FinRL-Meta: Market Environments and Benchmarks for Data-Driven Financial Reinforcement Learning

Xiao-Yang Liu¹, Ziyi Xia¹, Jingyang Rui², Jiechao Gao³, Hongyang Yang¹, Ming Zhu⁴, Christina Dan Wang⁵, Zhaoran Wang⁶, Jian Guo⁷
¹Columbia University, ²The University of Hong Kong, ³University of Virginia, ⁴SIAT-CAS, ⁵New York University (Shanghai), ⁶Northwestern University, ⁷IDEA Research

Abstract

Finance is a particularly difficult playground for deep reinforcement learning. Establishing high-quality market environments and benchmarks on financial reinforcement learning is challenging due to three major factors, namely, low signal-to-noise ratio of financial data, survivorship bias of historical data, and information leakage in the backtesting stage. In this paper, we present an openly accessible FinRL-Meta library that has been actively maintained by the FinRL community. First, following a DataOps paradigm, we provide hundreds of market environments through an automatic pipeline that collects dynamic datasets from real-world markets and processes them into standard gym-style market environments. Second, we benchmark popular papers as stepping stones for users to design new trading strategies. We also hold our benchmarks on cloud platforms so that users can visualize their own results and assess the relative performance via community-wise competitions. Third, FinRL-Meta provides tens of Jupyter/Python demos organized in a curriculum and a documentation website to serve the rapidly growing community. FinRL-Meta is available at: https://github.com/AI4Finance-Foundation/FinRL-Meta

1 Introduction

Finance is a particularly difficult playground for deep reinforcement learning (DRL) [41], including developing profitable strategies [53, 51, 53], assessing portfolio risk [53? ], and studying market fragility [40]. However, establishing high-quality market environments and benchmarks on financial reinforcement learning are challenging due to three major factors, namely, low signal-to-noise ratio (SNR) of financial data, survivorship bias of historical data, and information leakage in the backtesting stage. Such a simulation-to-reality gap [12, 13] degrades the performance of DRL strategies in real markets. Therefore, high-quality market environments and DRL benchmarks are crucial for the research and industrialization of data-driven financial reinforcement learning.

Existing works have applied various DRL methods in financial applications [33, 52, 27, 38]. Many of them have shown better trading performance in terms of cumulative return and Sharpe ratio. Several recent works [53, 2, 42] showed the great potential of DRL-based market simulators that are not publicly available yet. Therefore, these works are difficult to reproduce. The FinRL library [31, 52] provided an open-source framework for financial reinforcement learning. However, it focused on guaranteeing reproducibility of backtesting performance while several market environments were provided. Our workshop version of FinRL-Meta [29] provided data processors to access and clean unstructured market data. But we did not provide benchmarks back then.

The DataOps paradigm [39, 5, 14] refers to a set of practices, processes, and technologies that combines automated data engineering and agile development [14]. It helps reduce the cycle time of data engineering and improve data quality. To deal with financial big data (usually unstructured), we fol-
The DataOps paradigm and implement an automatic pipeline in Fig. 1 (left): task planning, data processing, training-testing-trading, and monitoring agents’ performance. Through this pipeline, we continuously produce DRL benchmarks on dynamic market datasets.

In this paper, we present an openly accessible FinRL-Meta library that has been actively maintained by the FinRL community. We aim to create an infrastructure to enable near real-time paper trading and facilitate the real-world adoption of financial reinforcement learning. This is relevant to the broader RL research community since it provides a rare case of a task that can be tested against real-world performance without major investment, while robotics requires simulation or expensive equipment and games are available in simulations.

