading With Ensemble Methods

In our experiment, the crypto trading task for Bitcoin (BTC) ($K = 1$) is performed using three ensemble models, and individual DQN, Double DQN and Duelling DQN agents.

6.3.1.1 | Crypto Data. The dataset comprises second-level LOB data for BTC from 7 April 2021 11:32:42 to 19 April 2021 09:54:22. Adaptations from 101 formulaic alphas [33] are calculated based on LOB data to extract insights into market behaviours, such as momentum, mean-reversion and market anomalies. A recurrent neural network (RNN) further processes the 101 alphas into $I = 8$ technical indicators. It reduces the complexity of the input data and enhances the ability to predict market trends, thus improving generalisation and avoiding overfitting. The RNN is trained on data from 7 April 2021 11:32:42 to 17 April 2021 00:38:02 without future information leaks. The agents are trained on the in-sample data from 04/17 00:38:03 to 04/19 09:09:21 and tested on the out-of-sample data from 04/19 09:09:22 to 04/19 09:54:22.

6.3.1.2 | Agents for Crypto Trading Task. We use DQN, Double DQN, and Duelling DQN agents. The policy network for each agent consists of a feed-forward neural network with three 128-unit hidden layers. We set an exploration rate of 0.005, a learning rate of $2 \cdot 10^{-6}$ and a batch size of 512.

6.3.1.3 | Ensemble Methods. The first method (Ensemble 1) consists of 1 DQN, 1 Double DQN, and 1 Duelling DQN agents; the second method (Ensemble 2) consists of 3 DQN, 3 Double DQN and 3 Duelling DQN agents; and the third method (Ensemble 3) consists of 10 agents for each type. As in Section 5.1.2.3, three ensemble models use a majority voting approach to aggregate the agents' actions. All ensemble models and individual agents are trained on the in-sample data and tested on the out-of-sample data.

6.3.1.4 | Results. As seen in Table 7 and Figure 10b, Double DQN and Duelling DQN agents have similar performance, with cumulative returns of 0.48%. This is lower than the BTC price baseline. Despite this, they achieve higher Sharpe ratios of 0.21 and lower maximum drawdowns of −0.98% than the fixed-time exit strategy and BTC price baseline, suggesting effective risk management. The three ensembles' cumulative returns are close to the BTC price baseline. Moreover, the ensemble models outperform all individual agents in all metrics, achieving the highest Sharpe ratio of 0.28 and the lowest maximum drawdown of −0.73%. They also have the highest win/loss ratio of 1.62. This shows that ensemble methods can mitigate the risks associated with the decision-making failures of single agents. The three different ensemble models have similar performances, which may be due to the limited action space at each timestep, causing agents to output identical actions. In our experiment,

TABLE 7 | Crypto trading task performance.

Model	Ensemble-1	Ensemble-2	Ensemble-3	DQN	Double DQN	Duelling DQN	Fixed-time exit	BTC price
Cumulative return	0.66%	0.66%	0.66%	0.34%	0.48%	0.48%	−0.1%	**0.74%**
Sharpe ratio	**0.28**	**0.28**	**0.28**	0.15	0.21	0.21	−0.03	0.20
Maximum drawdown	**−0.73%**	**−0.73%**	**−0.73%**	−0.93%	−0.98%	−0.98%	−1.00%	−1.3%
RoMaD	**0.90**	**0.90**	**0.90**	0.37	0.49	0.49	0.10	0.59
Sortino ratio	**0.39**	**0.39**	**0.39**	0.20	0.29	0.29	−0.04	0.28
Omega ratio	**1.08**	**1.08**	**1.08**	1.04	1.05	1.05	0.99	1.05
Win/Loss ratio	**1.622**	**1.622**	**1.622**	1.309	1.617	1.617	0.5	—

Note: The second-level LOB data for Bitcoin is split into out-of-sample data for training and in-sample data for testing. Ensemble models use majority voting on agent actions. The best results are highlighted in bold.

the action space is {sell, hold, and buy} one unit of BTC. This restricted action space allows agent policies to converge to similar behaviour.

We observe that individual agents and majority voting ensembles have near-identical performances; in the restricted action space, agent policies may be converging, indicating a greater risk of significant losses due to less diversified trading actions. Compared to the spot price and fixed-time exit, the ensemble consistently achieved a higher return over the maximum drawdown ratio, highlighting superior risk-adjusted returns. Because of the lack of diversity in agents, we observed near-identical results between ensembles and individual agents. Nonetheless, over a relatively short timespan of 30 min, we observe that the vectorised agents can outperform other strategies.

6.3.1.5 | Take-Home Messages. In cryptocurrency trading, ensemble methods can address the policy instability of individual agents. This is important in maintaining portfolio stability amid high market volatility. The ensemble has a higher risk-adjusted return and win/loss rate than the individual agents, showing its enhanced profitability and trading consistency in the volatile cryptocurrency markets.

