FinRL-Meta: A Universe of Near-Real Market Environments for Data-Driven Deep Reinforcement Learning in Quantitative Finance

Xiao-Yang Liu[1], **Jingyang Rui**[2], **Jiechao Gao**[3], **Liuqing Yang**[1], **Hongyang Yang**[1],
Zhaoran Wang[4], **Christina Dan Wang**[5], **Jian Guo**[6]
[1]Columbia University; [2]The University of Hong Kong; [3]University of Virginia;
[4]Northwestern University; [5]New York University (Shanghai); [6]IDEA Research.
XL2427@columbia.edu; zhaoranwang@gmail.com; guojian@idea.edu.cn

Abstract

Deep reinforcement learning (DRL) has shown huge potentials in building financial market simulators recently. However, due to the highly complex and dynamic nature of real-world markets, raw historical financial data often involve large noise and may not reflect the future of markets, degrading the fidelity of DRL-based market simulators. Moreover, the accuracy of DRL-based market simulators heavily relies on numerous and diverse DRL agents, which increases demand for a universe of market environments and imposes a challenge on simulation speed. In this paper, we present a FinRL-Meta framework that builds a universe of market environments for data-driven financial reinforcement learning. First, FinRL-Meta separates financial data processing from the design pipeline of DRL-based strategy and provides open-source data engineering tools for financial big data. Second, FinRL-Meta provides hundreds of market environments for various trading tasks. Third, FinRL-Meta enables multiprocessing simulation and training by exploiting thousands of GPU cores. Our codes are available online at `https://github.com/AI4Finance-Foundation/FinRL-Meta`.

1 Introduction

In quantitative finance, market simulators play important roles in studying the complex market phenomena and investigating financial regulations [1, 2]. Compared to traditional simulation models, deep reinforcement learning (DRL) has shown huge potentials in building financial market simulators through multi-agent systems [3]. However, due to the high complexity of real-world markets, raw historical financial data involve significant noise and may not reflect the future of markets. This issue usually degrades the fidelity of DRL-based simulation. Moreover, in reality, there are innumerable participators that impact together on the market. To better simulate the market, numerous and diverse DRL agents are needed to represent those participators with different aims and strategies.

Recently, researchers have explored various applications of DRL in quantitative finance [3, 4, 5, 6]. Lussange et al. [3] have proposed a market simulation model using multi-agent reinforcement learning. Although it has shown the feasibility of DRL-based market simulation, only a few DRL agents are used. The potentials of DRL-based market simulators are not fully explored yet. The FinRL framework [4] has proposed a DRL framework as a full pipeline of developing trading strategies and is growing an open-source community *AIFinannce* that can contribute diverse DRL agents. However, it [4] focuses on developing trading strategies instead of building market simulations.

In this paper, we develop a FinRL-Meta framework that is a universe of near real-market environments for data-driven financial reinforcement learning. First, we apply the DataOps paradigm

Workshop on Data Centric AI, 35th Conference on Neural Information Processing Systems (NeurIPS 2021).

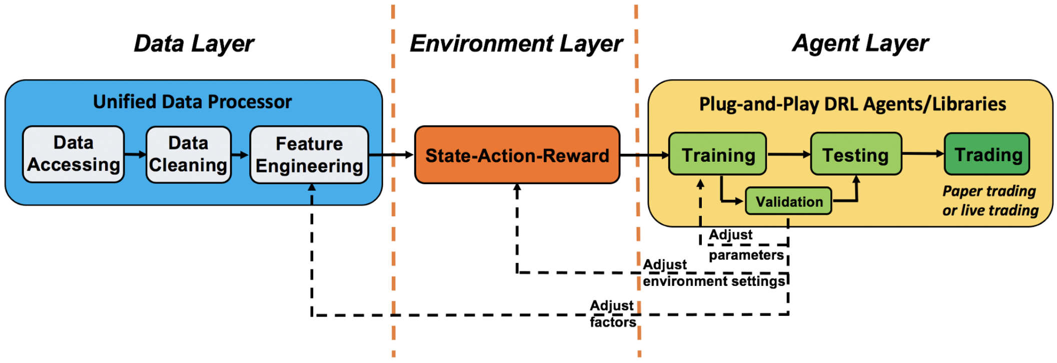

Figure 1: Overview of FinRL-Meta.