Fig. 1 (right) shows an overview of data-driven financial reinforcement learning. First, following a DataOps paradigm, we provide hundreds of market environments through an automatic pipeline that collects dynamic datasets from real-world markets and processes them into standard gym-style market environments. Second, we reproduce popular papers as benchmarks, including high-frequency stock trading, cryptocurrency trading, and stock portfolio allocation, serving as stepping stones for users to design new strategies. With the help of the data engineering pipeline, we hold our benchmarks on cloud platforms so that users can visualize their own results and assess the relative performance via community-wise competitions. Third, FinRL-Meta provides tens of Jupyter/Python demos as educational materials, organized in a curriculum, and a documentation website to serve the rapidly growing community.

The remainder of this paper is organized as follows. Section 2 reviews existing works. Section 3 describes challenges and presents an overview of our FinRL-Meta framework. Section 4 describes how we process data into market environments. In Section 5, we benchmark popular DRL papers. Finally, we conclude this paper in Section 6.

2 Related Works

We review DataOps practices and existing data-driven reinforcement learning and benchmarks.

DataOps practices: DataOps applies the ideas of lean development and DevOps to the data analytics field. DataOps practices have been developed in companies and organizations to improve the quality of and efficiency of data analytics. These implementations consolidate various data sources, unify and automate the pipeline of data analytics, including data accessing, cleaning, analysis, and visualization.

However, the DataOps methodology has not been applied to financial reinforcement learning researches. Most researchers access data, clean data, and extract technical indicators (features) in a case-by-case manner, which involves heavy manual work and may not guarantee the data quality.

Data-driven reinforcement learning: Environments are crucial for training DRL agents.

• OpenAI gym provides standardized environments for a collection of benchmark problems that expose a common interface, which is widely supported by many libraries. Three trading environments, TradingEnv, ForexEnv, and StocksEnv, are included to support Stock and FOREX markets. However, it has not been updated for years.

• D4RL introduces the idea of Datasets for deep data-driven reinforcement learning (D4RL). It provides benchmarks in offline RL. However, D4RL does not provide financial environments.
Table 1: List of state space, action space, and reward function.

<table>
<thead>
<tr>
<th>Key components</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>State</strong></td>
<td>Balance $b_t \in \mathbb{R}^+$; Shares $h_t \in \mathbb{Z}^n_+$; Opening/high/low/close prices $o_t, h_t, l_t, p_t \in \mathbb{R}^n_+$; Trading volume $v_t \in \mathbb{R}^n_+$; Fundamental indicators; Technical indicators; Social data; Sentiment data; Smart beta indexes, etc.</td>
</tr>
<tr>
<td><strong>Action</strong></td>
<td>Buy/Sell/Hold; Short/Long; Portfolio weights</td>
</tr>
<tr>
<td><strong>Reward</strong></td>
<td>Change of portfolio value; Portfolio log-return; Sharpe ratio</td>
</tr>
<tr>
<td><strong>Environments</strong></td>
<td>Dow-30, S&amp;P-500, NASDAQ-100; Cryptocurrencies; Foreign currency and exchange; Futures; Options; ETFs; Forex; CN securities; US securities; NMS US securities; Paper trading; Living trading</td>
</tr>
</tbody>
</table>

- **FinRL** [31, 32] is an open-source library for financial reinforcement learning that builds a full pipeline for financial reinforcement learning. It contains three market environments, i.e., stock trading, portfolio allocation, crypto trading, and two data sources, i.e., Yahoo Finance and WRDS. However, those market environments of FinRL cannot meet the community’s growing demands.

- **NeoRL** [39] collected offline RL environments for four areas, CityLearn [50], FinRL [31, 32], Industrial Benchmark [21], and MuJoCo [48], where each area contains several gym-style environments. Regarding financial aspects, it directly borrows market environments from FinRL.

**Benchmarks of financial reinforcement learning:** Many researches applied DRL in quantitative finance [30, 53, 54] by building their own market environments. Despite the above-mentioned open-source libraries that provide some useful environments, there are no established benchmarks yet. On the other hand, the data accessing, cleaning and factor extraction processes are usually limited to data sources like Yahoo Finance and Wharton Research Data Services (WRDS).