6.3.2 | Crypto Trading in FinRL Contests

Based on the previous experiments, we designed tasks allowing participants to explore novel factor mining strategies or ensemble methods. The training dataset is a second-level LOB dataset for Bitcoin from 7 April 2021 11:32:42 to 17 April 2021 00:38:02. The testing dataset is from 17 April 2021 00:38:02 to 19 April 2021 09:54:22. We re-partitioned the dataset to allow for a longer testing period. The performance of winner participant teams' open-source solution, Mt. Everest[10], is shown in Table 8.

- **Fermion** used ensemble methods to combine the strengths of different on-policy and off-policy algorithms. The team applied majority voting based on confidence scores to determine the trading action.
- **Mt. Everest** added new technique indicators in the state of the market environment, including order book imbalance, spread-based liquidity, order flow imbalance, price impact of trades, market-to-limit order ratio, intraday volatility, momentum scores and noise-to-signal ratio. In addition, the team used principal component analysis (PCA) to extract the top 10 strongest factors from 101 alpha factors [33]. To construct the agent pool, they applied different policy networks, including LSTM, transformer and dilated CNNs. The action was determined through majority voting.
- **Cryptoalphaseek** [61] proposed a hybrid deep learning architecture for cryptocurrency trading, combining advanced sequence modelling with reinforcement learning. They integrated Temporal Convolutional Networks (TCN) for local feature extraction, parallel Mamba State Space Models (SSM) for capturing long-term dependencies and multi-head attention for modelling global context. These outputs were passed through a gated mechanism, followed by an adaptive signal smoothing module to further mitigate noise in the generated signals. The output signals were then fed into a reinforcement learning framework for crypto trading.

All teams outperformed the baseline BTC price in cumulative returns, Sharpe ratio and maximum drawdown. Fermion achieved a positive cumulative return of 0.23% and a Sharpe ratio of 0.0037. All teams' higher maximum drawdown shows better downside risk control and loss mitigation, which is a notable achievement given BTC's inherent volatility.

In addition, FinRL-Crypto[11] [27] introduces a multi-crypto trading strategy. The agents were trained using 5-min-level OHLCV data for $K = 10$ cryptocurrencies from 2 February 2022 to 30 April 2022. The agents were trained with multiple hyperparameter settings across 10 training-validation splits. To address backtest overfitting, agents with a high probability of overfitting were rejected at a 10% significance level. The model was evaluated on the dataset from 1 May 2022 to 27 June 2022, which includes two market crashes. The best-performing PPO agent achieved a cumulative return of −34.96%, outperforming the S&P crypto Broad Digital Market Index (S&P BDM Index, −50.78%) and the equal-weight strategy (−47.78%). It also had lower volatility (0.0020), compared with S&P BDM Index (0.0581) and the equal-weight strategy (0.0042).

6.3.2.1 | Take-Home Messages. In the volatile cryptocurrency markets, extracting actionable factors is crucial for trading strategies. Team Mt. Everest outperformed the baseline using traditional and PCA-derived factors, whereas Team Cryptoalphaseek's complex model design did not have superior results. It shows the need for further validation of factor extraction methods. In addition, FinRL-Crypto shows that cross-validation and overfitting control are effective in building robust trading agents that can generalise well under extreme market conditions.

TABLE 8 | The performance of winners for FinRL Contest 2024 Task 1 'Crypto Trading with Ensemble Learning' and FinRL Contest 2025 Task 2 'FinRL-AlphaSeek for Crypto Trading'.

Contest	Team	Cumulative return	Sharpe ratio	Maximum drawdown
FinRL Contest 2024	Fermion	0.23%	0.0037	−0.28%
FinRL Contest 2025	Mt. Everest	−0.05%	−0.0008	−1.23%
	Cryptoalphaseek	−4.67%	−0.03	−5.54%
Baseline	BTC price	−7.35%	−0.0022	−17.02%

7 | Organisational Aspects

We held FinRL Contests at several academic conferences, including ACM International Conference on AI in Finance (ACM ICAIF 2023, 2024), IEEE International Conference on Intelligent Data and Security (IEEE IDS 2025) and IEEE International Conference on Cyber Security and Cloud Computing (IEEE CSCloud 2025). In this section, we summarise the organisational experience of the FinRL Contests, including encouraging participation, contest process, platform setup, event promotion, registration and submission approaches and communication.

7.1 | Participation

From 2023 to 2025, a total of 230+ students, researchers and practitioners participated. Figure 11 shows the distribution of participation by geography and the word cloud of affiliations. Among the 162 participants who reported their institutions, 39% of participants are from North America, including the United States and Canada. 44% of participants come from Asia, mainly from China, South Korea, and India. The participants come from both academia (e.g., Tsinghua University, Columbia University) and industry (e.g., FTI Consulting, Logictronix Technologies).