Data Source	Type	Range and Frequency	Request Limits	Raw Data	Preprocessed Data
Yahoo! Finance	US Securities	Frequency-specific, 1min	2,000/hour	OHLCV	Prices & Indicators
CCXT	Cryptocurrency	API-specific, 1min	API-specific	OHLCV	Prices & Indicators
WRDS.TAQ	US Securities	2003-now, 1ms	5 requests each time	Intraday Trades	Prices & Indicators
Alpaca	US Stocks, ETFs	2015-now, 1min	Account-specific	OHLCV	Prices & Indicators
RiceQuant	CN Securities	2005-now, 1ms	Account-specific	OHLCV	Prices & Indicators
JoinQuant	CN Securities	2005-now, 1min	3 requests each time	OHLCV	Prices & Indicators
QuantConnect	US Securities	1998-now, 1s	NA	OHLCV	Prices & Indicators

Table 1: Data platforms. OHLCV means open, high, low, close, volume data.

[7] to the data engineering pipeline, providing agility to agent deployment. We offer a unified and automated data processor for data accessing, data cleaning and feature engineering. Second, we build hundreds of near real-market DRL environments for various trading tasks such as high-frequency trading, cryptocurrencies trading, stock portfolio allocation, etc.. The environments are directly connected to our data processor. High-quality large datasets can be generated efficiently and encapsulated into our environments. Third, to accelerate the training process of DRL agents in large datasets, we utilize thousands of GPU cores to perform multiprocessing training.

2 Proposed FinRL-Meta Framework

MDP Model for Trading Tasks: We model a trading task as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$ [8], where $\mathcal{S}$ and $\mathcal{A}$ denote the state space and action space, respectively, $P(s'|s, a)$ denotes the transition probability, $r(s, a)$ is a reward function, and $\gamma \in (0, 1)$ is a discount factor. Specifically, the state denotes an observation that a DRL agent receives from a market environment; the action space consists of actions that an agent is allowed to take at a state; the reward function $r(s, a, s')$ is the incentive for agents to learn a better policy. A trading agent aims to learn a policy $\pi(s_t|a_t)$ that maximizes the expected return $R = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$.

Overview of FinRL-Meta: We utilize a layered structure, as shown in Fig. 1. FinRL-Meta consists of three layers: data layer, environment layer, and agent layer. Each layer executes its functions and is relatively independent. Meanwhile, layers interact through end-to-end interfaces to implement the complete workflow of algorithm trading.

DataOps for Data-Driven DRL in Finance: We follow the DataOps paradigm [7] in the data layer. First, we establish a standard pipeline for financial data engineering, ensuring data of different formats from different sources can be incorporated in a unified RL framework. Second, we automate this pipeline with a data processor, which can access data, clean data and extract features from various data sources with high quality and efficiency. Our data layer provides agility to model deployment. The data sources are shown in Table 1.

Multiprocessing Training: We utilize thousands of GPU cores to perform multiprocessing training, which significantly accelerates the training process. In each CUDA core, a trading agent interacts with a market environment to produce transitions in the form of (state, action, reward, next state). Then all the transitions are stored in a replay buffer and used to update a learner and evaluator. By adopting this technique, we successfully achieve multiprocessing simulation of hundreds of market environments to improve the performance of DRL trading agents on large datasets.

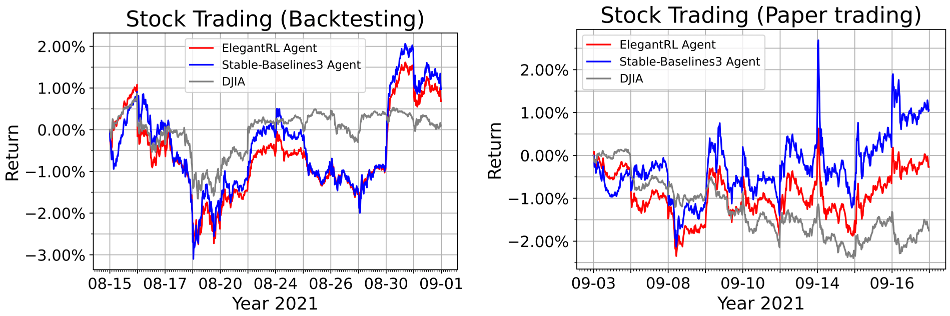

Figure 2: Cumulative returns (5-minute) of stock trading in backtesting and paper trading.

	ElegantRL [10]	**Stable-baselines3** [11]	**DJIA**
Cumul. return	0.968% / -0.652%	1.335% / 0.191%	0.099% / -1.56%
Annual return	22.425% / -16.746%	32.106% / 5.492%	2.108% / -35.522%
Annual volatility	15.951% / 14.113%	19.871% / 15.953%	9.196% / 9.989%
Sharpe ratio	1.457 / -1.399	1.621 / 0.447	0.289 / -4.894
Max drawdown	-2.657% / -1.871%	-2.932% / -1.404%	-1.438% / -2.220%

Table 2: Performance of backtesting (red) and paper trading (blue) for stock trading.

Plug-and-Play: In the development pipeline, we separate market environments from the data layer and the agent layer. Any DRL agent can be directly plugged into our environments, then trained and tested. Different agents can run on the same benchmark environment for fair comparison.