3 Financial Reinforcement Learning and FinRL-Meta Framework

We describe financial reinforcement learning and its challenges, then provide an overview of our FinRL-Meta framework.

3.1 Financial Reinforcement Learning and Challenges

**Assuming full observability,** we model a trading task as a Markov Decision Process (MDP) with five tuples $\langle S, A, P, r, \gamma \rangle$ (where $S$ and $A$ denote the state space and action space, respectively, $\mathbb{P}(s'|s, a)$ is the transition probability of an unknown environment, $r(s, a, s')$ is a reward function, and $\gamma \in (0, 1)$ is a discount factor. A trading agent learns a policy $\pi(s_t|a_t)$ that maximizes the discounted cumulative return $R = \sum_{t=0}^{T} \gamma^t r(s_t, \alpha_t, s_{t+1})$ over a trading period $t = 0, 1, ..., T$.

The historical dataset before time $0$ is used to train the trading agent. Note that we process the dataset into a market environment, following the *de facto* standard of OpenAI gym [7]. In Table I we list the state space, action space, and reward function.

- **State** $s \in S$: A state represents an agent’s perception of a market environment, which may include balance, shares, OHLCV values, technical indicators, social data, sentiment data, etc.
Figure 2: Overview of FinRL-Meta framework.

- **Action** $a \in \mathcal{A}$: An action is taken from the allowed action set at a state. Actions may vary for different trading tasks, e.g., for stock trading, the actions are the number of shares to buy/sell for each stock, while for portfolio allocation, the actions are the allocation weights of the capital.

- **Reward** $r(s, a, s')$: an incentive mechanism for an agent to learn a better policy. Several common reward functions are provided: 1). Change of portfolio value $r(s, a, s') = v' - v$, where $v'$ and $v$ are portfolio values at state $s'$ and $s$, respectively; 2). Portfolio log return $r(s, a, s') = \log(v'/v)$; and 3). Sharpe ratio defined in Section 5.1.

The above full observability assumption can be extended to partial observation (the underlying states cannot be directly observed), i.e., partially observable Markov Decision Process (POMDP). A POMDP model utilizes a Hidden Markov Model (HMM) to model a time series that is caused by a sequence of unobservable states. Considering the noisy financial data, it is natural to assume that a trading agent cannot directly observe market states. Studies suggested that the POMDP model can be solved by using recurrent neural networks, e.g., an off-policy Recurrent Deterministic Policy Gradient (RDPG) algorithm, and a long short-term memory (LSTM) network that encodes partial observations into a state of a reinforcement learning.

Training and testing environments based on historical data may not simulate real markets accurately due to the simulation-to-reality gap, and thus a trained agent cannot be directly deployed in real-world markets. We summarize three major factors for the simulation-to-reality gap in financial reinforcement learning as follows:

- **Low signal-to-noise ratio (SNR) of financial data**: Data from different sources may contain large noise such as random noise, outliers, etc. It is challenging to identify alpha signals or build smart beta indexes using noisy datasets.

- **Survivorship bias of historical market data**: Survivorship bias is caused by a tendency to focusing on existing stocks and funds without consideration of those that are delisted. It could lead to an overestimation of stocks and funds, which will mislead the agent.

- **Information leakage in backtesting stage**: Existing research mainly report backtesting results. It is possible to tune hyper-parameters and retrain the agent multiple times to obtain better backtesting results, causing information leakage.

### 3.2 Overview of FinRL-Meta

FinRL-Meta builds a universe of market environments for data-driven financial reinforcement learning. FinRL-Meta follows the de facto standard of OpenAI Gym and the lean principle of software development. It has the following unique features.