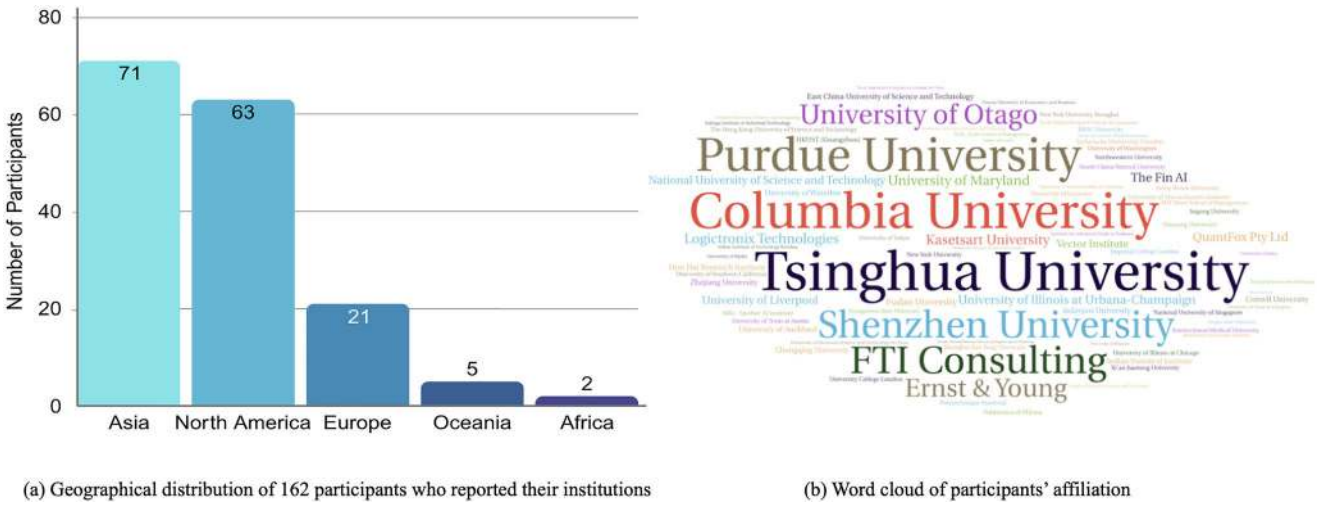
7.2 | Contest Process

We first set important dates, including team registration, starter-kit release, model submission deadline, report submission deadline and leaderboard release. The full contest typically lasts 2–3 months. It aims to provide participants with sufficient time for solution development while ensuring timely evaluation and feedback.

- **Team registration.** Each team consists of 1–4 people. Team registration starts as early as possible to leave enough time for promotion. The starter kit is nearly ready when team registration begins so that registered teams can be engaged and start to brainstorm early. Registration remains open until the model submission deadline to continuously encourage new teams to join.
- **Starter kit release.** The starter kit is released within a week after team registration begins. This short buffer period is used to organise and clean the prepared datasets, code and instructions. The early release ensures that teams can be engaged early and have ample time to develop their solutions.
- **Model submission.** The model submission deadline is set more than a month after the starter kit release. This period gives the teams sufficient time to get familiar with the tasks, study the tutorials, explore the dataset, use the code and develop their own models.
- **Report submission.** In addition to models, participant teams are encouraged to submit a 2–3 page short paper describing the methodologies and performance. The report submission deadline is set 1 week after the model submission deadline. Participant teams can finalise documentation after completing model development. We invited a panel of 11 expert reviewers, including Ph.D. researchers, postdoctoral fellows and professors from Columbia University, New York University Shanghai, Stevens Institute of Technology and Princeton University. They review the submitted reports and give valuable revision advice.
- **Result announcement.** Final results are announced 15 days after the report submission deadline. During this period, evaluators assess the submitted models, and experts review the reports. Final rankings are determined based on a weighted score, 60% of model performance and 40% of the report assessment.

7.3 | Platform Setup

Multiple platforms are set up to host different resources for the contest, as shown in Table 1:



(a) Geographical distribution of 162 participants who reported their institutions

(b) Word cloud of participants' affiliation

FIGURE 11 | Geographic distribution of participation and word cloud of participants' affiliations (162 participants reported their affiliations).

- **Contest website.** The website includes the contest overview, task description, contest timeline, registration and submission guidelines, contact information and links to related resources. It is set up by using GitHub Pages for wide accessibility. It also serves as the main portal for announcements and updates.
- **GitHub repository.** A GitHub repo is set up to host the starter kit. The datasets, code and detailed descriptions are organised in a folder for each task. It also contains around 8 tutorial notebooks for participant teams to get started.
- **Documentation website.** The documentation serves as a detailed guide and resource hub of FinRL Contests for participants. It includes FinRL introduction, task descriptions and detailed explanations of the starter kit and baseline solutions.

7.4 | Event Promotion

The FinRL Contests welcome students, researchers and practitioners who are passionate about finance and machine learning. The contest is promoted widely through mailing lists of Google Groups and universities, and social media platforms, such as LinkedIn, Twitter, and Facebook, highlighting its inclusive and challenging nature. We also collaborate with some media partners, including Wilmott, PyQuant News and Paris Machine Learning Group.

7.5 | Registration and Submission

Team registration and model submission are through Google Forms for easy management. Registration requires a team name and members' information, including name, email, and affiliation. For model submission, teams provide a link to the GitHub and/or Hugging Face repository. The repository should be well organised, including code, model weights, scripts and detailed instructions for evaluation. The report submission is through OpenReview or the channel required by the conference.