Training-Testing-Trading Pipeline: We employ a training-testing-trading pipeline. The DRL agent first learns from the training environment and is then validated in the validation environment for further adjustment. Then the validated agent is tested on historical datasets. Finally, the tested agent will be deployed in paper trading or live trading markets. First, this pipeline solves the information leakage problem because the trading data are generated yet when adjusting agents. Second, a unified pipeline allows fair comparisons among different trading strategies.

Supported Trading Tasks: We have supported and achieved satisfactory trading performance for trading tasks such as stock trading, cryptocurrency trading, and portfolio allocation. Derivatives such as futures and forex are also supported. Besides, we have supported multi-agent simulation and execution optimizing tasks by reproducing the experiment in [9].

3 Performance Evaluation

To provide benchmarks for researchers, we will continuously add typical trading tasks with corresponding environments. Here, we show results of stock trading and cryptocurrency trading.

3.1 Experiment Settings

Stock trading task: We select the 30 constituent stocks in Dow Jones Industrial Average (DJIA), accessed at the beginning our testing period. We use the Proximal Policy Optimization (PPO) algorithm [12] of ElegantRL [10], Stable-baselines3 [11] and RLlib [13], respectively, to train agents and use the DJIA index as the baseline. We use 1-minute data from 06/01/2021 to 08/14/2021 for training and data from 08/15/2021 to 08/31/2021 for validation (backtesting). Then we retrain the agent using data from 06/01/2021 to 08/31/2021 and conduct paper trading from 09/03/2021 to 09/16/2021. The historical data and real-time data are accessed from the Alpaca's database and paper trading APIs.

Cryptocurrency trading task: We select top 10 market cap cryptocurrencies [1]. We use the PPO algorithm [12] of ElegantRL [10] to train an agent and use the Bitcoin (BTC) price as the baseline. We use 5-minute data from 06/01/2021 to 08/14/2021 for training and data from 08/15/2021 to 08/31/2021 for validation (backtesting). Then we retrain the agent using data from 06/01/2021

[1] The top 10 market cap cryptocurrencies as of Oct 2021 are: Bitcoin (BTC), Ethereum (ETH), Cardano (ADA), Binance Coin (BNB), Ripple (XRP), Solana (SOL), Polkadot (DOT), Dogecoin (DOGE), Avalanche (AVAX), Uniswap (UNI). Tether (USDT) and USD Coin (USDC) are excluded.

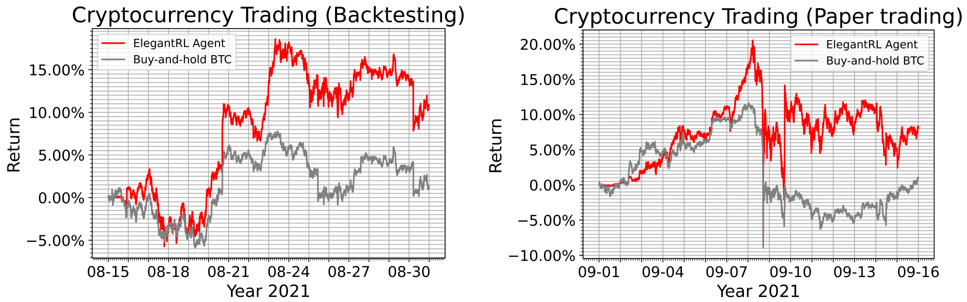

Figure 3: Cumulative returns (5-minute) of cryptocurrency trading in backtesting and paper trading.

	ElegantRL [10]	**BTC buy and hold**
Cumul. return	10.857% / 4.844%	1.332% / -1.255%
Annual return	360.823% / 121.380%	21.666% / 5.492%
Annual volatility	59.976% / 65.857%	47.410% / 57.611%
Sharpe ratio	2.992 / 1.608	0.657 / -0.113
Max drawdown	-6.396% / -10.474%	-7.079% / -14.849%

Table 3: Performance of backtesting (red) and paper trading (blue) for cryptocurrency trading.

to 08/31/2021 and conduct paper trading from 09/01/2021 to 09/15/2021. The historical data and real-time data are accessed from Binance.

3.2 Trading Performance

Stock trading: In the backtesting stage, both ElegantRL [10] agent and Stable-baselines3 [11] agent outperform DJIA in annual return and Sharpe ratio, as shown in Fig. 2 and Table 2. The ElegantRL agent achieves an annual return of 22.425% and a Sharpe ratio of 1.457. The Stable-baselines3 agent achieves an annual return of 32.106% and a Sharpe ratio of 1.621. In the paper trading stage, the results are consistent with the backtesting results. Both the ElegantRL agent and the Stable-baselines3 agent outperform the baseline.