**Layer structure and extensibility**: As shown in Fig. 2, we adopt a layered structure that consists of three layers, data layer, environment layer, and agent layer. Layers interact through end-to-end interfaces, achieving high extensibility. For updates and substitutes inside a layer, this structure minimizes the impact on the whole system. Moreover, the layer structure allows easy extension of user-defined functions and fast updating of algorithms with high performance.
**Training-testing-trading pipeline**: We employ a training-testing-trading pipeline that the DRL approach follows a standard end-to-end pipeline. The DRL agent is first trained in a training environment and then fine-tuned (adjusting hyperparameters) in a validation environment. Then the validated agent is tested on historical datasets (backtesting). Finally, the tested agent will be deployed in paper trading or live trading markets.

**Plug-and-play**: In the above training-testing-trading pipeline, a DRL agent can be directly plugged in, then trained and tested. The following DRL libraries are supported:

- **ElegantRL [27]**: Lightweight, efficient and stable DRL implementation using PyTorch.
- **Stable-Baselines3 [31]**: Improved DRL algorithms based on OpenAI Baselines.
- **RLlib [25]**: An open-source DRL library that offers high scalability and unified APIs.

## 4 Financial Big Data and DataOps for Dynamic Datasets

Financial big data is usually unstructured in shape and form. We process four types of data [11] into market environments, including fundamental data (e.g., earning reports), market data (e.g., OHLCV data), analytics (e.g., news sentiment), and alternative data (e.g., social media data, ESG data).

### 4.1 Data Layer for Unstructured Financial Big Data

**DataOps pipeline for data-driven financial reinforcement learning**: We follow the DataOps paradigm [13], 5, 39] in the data layer. As shown in Fig. 3, we establish a standard pipeline for financial data engineering, which processes data from different sources into a unified market environment following the de facto standard of OpenAI gym [11]. We automate this pipeline with a data processor with the following functionalities.

- **Data accessing**: Users can connect data APIs of different platforms via our common interface. Users can access data agilely by specifying the start date, end date, stock list, time interval, and other parameters. FinRL-Meta has supported more than 20 data sources, covering stocks, cryptocurrencies, ETFs, forex, etc.
• **Data cleaning**: Raw data retrieved from different data sources are usually of various formats and with erroneous or missing data to different extents. It makes data cleaning highly time-consuming. With a data processor, we automate the data cleaning process. In addition, we use stock ticker name and data frequency as unique identifiers to merge all types of data into one unified data table.

• **Feature engineering**: In feature engineering, FinRL-Meta aggregates effective features which can help improve model predictive performance. We provide various types of features, including but not limited to fundamental, market, analytics, and alternative features. Users can quickly add features using open-source libraries or add user-defined features. Users can add new features in two ways: 1) Write a user-defined feature extraction function directly. The returned features are added to a feature array. 2) Store the features in a file, and put it in a default folder. Then, an agent can read these features from the file.

**Automated feature engineering**: FinRL-Meta currently supports four types of features in our automated pipeline.

• **Fundamental features**: Fundamental features are processed based on the earnings data in SEC filings queried from WRDS. The data frequency is low, typical quarterly, e.g., four data points in a year. To avoid information leakage, we use a two-month lag beyond the standard quarter end date, e.g., Apple released its earnings report on 2022/07/28 for the third quarter (2022/06/25) of year 2022. Thus for the quarter between 04/01 and 06/30, our trade date is adjusted to 09/01 (same method for other three quarters). We also provide functions in our data processor for calculating financial ratios based on earnings data such as earnings per share (EPS), return on asset (ROA), price to earnings (P/E) ratio, net profit margin, quick ratio, etc.

• **Market features**: Open-high-low-close price and volume data are the typical market data we can directly get from querying the data API. They have various data frequencies, such as daily prices from YahooFinance, TAQ (Millisecond Trade and Quote) from WRDS. In addition, we automate the calculation of technical indicators based on OHLCV data by connecting the Stockstats\(^1\) or TA-lib library\(^2\) in our data processor, such as Moving Average Convergence Divergence (MACD), Average Directional Index (ADX), Commodity Channel Index (CCI), etc.

