Standard Market Environments for Financial Reinforcement Learning

Chunlin Feng[1], Lijian Huang[1], Keyi Wang[2], Yupeng Cao[3], Ming Zhu[2], Jiechao Gao[2,4], Xiao-Yang Liu[2]*
[1]Rensselaer Polytechnic Institute [2]Columbia University [3]Stevens Institute of Technology [4]Stanford University
chunlinfeng0920@gmail.com, huangl9@rpi.edu, kw2914@columbia.edu, ycao33@stevens.edu, zhumingpassional@gmail.com, jiechao@stanford.edu, XL2427@columbia.edu

Abstract

Market environments play a critical role in obtaining a robust trading agent. However, current open-source market environments are unorganized, hindering reproduction in the community. In this paper, we organize four market environments. First, we introduce two standard evaluation pipelines, corresponding to stock and crypto markets. Each pipeline integrates financial data, performance metrics, and baseline strategies. Within these pipelines, two stock market environments and two crypto market environments are evaluated. They are compared with the mean-variance porfolio allocation strategy, equal-weight strategy, and market indices, DJIA (stock) and S&P BDA (crypto). The environments are standardized to Gymnasium-Style with documentation provided. Both stock environments outperform DJIA and equal-weight strategy in annual return, and both crypto environments outperform all three baselines in annual return.

1 Introduction

In recent years, financial reinforcement learning (FinRL [5]) has shown great potentials in designing trading strategies [3][10]. The noisy and complex real-world markets makes FinRL a compelling approach to learn the optimal trading strategy. Market environments are needed to reduce the simulation-to-reality gap.

Despite the importance, many open-source market environments do not have good maintenance. Differences in data sources and API design make it difficult to integrate and test agents across environments. Moreover, community initiatives such as FinRL Contests [13] require standardized market environments to ensure fair and reproducible evaluation of agents. The challenges highlights the need to organize open-source market environments.

To address these challenges, we make the following contributions:

- We developed two standard evaluation pipelines integrating financial data, baseline strategies, and performance metrics.
- We evaluated two stock market environments and two crypto market environments.
- We release our experimental results publicly and also the corresponding documentation.

2 Motivation

We conducted a survey of open-source market environments on a set of repository and environment attributes. Table 3 reveals the following issues.

- Maintenance status (e.g., last commit age, tutorials)
  - Some repositories show low commit activities or few approved pull requests, indicating low ongoing support and maintenance. Outdated environments may not work with modern libraries.

*Corresponding author.

Attribute	stock_market_RL [4]	Gym-Anytrading [1]	TradingGym [15]	TensorTrade [11]	Microsoft Qlib [14]	Gym-Trading-Env [2]	TradeMaster [9]	FinRL-Meta [6]
Repository Attributes								
General Info								
Number of Envs	1	3	4	36	8	2	9	100+
Supported Market Types	Stock	Stock/FX	Stock	Stock	Stock/Crypto/FX	Stock	Stock/Crypto	Stock/Crypto/FX
Supported Trading Tasks	AT	AT	AT	AT/PM/OE/MM	PM/OE	AT	AT/PM/OE/MM	AT/PM/OE/MM
Supported Data Sources	3	2	0	1	4	1	1	12
Tutorials	✗	✓	✓	✓	✓	✓	✓	✓
Maintenance								
Last Commit (Age)	>9y	>2y	>2y	>1y	<6m	<5m	<1m	<3w
Approved PRs (6 mo.)	0	0	0	0	22	2	2	8
Release Date (months)	108	73	101	72	35	31	31	40
GitHub Stars	804	≈2.3k	≈1.8k	≈5.4k	≈30.8k	443	≈2.0k	≈1.6k
GitHub Forks	317	489	358	≈1.1k	≈4.4k	96	407	694
Github Issues (Open/Closed)	18/0	9/89	8/4	47/204	257/748	14/8	19/57	74/55
Standard	gym	gymnasium	none	gymnasium	gym	gymnasium	gym	gym
Parallel Environments	✗	✗	✗	✓	✓	✓	✓	✓
Environment Attributes								
State Space								
Balances; Shares	✓	✓	✗	✓	✓	✓	✓	✓
OHLCV Data	✓	✓	✓	✓	✓	✓	✓	✓
LOB Data	✗	✗	✗	✗	✗	✗	✓	–
Technical Indicators	✗	✗	✗	✗	✗	✓	✓	✓
Fundamental Indicators	✗	✗	✗	✗	✗	✗	✗	✗
Multiple Assets	✗	✗	✗	✓	✓	✗	✓	✓
Action Space								
Buy/Sell/Hold	✗	✗	✗	✓	✓	✗	✓	✓
Short/Long	✓	✓	✓	✗	✗	✓	✗	✗
Reward								
Portfolio Value Change	✓	✓	✗	✓	✓	✓	✓	✓
Sharpe Ratio	✗	✗	✗	✓	✗	✗	✗	✗
Risk Penalty	✗	✗	✗	✓	✗	✗	✗	–