7.6 | Communication Channels

We have multiple channels to communicate with participants for their questions, announcements and tutorials.

- **Contact email.** We use an official contact email, finrl-contest@gmail.com, to send important announcements and respond to enquiries.
- **Group chat.** We set up a Discord channel of 550+ people and 2 WeChat groups of 600+ people to facilitate networking, discussion and direct Q&A.
- **GitHub issue.** Participants are encouraged to create GitHub issues in the starter kit repository to seek technical support.
- **Online sessions.** Three online Q&A sessions were organised via Zoom to address common concerns and questions. One online tutorial session was organised to introduce the FinRL framework, the contest and the usage of the starter kit.

7.7 | Contest Award

We issue official contest certificates recognised by the organising academic conferences and Columbia University to contest winners, as shown in Figure 12. For the FinRL Contest 2025 and FinAI Contest 2025, participants are also given the opportunity to publish a short paper in the conference proceedings. In IEEE IDS 2025, there are 11 papers accepted and published through the special track Financial Reinforcement Learning and Foundation Models (FinRLFM). During the organisation, we also collected feedback from participants regarding the need for financial support for computing resources and data APIs. We have noted this need and will work hard to address it in future contests.

8 | Conclusions and Future Works

In this paper, we present the financial reinforcement learning benchmark through FinRL Contests, held from 2023 to 2025.

FinRL Contest	Task	Winner Team	Affiliation
FinRL 2023	Data-Centric Stock Trading	SZU-FIN-621	Shenzhen University
		Nik-Elena	Columbia University
		WeCan	Tsinghua University, The Hong Kong University of Science and Technology
	Real Time Order Execution	Nik-Elena	Columbia University
		QuantFox	QuantFox Pty Ltd
		DROP TABLE *	Tsinghua University, University of Illinois at Urbana-Champaign
FinRL 2024	Crypto Trading with Ensemble Learning	Fermion	FTI Consulting, Ernst & Young
	LLM-Engineered Signals with RLMF	AetherNet	Purdue University
FinRL 2025	FinRL-DeepSeek for Stock Trading	Otago Alpha	University of Otago
		Ruijian & Sally	Columbia University
		Kuxlnw	Kasetsart University
		Queen's Gambit	National University of Sciences and Technology
	FinRL-AlphaSeek for Crypto Trading	cryptoalphaseek	East China University of Science and Technology
		Mt.Everest	Logictronix Technologies

FIGURE 12 | Winner teams and their affiliations in FinRL Contests (2023–2025). The FinAI Contest 2025 is ongoing as of this writing.

The contests provided standardised task definitions, curated market datasets, GPU-optimised parallel market environments and defined evaluation metrics to ensure reproducibility and comparability across various financial applications such as stock trading, cryptocurrency trading, and the integration of signals from LLMs. Participant teams employed diverse strategies, including feature engineering, ensemble learning methods and LLM-generated market signals, demonstrating significant progress in both performance and risk management compared to conventional market benchmarks.

For future work, we will continue exploring and integrating LLM-engineered signals from multimodal financial data [34] into FinRL, such as SEC filings, earnings conference calls, alternative data, etc. The recent breakthroughs in LLMs, particularly their enhanced reasoning capabilities achieved through reinforcement learning-based post-training, hold significant potential for financial tasks. For instance, FLAG-TRADER [62] integrates LLMs with gradient-based reinforcement learning policy optimisation to enable the adaptation of an LLM's reasoning capabilities to financial tasks, in which a partially fine-tuned LLM acts as the policy network. Therefore, we will further develop FinRL trading agents that can be integrated into real-time intelligent trading systems, capable of processing and reasoning over multimodal data. We will utilise LLMs to process real-time financial data (e.g., financial dataset and Finnhub) into trading signals, which will be put into the state of the trading agent. We will also extend our benchmark to support deep generative models for offline policy learning. We will explore the combination of behavioural cloning and reinforcement learning.

For the contest organisation, we encourage participants to open-source their solutions for sharing. Given that the contest requires submitting the full model development lifecycle, we encourage participants to more standardly release their models using the Model Openness Framework [63] and licence them under OpenMDW.[12] To address the needs of industry participants for private data or model weights, we will utilise zero-knowledge proof [64] techniques to protect data and models, thereby encouraging wider industry participation. In future contests, we will also provide computing resources to support participants and welcome industry collaboration to fund prizes. Furthermore, we will also build a more standardised and transparent leaderboard. We will leverage our experience of the Open FinLLM Leaderboard [65] and FinBen [66], and contribute financial tasks to the Open FinLLM Leaderboard as part of the FinAI benchmarking ecosystem.

We will continue to actively integrate the most cutting-edge techniques with financial applications and improve the contest organisation.

Author Contributions

Keyi Wang contributed to the conceptualization, methodology design, validation, and writing of the manuscript, including both the original draft and revisions. Nikolaus Holzer contributed to the conceptualization, methodology design, validation, and writing of the original draft. Ziyi Xia contributed to the conceptualization, methodology design, validation, and writing of the original draft. Yupen Cao contributed to the writing of the original draft and manuscript revisions. Jiechao Gao contributed to the writing of the original draft and manuscript revisions. Anwar Walid and Kairong Xiao provided supervision of the FinRL contests and paper revisions. Xiao-Yang Liu (Yanglet) contributed to manuscript revisions and provided supervision of the FinRL contests.