• **Analytics features**: We provide news sentiment for analytics features. First, we get the news headline and content from WRDS. Next, we use NLTK.Vader\(^3\) to calculate sentiment based on the sentiment compound score of a span of text by normalizing the emotion intensity (positive, negative, neutral) of each word. For the time alignment with market data, we use the exact enter time, i.e., when the news enters the database and becomes available, to match the trade time. For example, if the trade time is every ten minutes, we collect the previous ten minutes’ news based on the enter time; if no news is detected, then we fill the sentiment with 0.

• **Alternative features**: Alternative features are useful, but hard-to-obtain from different data sources \[^11\], such as ESG data, social media data, Google trend searches, etc. ESG (Environmental, social, governance) data are widely used to measure the sustainability and societal impacts of an investment. The ESG data we provide is from the Microsoft Academic Graph database, which is an open resource database with records of scholar publications. We have functions in our data processor to extract AI publication and patent data, such as paper citations, publication counts, patent counts, etc. We believe these features reflect companies’ research and development capacity for AI technologies \[^15\],\[^9\]. It is a good reflection of ESG research commitment.

4.2 Environment Layer for Creating Dynamic Market Environments

FinRL-Meta follows the OpenAI gym-style \[^7\] to create market environments using the cleaned data from the data layer. It provides hundreds of environments with a common interface. Users can build their environments using FinRL-Meta’s interfaces, share their results and compare a strategy’s trading performance.

\(^1\) Github repo: [https://github.com/jealous/stockstats](https://github.com/jealous/stockstats)
\(^2\) Github repo: [https://github.com/mrjbq7/ta-lib](https://github.com/mrjbq7/ta-lib)
\(^3\) Github repo: [https://github.com/nltk/nltk](https://github.com/nltk/nltk)
We plan to add more environments for users’ convenience. For example, we are actively building market simulators using limit-order-book data\(^4\), where we simulate the market from the playback of historical limit-order-book-level data and an order matching mechanism. We foresee the flexibility and potential of using a Hidden Markov Model (HMM) \(^[37]\) or a generative adversarial net (GAN) \(^[17]\) to generate market scenarios \(^[11]\).

**Incorporating trading constraints to model market frictions:** To better simulate real-world markets, we incorporate common market frictions (e.g., transaction costs and investor risk aversion) and portfolio restrictions (e.g., non-negative balance).

- **Flexible account settings:** Users can choose whether to allow buying on margin or short-selling.
- **Transaction cost:** We incorporate the transaction cost to reflect market friction, e.g., 0.1% of each buy or sell trade.
- **Risk-control for market crash:** In FinRL \(^[31, 32]\), a turbulence index \(^[23]\) is used to control risk during market crash situations. However, calculating the turbulence index is time-consuming. It may take minutes, which is not suitable for paper trading and live trading. We replace the financial turbulence index with the volatility index (VIX) \(^[51]\) that can be accessed immediately.

**Multiprocessing training via vector environment:** We utilize GPUs for multiprocessing training, namely, the vector environment technique of Isaac Gym \(^[35]\), 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 later used to update a learner. By adopting this technique, we successfully achieve the multiprocessing simulation of hundreds of market environments to improve the performance of DRL trading agents on large datasets.

### 4.3 Advantages

Our DataOps pipeline is automatic, which gives us the following three advantages.

**Curriculum for newcomers:** We provide an educational curriculum, as shown in Fig. \(^[9]\), for community newcomers with different levels of proficiency and learning goals. Users can grow programming skills by gradually changing the data/environment layer following instructions on our website.

**Benchmarks on cloud:** We provide demos on a cloud platform, Weights & Biases \(^[5]\), to demonstrate the training process. We define the hyperparameter sweep, training function, and initialize an agent to train and tune hyperparameters. On the cloud platform Weights & Biases, users are able to visualize their results and assess the relative performance via community-wise competitions.