Table 1: Comparison of open-source market environments sorted by last commit age (oldest to newest). Key: – means the attribute is partially present, FX: Foreign Exchange, AT: Algorithmic Trading, PM: Portfolio Management, OE: Order Execution, MM: Market Making.

  - Several lack documentation and tutorials needed for new users to adopt.

- Coverage and diversity (e.g., market types and number of environments and data sources)
  - Some repositories have limited use of available mainstream data sources, which hinder users from using data of different market types and time frames.
  - Some repositories lack support of other markets besides the stock market, such as crypto and forex.

- Environment design (state space, action space, and reward)
  - State space, action space, and reward vary between environments, so a clear organization and classification would provide insight for users.
  - Some repositories do not support tasks such as multiple-asset portfolio management, limiting their applicability.

These observations emphasize the need to organize the market environments. To meet the demand, we develop standard evaluation pipelines for stock and crypto trading tasks to evaluate 4 selected market environments.

3 Evaluation Pipelines for Market Environments

Our work introduces two standard evaluation pipelines for stock and crypto market environments respectively.

3.1 Pipeline for Stock Market Environments

Stock Trading Tasks: We evaluate the stock market environments on algorithmic trading. Transaction cost for both buy and sell are set to 0.1%.

Data Preprocessing: We load daily OHLCV data of Apple Stock (AAPL) from YahooFinance, and apply a standardized preprocessing pipeline. This includes filling missing values and adding turbulence index. Two technical indicators, Moving Average Convergence Divergence (MACD) and Relative Strength Index (RSI), are also added to the dataset. It is then split into training, validation, and testing sets, as illustrated by Fig. 1. We use data from 01/01/2009 to 12/31/2021 for training, data from 01/01/2022 to 12/31/2022 for validation, and data from 01/01/2023 to 12/31/2024 for testing.

Training-Validation-Testing: Proximal Policy Optimization (PPO) algorithm [8] from Stable-Baselines3 [7] is employed to train the agent. The validation set is used to check correctness and tune hyperparameters. Performance is reported on the testing set.

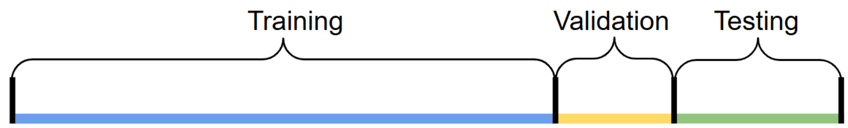

Figure 1: Illustration of data splitting.

Metric	FinRL-Meta [6]	Gym-anytrading [1]	DJIA	Mean-Variance	Equal-Weight
Cumul. return	54.6%	41.1%	28.5%	54.8%	30.0%
Annual return	37.5%	30.6%	20.0%	37.6%	21.1%
Annual volatility	18.8%	19.0%	13.7%	11.4%	11.2%
Sharpe ratio	1.26	1.06	1.16	1.68	1.24
Max drawdown	-16.5%	-16.5%	-9.0%	-9.9%	-8.7%

Table 2: Performance for stock trading task.

Comparison: To assess the performance of the agent in each environment, test results are compared against three baselines: Dow Jones Industrial Average (DJIA) index, equal-weight strategy, and the mean-variance portfolio allocation strategy. They are compared on the selected performance metrics, including cumulative return, annualized return, annualized volatility, Sharpe ratio, and maximum drawdown.

3.2 Pipeline for Crypto Market Environments

Crypto Trading Tasks: We evaluate the crypto market environments on algorithmic trading. Transaction cost for both buy and sell are set to 0.1%.