Acknowledgements

We thank Li Deng, Zihan Ding, Jian Guo, Zhouchi Lin, Christina Dan Wang, Zhaoran Wang, Matt White, Bo Wu, Xiaojun Wu, Zhuoran Yang, Daochen Zha and Chuheng Zhang for serving as advisors of FinRL Contest 2023–2025. We thank Mostapha Benhenda, Jin Bo, Jiale Chen, Qian Chen, Arnav Grover, Ethan Havemann, Sarah Huang, Colin Lin, Shivan Mukherjee, Jaisal Patel, Charlie Shen, Kent Wu, Yangyang Yu and Andy Zhu for serving as organisers, and Steve Ewald, Andrew Li, Nikola Maruszewski, Christopher Minn and Gavin Wang for creating the evaluation platform. We thank Zihan Dong, Allan Feng, Astarag Mohapatra, Louis Owen, Guoxuan Wang, Zhiyuan Wang, Bruce Yang, Luna Zhang, Jiahao Zheng and Ming Zhu from the open-source AI4Finance community, who contributed to the open-source FinRL project. We also thank Jimin Huang from the FinAI community. We thank Renyuan Xu, Huining Yang and Bo An for academic feedback on FinRL papers. We thank all participant teams!

The authors thank Vatic Investment and the Shanghai Frontiers Science Centre of Artificial Intelligence and Deep Learning at NYU Shanghai for covering registration fees for participant teams.

Keyi Wang, Nikolaus Holzer, Jiechao Gao, Anwar Walid and Xiao-Yang Liu Yanglet acknowledge the support from Columbia's SIRS and STAR Program, as well as The Tang Family Fund for Research Innovations in FinTech, Engineering and Business Operations. Xiao-Yang Liu Yanglet also acknowledges the support from a NSF IUCRC CRAFT center research grant (CRAFT Grant 22017) for this research. The opinions expressed in this publication do not necessarily represent the views of NSF IUCRC CRAFT.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available in FinRL Contest 2023–2025 at https://github.com/Open-Finance-Lab. These data were derived from the following resources available in the public domain:—Yahoo Finance, https://pypi.org/project/yfinance/—BTC LOB data, https://www.kaggle.com/datasets/martinsn/high-frequency-crypto-limit-order-book-data/data—FNSPID, https://huggingface.co/datasets/Zihan1004/FNSPID—Bitcoin News, https://huggingface.co/datasets/edaschau/bitcoin_news—Crypto News, https://github.com/soheilrahsaz/cryptoNewsDataset.

Endnotes

[1] https://yfinance-python.org/.

[2] https://alpaca.markets/.

[3] https://github.com/AI4Finance-Foundation/FinRL-Meta.

[4] https://www.kaggle.com/datasets/martinsn/high-frequency-crypto-limit-order-book-data/data.

[5] The aggregated BTC news dataset: https://huggingface.co/datasets/SecureFinAI-Lab/FinRL_BTC_news_signals. Data sources: https://huggingface.co/datasets/edaschau/bitcoin_news; https://github.com/soheilrahsaz/cryptoNewsDataset.

[6] https://github.com/AI4Finance-Foundation/FinGPT.

[7] GitHub repo for models and reports: https://github.com/Open-Finance-Lab/FinRL_Contest_2023/tree/main/Task_1.

[8] https://github.com/Arnav-Gr0ver/ICAIF_FinRL-2024.

[9] Participant teams' open-source solution: Ruijian and Sally (https://github.com/Ruijian-Zha/FinRL-DAPO-SR), Kuxlnw (https://github.com/Vorakorn1001/FinRL-DeepSeek), Queen's Gambit (https://github.com/sahar-arshad/QueensGambit-FinRL2025).

[10] Participant teams' open-source solution: Mt.Everest (https://github.com/TiwariLaxuu/Finrl_Alphaseek_Crypto), Cryptoalphaseek (https://github.com/hardwayp3nny/task2_crypto_hft).

[11] https://github.com/AI4Finance-Foundation/FinRL_Crypto.

[12] https://openmdw.ai/.

References

1. X. Y. Liu, H. Yang, Q. Chen, et al., "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance," in *Deep Reinforcement Learning Workshop, NeurIPS* (2020).

2. X. Y. Liu, H. Yang, J. Gao, and C. D. Wang, "FinRL: Deep Reinforcement Learning Framework to Automate Trading in Quantitative Finance," in *ACM International Conference on AI in Finance* (Association for Computing Machinery, 2022).

3. X. Y. Liu, Z. Xiong, S. Zhong, H. Yang, and A. Walid, "Practical Deep Reinforcement Learning Approach for Stock Trading," in *Workshop on Challenges and Opportunities for AI in Financial Services, NeurIPS* (2018).