**Curriculum learning for agents:** Based on FinRL-meta (a universe of market environments, say \(\geq 100\)), one is able to construct an environment by sampling data samples from multiple market datasets, similar to XLand \(^[47]\). In this way, one can apply the curriculum learning method \(^[47]\) to train a generally capable agent for several financial tasks.

### 5 Tutorials and Benchmarks of Financial Reinforcement Learning

We provide tens of tutorial notebooks to serve as stepping stones for newcomers and reproduce popular papers as benchmarks for follow-up research.

#### 5.1 Metrics and Baselines for Evaluating Performance

We provide the following metrics to measure the trading performance:

- **Cumulative return** \(R = \frac{V - V_0}{V_0}\), where \(V\) is the final portfolio value, and \(V_0\) is the original capital.
- **Annualized return** \(r = (1 + R)^{\frac{t}{365}} - 1\), where \(t\) is the number of trading days.

\(^4\)Github repo: [https://github.com/AI4Finance-Foundation/Market_Simulator](https://github.com/AI4Finance-Foundation/Market_Simulator)

\(^5\)Website: [https://wandb.ai/site](https://wandb.ai/site)
Figure 4: Demos of FinRL-Meta, organized in a curriculum structure.

- **Annualized volatility** \( \sigma_a = \sqrt{\frac{\sum_{i=1}^{n} (r_i - \bar{r})^2}{n-1}} \), where \( r_i \) is the annualized return in year \( i \), \( \bar{r} \) is the average annualized return, and \( n \) is the number of years.
- **Sharpe ratio** \[\begin{align*} S_T &= \frac{\text{mean}(R_t) - r_f}{\text{std}(R_t)} \end{align*}\], where \( R_t = \frac{v_t - v_{t-1}}{v_{t-1}} \), \( r_f \) is the risk-free rate, and \( t = 1, ..., T \).
- **Max. drawdown**: The maximal percentage loss in portfolio value.

The following baseline trading strategies are provided for comparisons:

- **Passive trading strategy** [36], a well-known long-term strategy. The investors just buy and hold selected stocks or indexes without further activities.
- **Mean-variance and min-variance strategy** [3], both strategies look for a balance between risks and profits. It selects a diversified portfolio to achieve higher profits at lower risk.
- **Equally weighted strategy**, a portfolio allocation strategy that gives equal weights to different assets, avoiding allocating overly high weights on particular stocks.

5.2 Tutorials and Demos in Jupyter Notebooks

For educational purposes, we provide Jupyter notebooks as tutorials to help newcomers get familiar with the whole pipeline.

- **Stock trading** [30]: We apply popular DRL algorithms to trade multiple stocks.
- **Portfolio allocation** [31]: We use DRL agents to optimize asset allocation in a set of stocks.
- **Cryptocurrency trading** [31]: We reproduce the experiment [31] on 10 popular cryptocurrencies.
- **Multi-agent RL for liquidation strategy analysis** [6]: We reproduce the experiment in [6]. The multi-agent optimizes the shortfalls in the liquidation task, which is to sell given shares of one stock sequentially within a given period, considering the costs arising from the market impact and the risk aversion.
- **Ensemble strategy for stock trading** [53]: We reproduce the experiment in [53] that employed an ensemble strategy of several DRL algorithms on the stock trading task.
- **Paper trading demo**: We provide a demo for paper trading. Users could combine their own strategies or trained agents in paper trading.
5.3 Reproducing Prior Papers as Benchmarks

We have reproduced experiments in several papers as benchmarks. Users can study our codes for research purpose or use them as stepping stones for deploying trading strategies in live markets.

Stock trading task [30]: We access Yahoo! Finance database and select the 30 constituent stocks (accessed at 07/01/2020) in Dow Jones Industrial Average (DJIA). We use data from 01/01/2009 to 06/30/2020 for training and data from 07/01/2020 to 05/31/2022 for testing. We use technical indicators in our state space, e.g., Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), Commodity Channel Index (CCI), Average Directional Index (ADX), etc. As shown in Fig. 5 (left), we trained five popular DRL algorithms to trade and compare their results with the DJIA index. We show a detailed walkthrough of how DRL works in the stock trading task, on which many subsequent works are based [30]. This benchmark is beneficial for getting into the field of RL in finance.