Cryptocurrency trading: In the backtesting stage, the ElegantRL agent outperforms the benchmark (BTC price) in most performance metrics, as shown in Fig. 3 and Table 3. It achieves an annual return of 360.823% and a Sharpe ratio of 2.992. The ElegantRL agent also outperforms the benchmark (BTC price) in the paper trading stage, which is consistent with the backtesting results.

4 Conclusion

In this paper, we followed the DataOps paradigm and developed a FinRL-Meta framework. FinRL-Meta provides open-source data engineering tools and hundreds of market environments with multiprocessing simulation.

For future work, we are building a multi-agent based market simulator that consists of over ten thousands of agents, namely, a FinRL-Metaverse. First, FinRL-Metaverse aims to build a universe of market environments, like the Xland environment [14] and planet-scale climate forecast [15] by DeepMind. To improve the performance for large-scale markets, we will employ GPU-based massive parallel simulation as Isaac Gym [16]. Moreover, it will be interesting to explore the evolutionary perspective [17][18][19][20] to simulate the markets. We believe that FinRL-Metaverse will provide insights into complex market phenomena and offer guidance for financial regulations.

References

[1] Marco Raberto, Silvano Cincotti, Sergio M Focardi, and Michele Marchesi. Agent-based simulation of a financial market. *Physica A: Statistical Mechanics and its Applications*, 299(1-2):319–327, 2001.

[2] Takanobu Mizuta. A brief review of recent artificial market simulation (agent-based model) studies for financial market regulations and/or rules. *Available at SSRN 2710495*, 2016.

[3] Johann Lussange, Ivan Lazarevich, Sacha Bourgeois-Gironde, Stefano Palminteri, and Boris Gutkin. Modelling stock markets by multi-agent reinforcement learning. *Computational Economics*, 57(1):113–147, 2021.

[4] Xiao-Yang Liu, Hongyang Yang, Jiechao Gao, and Christina Dan Wang. FinRL: Deep reinforcement learning framework to automate trading in quantitative finance. *ACM International Conference on AI in Finance (ICAIF)*, 2021.

[5] Michaël Karpe, Jin Fang, Zhongyao Ma, and Chen Wang. Multi-agent reinforcement learning in a realistic limit order book market simulation. *ACM International Conference on AI in Finance*, 2020.

[6] Tidor-Vlad Pricope. Deep reinforcement learning in quantitative algorithmic trading: A review. *arXiv preprint arXiv:2106.00123*, 2021.

[7] Crystal Valentine and William Merchan. DataOps: An agile methodology for data-driven organizations. `https://www.oracle.com/a/ocom/docs/oracle-ds-data-ops-map-r.pdf`, 2018.

[8] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[9] Wenhang Bao and Xiao yang Liu. Multi-agent deep reinforcement learning for liquidation strategy analysis, 2019.

[10] Xiao-Yang Liu, Zechu Li, Zhaoran Wang, and Jiahao Zheng. ElegantRL: A lightweight and stable deep reinforcement learning library. `https://github.com/AI4Finance-Foundation/ElegantRL`, 2021.

[11] Antonin Raffin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. Stable baselines3. `https://github.com/DLR-RM/stable-baselines3`, 2019.

[12] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 07 2017.

[13] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. RLlib: Abstractions for distributed reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 3053–3062. PMLR, 10–15 Jul 2018.

[14] Adam Stooke DeepMind-OEL, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, Nat McAleese, et al. Open-ended learning leads to generally capable agents. *arXiv preprint arXiv:2107.12808*, 2021.

[15] Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, Rachel Prudden, Amol Mandhane, Aidan Clark, Andrew Brock, Karen Simonyan, Raia Hadsell, Niall Robinson, Ellen Clancy, Alberto Arribas, and Shakir Mohamed. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, Sep 2021.

[16] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High performance GPU-based physics simulation for robot learning, 2021.

[17] Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. Embodied intelligence via learning and evolution. *Nature Communications*, 2021.

[18] Maarten P Scholl, Anisoara Calinescu, and J Doyne Farmer. How market ecology explains market malfunction. *Proceedings of the National Academy of Sciences*, 118(26), 2021.

[19] Zechu Li, Xiao-Yang Liu, jiahao Zheng, Zhaoran Wang, Anwar Walid, and Jian Guo. FinRL-Podracer: High performance and scalable deep reinforcement learning for quantitative finance. *ACM International Conference on AI in Finance (ICAIF)*, 2021.

[20] Xiao-Yang Liu, Zechu Li, Zhuoran Yang, Jiahao Zheng, Zhaoran Wang, Anwar Walid, Jian Guo, and Michael Jordan. ElegantRL-Podracer: Scalable and elastic library for cloud-native deep reinforcement learning. *Deep RL Workshop, NeurIPS 2021*, 2021.