4. Z. Li, X. Y. Liu, J. Zheng, Z. Wang, A. Walid, and J. Guo, "FinRL-Podracer: High Performance and Scalable Deep Reinforcement Learning for Quantitative Finance," in *ACM International Conference on AI in Finance* (Association for Computing Machinery, 2021), 1–9.

5. X. Y. Liu, Z. Xia, J. Rui, et al., "FinRL-Meta: Market Environments and Benchmarks for Data-Driven Financial Reinforcement Learning," *Advances in Neural Information Processing Systems* 35 (2022): 1835–1849.

6. X. Y. Liu, Z. Xia, H. Yang, et al., "Dynamic Datasets and Market Environments for Financial Reinforcement Learning," *Machine Learning - Nature* 113, no. 5 (2024): 2795–2839, https://doi.org/10.1007/s10994-023-06511-w.

7. B. Hambly, R. Xu, and H. Yang, "Recent Advances in Reinforcement Learning in Finance," *Mathematical Finance* 33, no. 3 (2023): 437–503, https://doi.org/10.1111/mafi.12382.

8. A. Charpentier, R. Elie, and C. Remlinger, "Reinforcement Learning in Economics and Finance," *Computational Economics* 62 (2021): 1–38, https://doi.org/10.1007/s10614-021-10119-4.

9. T. G. Fischer, Reinforcement Learning in Financial Markets—A Survey. FAU Discussion Papers in Economics (Friedrich-Alexander-Universität Erlangen-Nürnberg, Institute for Economics, 2018).

10. S. Sun, R. Wang, and B. An, "Reinforcement Learning for Quantitative Trading," *ACM Transactions on Intelligent Systems and Technology* 14, no. 3 (2023): 1–29, https://doi.org/10.1145/3582560.

11. Y. Bai, Y. Gao, R. Wan, S. Zhang, and R. Song, "A Review of Reinforcement Learning in Financial Applications," *Annual Review of Statistics and Its Application* 12, no. 1 (2025): 209–232, https://doi.org/10.1146/annurev-statistics-112723-034423.

12. M. Towers, A. Kwiatkowski, J. Terry, et al., "Gymnasium: A Standard Interface for Reinforcement Learning Environments," *arXiv:2407.17032* (2024).

13. R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct Preference Optimization: Your Language Model Is Secretly a Reward Model," *Advances in Neural Information Processing Systems* 36 (2023): 53728–53741.

14. J. Achiam, Spinning Up in Deep Reinforcement Learning (2018), https://spinningup.openai.com.

15. C. Ying, X. Zhou, D. Yan, and J. Zhu, "Towards Safe Reinforcement Learning via Constraining Conditional Value at Risk," in *ICML 2021 Workshop on Adversarial Machine Learning* (2022).

16. 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* 28, no. 3 (2016): 653–664, https://doi.org/10.1109/tnnls.2016.2522401.

17. L. Avramelou, P. Nousi, N. Passalis, and A. Tefas, "Deep Reinforcement Learning for Financial Trading Using Multi-Modal Features," *Expert Systems with Applications* 238 (2024): 121849, https://doi.org/10.1016/j.eswa.2023.121849.

18. H. Yang, X. Y. Liu, S. Zhong, and A. Walid, "Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy," in *ACM International Conference on AI in Finance* (Association for Computing Machinery, 2020).

19. Z. Zhang, S. Zohren, and S. Roberts, "Deep Reinforcement Learning for Trading," *Journal of Financial Data Science.* 2, no. 2 (2020): 25–40, https://doi.org/10.3905/jfds.2020.1.030.

20. L. Ardon, N. Vadori, T. Spooner, M. Xu, J. Vann, and S. Ganesh, "Towards a Fully RL-Based Market Simulator," in *ACM International Conference on AI in Finance (ICAIF)* (Association for Computing Machinery, 2021).

21. S. Amrouni, A. Moulin, J. Vann, S. Vyetrenko, T. Balch, and M. Veloso, "ABIDES-Gym: Gym Environments for Multi-Agent Discrete Event Simulation and Application to Financial Markets," in *ACM International Conference on AI in Finance (ICAIF)* (Association for Computing Machinery, 2021).

22. A. Coletta, M. Prata, M. Conti, et al., "Towards Realistic Market Simulations: A Generative Adversarial Networks Approach," in *ACM International Conference on AI in Finance (ICAIF)* (Association for Computing Machinery, 2021).

23. X. Y. Liu, G. Wang, H. Yang, and D. Zha, "FinGPT: Democratizing Internet-Scale Data for Financial Large Language Models," in *Workshop on Instruction Tuning and Instruction Following, NeurIPS* (2023).

24. Y. Nie, Y. Kong, X. Dong, et al., "A Survey of Large Language Models for Financial Applications: Progress, Prospects and Challenges," *arXiv:2406.11903* (2024).

25. P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep Reinforcement Learning From Human Preferences," *Advances in Neural Information Processing Systems* 30 (2017).

26. M. Benhenda, "FinRL-DeepSeek: LLM-Infused Risk-Sensitive Reinforcement Learning for Trading Agents," *arXiv:2502.07393* (2025).