Data Preprocessing: We load five-minute level OHLCV data of Bitcoin (BTC) from Binance. A standardized preprocessing pipeline is then applied. It fills missing values, adds technical indicators (MACD and RSI), and incorporates turbulence index into the dataset. It is also split into training, validation, and testing sets, as illustrated by Fig. 1. We use data from 07/01/2024 00:00 to 07/14/2024 23:55 for training, data from 07/15/2024 00:00 to 07/15/2024 23:55 for validation, and data from 07/16/2024 00:00 to 07/19/2024 23:55 for testing.

Training-Validation-Testing: PPO algorithm [8] from Stable-Baselines3 [7] is used to train the agent. The validation set is used to check correctness and tune hyperparameters. Performance is reported on testing set.

Comparison: Trading results are compared against three baselines: S&P Cryptocurrency Broad Digital Asset (S&P BDA) Index, equal-weight strategy, and the mean-variance portfolio allocation strategy. They are compared on the selected performance metrics, including cumulative return, annualized return, annualized volatility, Sharpe ratio, and maximum drawdown.

3.3 Reproduction Validation

Each environment is evaluated within the standard pipeline to reproduce its results. Correctness is validated by comparing with the original implementations on the same dataset.

3.4 Environment Standardization

To enhance extensibility, usability, and reproducibility, market environments are standardized following the Gymnasium-Style [12] standard. Method names, parameter designs, and return formats are unified. This allows easier integration of new environments and more consistent reproductions.

4 Experimental Results

4.1 Stock Trading Task

We evaluate stock trading environments from FinRL-Meta [6] and Gym-anytrading [1] in our pipeline. Our reproduced results follow the same trend direction, and outperform the original result in cumulative return, as shown in Fig. 2. This consistency shows our reproduction is correct for both environments. FinRL-Meta outperforms two baselines (DJIA, Equal-Weight) in both annual return and Sharpe ratio, and matches the performance of the Mean-Variance strategy, achieving an annual return of 37.5% and a Sharpe ratio of 1.26. Gym-anytrading outperforms DJIA and Equal-Weight in terms of annual return, reaching 30.6%, though its Sharpe ratio of 1.06 indicates relatively higher risk, as shown in Table 2.

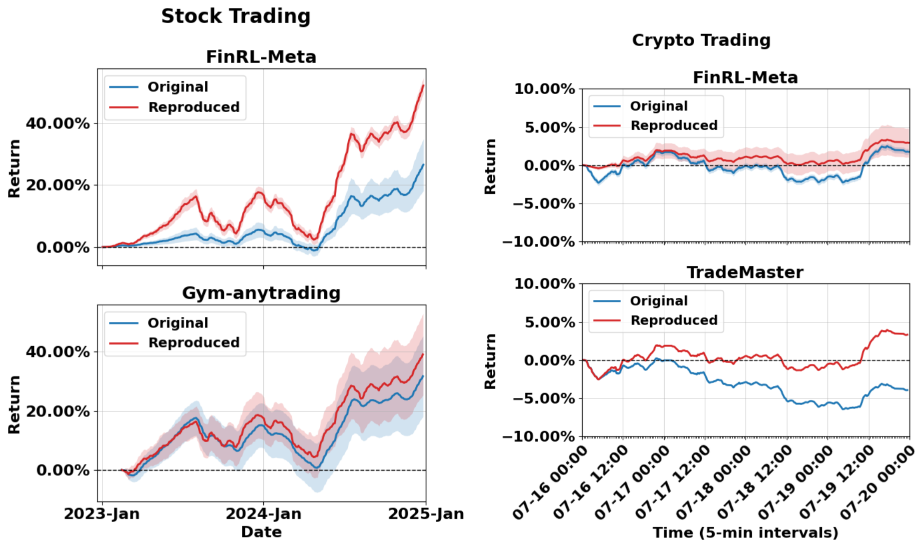

Figure 2: Comparison of original and reproduced results for stock trading and crypto trading tasks.

4.2 Crypto Trading Task

We select crypto trading environments from FinRL-Meta [6] and TradeMaster [9]to be evaluated with our pipeline. Our reproduced results follow the same trend direction, and slightly outperform the original result in cumulative return, as shown in Fig. 2. This shows our reproduced results are consistent with the original results.

FinRL-Meta outperforms the three baselines (S&P BDA Index, Mean-Variance, and Equal-Weight) in both annual return and Sharpe ratio, achieving 940.3% annual return with a Sharpe ratio of 26.87. TradeMaster also outperforms three baselines (S&P BDA Index, Mean-Variance, and Equal-Weight), reaching an annual return of 1976.3% and a Sharpe ratio of 35.91, as shown in Table 3. Due to the short evaluation window, the Sharpe ratios lack practical financial meaning and are provided for reference only.