Liquidation analysis [6]: We build a simulated environment of stock prices according to the Almgren and Chriss model. Then we implement the multi-agent DRL algorithms for both competing and cooperative liquidation strategies. This benchmark demonstrates the trade execution task using deep reinforcement learning algorithms. When trading, traders want to minimize the expected trading cost, which is also called implementation shortfall. In Fig. 5 (right), there are two agents, and we see that the implementation shortfalls decrease during the training process.

Explainable financial reinforcement learning [13]: We reproduce [13] that compares the performance of DRL algorithms with machine learning (ML) methods on the multi-step prediction in the portfolio allocation task. We use four technical indicators MACD, RSI, CCI, and ADX as features. Random Forest (RF), Decision Tree Regression (DT), Linear Regression (LR), and Support Vector Machine (SVM) are the ML algorithms in comparison. We use data from Dow Jones 30 constituent stocks to construct the environment. We use data from 04/01/2009 to 03/31/2020 as the training set and data from 04/01/2020 to 05/31/2022 for backtesting. In Fig. 6, the results show that DRL methods have higher Sharpe ratio than ML methods. Also, DRL methods’ average correlation coefficient are significantly higher than that of ML methods (multi-step).

Podracer on the cloud [24, 28]: We reproduce cloud solutions of population-based training, e.g., generational evolution [24] and tournament-based evolution [28]. If GPUs are abundant, users can take advantage of this benchmark to meet the real-time requirement of high-frequency trading tasks. Detailed instructions are provided on our website.

Ensemble strategy [53]: The ensemble method combines different agents to obtain an adaptive one, which inherits the best features of the agents and performs remarkably well in practice. We consider three component algorithms, Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), and Deep Deterministic Policy Gradient (DDPG), which have different strengths and weaknesses. For instance, A2C is good at dealing with a bearish trend market. PPO is good at following trends...
and acts well in generating more returns in a bullish market. DDPG can be used as a complementary strategy to PPO in a bullish trend. Using a rolling window, an ensemble agent automatically selects the best model for each test period. Again on the 30 constituent stocks of the DJIA index, we use data from 04/01/2009 to 06/30/2019 for training, and data from 07/01/2020 to 03/31/2022 for validation and testing through a quarterly rolling window.

From Fig. 6 and Table 2, we observe that the ensemble agent outperforms other agents. In the experiment, the ensemble agent has the highest Sharpe ratio of 1.53, which means it performs the best in balancing risks and profits. This benchmark demonstrates that the ensemble strategy is effective in constructing a more reliable agent based on several component DRL agents.

6 Conclusion

In this paper, we followed the DataOps paradigm and developed a FinRL-Meta library that provides openly accessible dynamic financial datasets and reproducible benchmarks. For future work, FinRL-Meta aims to build a universe of financial market environments, like the XLand environment. To improve the performance for the large-scale markets, we are exploiting GPU-based massive parallel simulation such as Isaac Gym. Moreover, it will be interesting to explore the evolutionary perspectives to simulate the markets. We believe that FinRL-Meta will provide insights into complex market phenomena and offer guidance for financial regulations.
<table>
<thead>
<tr>
<th>(2020/07/01-2022/03/31)</th>
<th>Ensemble [53]</th>
<th>A2C</th>
<th>PPO</th>
<th>DDPG</th>
<th>DJIA index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annual Return</td>
<td>25.9%</td>
<td>23.3%</td>
<td>13.1%</td>
<td>12.7%</td>
<td>19.7%</td>
</tr>
<tr>
<td>Annual Volatility</td>
<td>15.9%</td>
<td>16.2%</td>
<td>13.4%</td>
<td>15.0%</td>
<td>14.4%</td>
</tr>
<tr>
<td>Sharpe Ratio</td>
<td>1.53</td>
<td>1.37</td>
<td>0.99</td>
<td>0.88</td>
<td>1.32</td>
</tr>
<tr>
<td>Sortino Ratio</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Calmar Ratio</td>
<td>2.27</td>
<td>1.97</td>
<td>0.88</td>
<td>0.85</td>
<td>1.74</td>
</tr>
<tr>
<td>Max Drawdown</td>
<td>-11.4%</td>
<td>-11.8%</td>
<td>-14.9%</td>
<td>-14.9%</td>
<td>-11.3%</td>
</tr>
</tbody>
</table>