27. B. J. D. Gort, X. Y. Liu, J. Gao, S. Chen, and C. D. Wang, "Deep Reinforcement Learning for Cryptocurrency Trading: Practical Approach to Address Backtest Overfitting," in *AAAI Bridge on AI for Financial Services* (2023).

28. J. Peters and J. A. Bagnell, "Policy Gradient Methods," in *Encyclopedia of Machine Learning* (Springer, 2010), 774–776.

29. S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, "Monte Carlo Gradient Estimation in Machine Learning," *Journal of Machine Learning Research* 21, no. 1 (2020).

30. J. Chen, B. Ganguly, Y. Xu, Y. Mei, T. Lan, and V. Aggarwal, "Deep Generative Models for Offline Policy Learning: Tutorial, Survey, and Perspectives on Future Directions," *Transactions on Machine Learning Research* (2024).

31. K. Wu, Z. Xia, S. Chen, and X. Y. Liu, "Curriculum Learning From Smart Retail Investors: Towards Financial Open-Endedness," in *2nd Workshop on Agent Learning in Open-Endedness, NeurIPS* (2023).

32. Z. Dong, X. Fan, and Z. Peng, "Fnspid: A Comprehensive Financial News Dataset in Time Series," in *Proceedings of the 30th ACM SIGKDD*

Conference on Knowledge Discovery and Data Mining (Association for Computing Machinery, 2024).

33. Z. Kakushadze, "101 Formulaic Alphas," *arXiv preprint arXiv:1601.00991* 2016, no. 84 (2016): 72–81, https://doi.org/10.1002/wilm.10525.

34. X. Y. L. Yanglet, Y. Cao, and L. Deng, "Multimodal Financial Foundation Models (MFFMs): Progress, Prospects, and Challenges," *arXiv preprint arXiv:2506.01973* (2025).

35. Y. Cao, Z. Chen, P. Kumar, et al., "RiskLabs: Predicting Financial Risk Using Large Language Model Based on Multimodal and Multi-Sources Data," *arXiv preprint arXiv:2404.07452* (2024).

36. Y. Cao, Z. Chen, Q. Pei, N. Lee, K. Subbalakshmi, and P. M. Ndiaye, "ECC Analyzer: Extracting Trading Signal From Earnings Conference Calls Using Large Language Model for Stock Volatility Prediction," in *Proceedings of the 5th ACM International Conference on AI in Finance* (Association for Computing Machinery, 2024), 257–265.

37. X. Y. Liu, J. Zhang, G. Wang, W. Tong, and A. Walid, "FinGPT-HPC: Efficient Pretraining and Finetuning Large Language Models for Financial Applications With High-Performance Computing," *arXiv preprint arXiv:2402.13533* (2024).

38. B. Zhang, H. Yang, and X. Y. Liu, "Instruct-FinGPT: Financial Sentiment Analysis by Instruction Tuning of General-Purpose Large Language Models," *arXiv preprint arXiv:2306.12659* (2023).

39. B. Zhang, H. Yang, T. Zhou, M. Ali Babar, and X. Y. Liu, "Enhancing Financial Sentiment Analysis via Retrieval Augmented Large Language Models," in *Proceedings of the 4th ACM International Conference on AI in Finance* (Association for Computing Machinery, 2023), 349–356.

40. K. Ouyang, Y. Liu, S. Li, R. Bao, K. Harimoto, and X. Sun, "Modal-Adaptive Knowledge-Enhanced Graph-Based Financial Prediction From Monetary Policy Conference Calls With LLM," in *Proceedings of the Joint Workshop of the 7th Financial Technology and Natural Language Processing, the 5th Knowledge Discovery from Unstructured Data in Financial Services, and the 4th Workshop on Economics and Natural Language Processing@ LREC-COLING 2024* (Association for Computational Linguistics, 2024), 59–69.

41. S. Han, H. Kang, B. Jin, X. Y. Liu, and S. Y. Yang, "XBRL Agent: Leveraging Large Language Models for Financial Report Analysis," in *Proceedings of the 5th ACM International Conference on AI in Finance* (Association for Computing Machinery, 2024), 856–864.

42. S. Lin, K. Wang, and X. Y. Liu, "Analyzing Cascading Outbreak of GameStop Event: A Practical Approach Using Network Analysis and Large Language Models," in *Proceedings of the 5th ACM International Conference on AI in Finance* (Association for Computing Machinery, 2024), 428–436.

43. J. Guo, S. Wang, L. M. Ni, and H. Y. Shum, "Quant 4.0: Engineering Quantitative Investment With Automated, Explainable, and Knowledge-Driven Artificial Intelligence," *Frontiers of Information Technology & Electronic Engineering.* 25, no. 11 (2024): 1421–1445, https://doi.org/10.1631/fitee.2300720.

44. B. Cao, S. Wang, X. Lin, et al., "From Deep Learning to LLMs: A Survey of AI in Quantitative Investment," *arXiv preprint arXiv:2503.21422* (2025).