Metric	FinRL-Meta [6]	TradeMaster [9]	S&P BDA Index	Mean-Variance	Equal-Weight
Cumul. return	2.6%	3.3%	2.5%	0.1%	0.9%
Annual return	940.3%	1917.0%	851.8%	10.4%	127.0%
Annual volatility	35.0%	53.4%	71.6%	101.3%	57.5%
Sharpe ratio	26.87	35.91	11.90	0.10	2.21
Max drawdown	−0.7%	−0.9%	−1.5%	−6.2%	−2.2%

Table 3: Performance for cryptocurrency trading task.

5 Conclusions

In this work, we address the lack of organization in open-source market environments for financial reinforcement learning. We propose two standardized evaluation pipelines—one for stock trading and one for crypto trading. The pipelines integrate financial data, baseline strategies, and five performance metrics. They enable fair and reproducible evaluation across environments. We evaluate two stock market environments and two crypto market environments. Experimental results show that our reproduction is correct. We further standardized the open-source environments to Gymnasium-Style interface[12]. This ensures consistent reproduction. Overall, this work establishes a standardized foundation that facilitates evaluation and promotes future research in financial reinforcement learning.

References

[1] AminHP. gym-anytrading: A collection of openai gym environments for reinforcement learning-based trading, 2025. URL `https://github.com/AminHP/gym-anytrading`. GitHub repository, MIT license. Accessed 2025-08-20.

[2] ClementPerroud. Gym-trading-env: A simple, easy, customizable gymnasium environment for trading. GitHub repository, 2025. URL `https://github.com/ClementPerroud/Gym-Trading-Env`. Accessed: 2025-08-20.

[3] Ben Hambly, Renyuan Xu, and Huining Yang. Recent advances in reinforcement learning in finance. *Mathematical Finance*, 33(3):437–503, 2023.

[4] kh-kim. stock_market_reinforcement_learning: Stock trading environment using openai gym with deep q-learning and policy gradient. GitHub repository, 2025. URL `https://github.com/kh-kim/stock_market_reinforcement_learning`. Accessed: 2025-08-20.

[5] Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang. FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance. *Deep RL Workshop, NeurIPS 2020*, 2020.

[6] Xiao-Yang Liu, Ziyi Xia, Jingyang Rui, Jiechao Gao, Hongyang Yang, Ming Zhu, Christina Wang, Zhaoran Wang, and Jian Guo. FinRL-Meta: Market environments and benchmarks for data-driven financial reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 1835–1849, 2022.

[7] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL `http://jmlr.org/papers/v22/20-1364.html`.

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

[9] Shuo Sun, Molei Qin, Wentao Zhang, Haochong Xia, Chuqiao Zong, Jie Ying, Yonggang Xie, Lingxuan Zhao, Xinrun Wang, and Bo An. TradeMaster: A holistic quantitative trading platform empowered by reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 59047–59061, 2023.

[10] Shuo Sun, Rundong Wang, and Bo An. Reinforcement learning for quantitative trading. *ACM Transactions on Intelligent Systems and Technology*, 14(3):1–29, 2023.

[11] tensortrade-org. Tensortrade: An open-source reinforcement learning framework for trading. GitHub repository, 202. URL `https://github.com/tensortrade-org/tensortrade/tree/master`. Accessed: 2025-08-20.

[12] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. *arxiv:2047.17032*, 2024.

[13] Keyi Wang, Nikolaus Holzer, Ziyi Xia, Yupeng Cao, Jiechao Gao, Anwar Walid, Kairong Xiao, and Xiao-Yang Liu Yanglet. FinRL Contests: Benchmarking data-driven financial reinforcement learning agents. *Wiley Artificial Intelligence for Engineering*, 2025.

[14] Xiao Yang, Weiqing Liu, Dong Zhou, Jiang Bian, and Tie-Yan Liu. Qlib: An ai-oriented quantitative investment platform, 2020. URL `https://arxiv.org/abs/2009.11189`.

[15] Yvictor. Tradinggym: A reinforcement learning environment for trading and backtesting, 2025. URL `https://github.com/Yvictor/TradingGym`. GitHub repository, MIT license. Accessed 2025-10-26.