45. A. Liu, B. Feng, B. Xue, et al., "DeepSeek-v3 Technical Report," *arXiv preprint arXiv:2412.19437* (2024).

46. H. Markowitz, "Portfolio Selection," *Journal of Finance* 7, no. 1 (1952): 77–91, https://doi.org/10.1111/j.1540-6261.1952.tb01525.x.

47. M. A. Wiering and vH. Hasselt, "Ensemble Algorithms in Reinforcement Learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 38, no. 4 (2008): 930–936.

48. F. Pérez-Cruz, "Kullback-Leibler Divergence Estimation of Continuous Distributions," in *2008 IEEE International Symposium on Information Theory* (IEEE, 2008), 1666–1670.

49. V. Mnih, K. Kavukcuoglu, D. Silver, et al., "Human-Level Control Through Deep Reinforcement Learning," *Nature* 518, no. 7540 (2015): 529–533, https://doi.org/10.1038/nature14236.

50. H. Hv, A. Guez, and D. Silver, "Deep Reinforcement Learning With Double Q-Learning," in *AAAI Conference on Artificial Intelligence* (AAAI Press, 2016), 2094–2100.

51. Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling Network Architectures for Deep Reinforcement Learning," *International Conference on International Conference on Machine Learning* 48 (2016): 1995–2003.

52. M. Ganaie, M. Hu, A. Malik, M. Tanveer, and P. Suganthan, "Ensemble Deep Learning: A Review," *Engineering Applications of Artificial Intelligence* 115, no. C (2022): 105151, https://doi.org/10.1016/j.engappai.2022.105151.

53. J. Yin, R. Wang, Y. Guo, et al., "Wealth Flow Model: Online Portfolio Selection Based on Learning Wealth Flow Matrices," *ACM Transactions on Knowledge Discovery from Data* 16, no. 2 (2021): 1–27, https://doi.org/10.1145/3464308.

54. P. J. Kremer, S. Lee, M. Bogdan, and S. Paterlini, "Sparse Portfolio Selection via the Sorted L1-Norm," *Journal of Banking & Finance* 110 (2020): 105687, https://doi.org/10.1016/j.jbankfin.2019.105687.

55. A. Grover, "FinRLlama: A Solution to LLM-Engineered Signals Challenge at FinRL Contest 2024," *arXiv:2502.01992* (2025).

56. S. Li, M. Yu, and F. Dossor, "Option-Driven Sentiment in FinRL: A PPO Approach to Trading," in *IEEE 11th International Conference on Intelligent Data and Security (IDS)* (IEEE Computer Society, 2025), 62–64.

57. R. Zha and B. Liu, "A New DAPO Algorithm for Stock Trading," in *IEEE 11th International Conference on Intelligent Data and Security (IDS)* (IEEE Computer Society, 2025), 46–48.

58. Q. Yu, Z. Zhang, R. Zhu, et al., "DAPO: An Open-Source LLM Reinforcement Learning System at Scale," *arXiv preprint arXiv:503.14476* (2025).

59. V. Kosidphokin, P. Loedtrakunchai, N. Sinamnuaiphon, and S. Kuptanon, "FinRL: Adaptive Model Selection for Reinforcement Learning in Stock Trading," in *IEEE 11th International Conference on Intelligent Data and Security (IDS)* (IEEE Computer Society, 2025), 71–72.

60. S. Arshad, H. Ameer, N. Azhar, and S. Latif, "FinRL Contest 2025 Task 1: Market-Aware In-Context Learning Framework for Proximal Policy Optimization in Stock Trading Using DeepSeek," in *IEEE 11th International Conference on Intelligent Data and Security (IDS)* (IEEE Computer Society, 2025), 76–78.

61. J. C. Liu, J. C. Ma, and Z. Q. Jiang, "AlphaSeek FinRL: A Hybrid Deep Learning Architecture for High-Frequency Cryptocurrency Trading," in *IEEE 11th International Conference on Intelligent Data and Security (IDS)* (IEEE Computer Society, 2025), 55–57.

62. G. Xiong, Z. Deng, K. Wang, et al., "FLAG-Trader: Fusion LLM-Agent With Gradient-Based Reinforcement Learning for Financial Trading," *arXiv preprint arXiv:2502.11433* (2025): 13921–13934, https://doi.org/10.18653/v1/2025.findings-acl.716.

63. M. White, I. Haddad, C. Osborne, et al., "The Model Openness Framework: Promoting Completeness and Openness for Reproducibility, Transparency, and Usability in Artificial Intelligence," *arXiv preprint arXiv:2403.13784* (2024).

64. H. Sun, J. Li, and H. Zhang, "zkLLM: Zero Knowledge Proofs for Large Language Models," in *ACM SIGSAC Conference on Computer and Communications Security* (Association for Computing Machinery, 2024).

65. S. C. Lin, F. Tian, K. Wang, et al., "Open FinLLM Leaderboard: Towards Financial AI Readiness," in *International Workshop on Multimodal Financial Foundation Models (MFFMs) at 5th ACM International Conference on AI in Finance* (2024).

66. Q. Xie, W. Han, Z. Chen, et al., "FinBen: A Holistic Financial Benchmark for Large Language Models," *Advances in Neural Information Processing Systems* 37 (2024): 95716–95743.