Table 2: Reproducing trading performance in paper [53].

References


Table 3: List of key terms with corresponding description.

<table>
<thead>
<tr>
<th>Key Terms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>A decision maker</td>
</tr>
<tr>
<td>Environment</td>
<td>A world with which an agent interacts with</td>
</tr>
<tr>
<td>Markov Decision Process (MDP)</td>
<td>A mathematical framework to model decision-making problems</td>
</tr>
<tr>
<td>State, Action, Reward</td>
<td>Three main factors in agent-environment interaction</td>
</tr>
<tr>
<td>Policy</td>
<td>A rule that agent follow to make decision</td>
</tr>
<tr>
<td>Policy gradient</td>
<td>An approach to solve RL problems by optimizing the policy directly</td>
</tr>
<tr>
<td>Hyperparameter tuning</td>
<td>Change hyperparameter during training to get a converged result faster</td>
</tr>
<tr>
<td>Ensemble strategy</td>
<td>A ML technique. Here we combine several DRL agent to a better model</td>
</tr>
<tr>
<td>Population-based training (PBT)</td>
<td>Optimise a population of models and hyperparameters, and select the optimal set</td>
</tr>
<tr>
<td>Generational evolution [24]</td>
<td>Employing an evolution strategy over generations</td>
</tr>
<tr>
<td>Tournament-based evolution [28]</td>
<td>An evolution by asynchronously updating a tournament board of models</td>
</tr>
<tr>
<td>Curriculum learning</td>
<td>Train an ML model from easier to harder data, imitating the order in human curricula</td>
</tr>
<tr>
<td>Simulation-to-reality gap</td>
<td>An unavoidable gap due to the difference of simulation environment and reality</td>
</tr>
<tr>
<td>Signal-to-noise ratio (SNR)</td>
<td>A ratio of desired signal (good data) to undesired signal (noise)</td>
</tr>
<tr>
<td>DataOps</td>
<td>A series of principles and practices to improve the quality of data science.</td>
</tr>
<tr>
<td>Sentiment data</td>
<td>A category in financial big data that contains subjective viewpoints.</td>
</tr>
<tr>
<td>Historical data</td>
<td>All kinds of data that already existed in the past</td>
</tr>
<tr>
<td>Backtesting</td>
<td>A method to see how a strategy performs on a certain period of historical data.</td>
</tr>
<tr>
<td>Surviviorship bias</td>
<td>A bias caused by only seeing existing examples, but not those already died out.</td>
</tr>
<tr>
<td>Information leakage</td>
<td>When the data contains future information, causing model overfitting</td>
</tr>
<tr>
<td>Paper trading</td>
<td>Simulation of buying and selling without using real money</td>
</tr>
<tr>
<td>OHLCV</td>
<td>A popular form of market data with: Open, High, Low, Close, Volume</td>
</tr>
<tr>
<td>Technical indicators</td>
<td>A statistical calculation based on OHLCV data to indicate future price trends</td>
</tr>
<tr>
<td>Smart beta index</td>
<td>An enhanced indexing strategy to beat a benchmark index</td>
</tr>
<tr>
<td>Market frictions</td>
<td>A financial market friction as anything that interferes with trade.</td>
</tr>
<tr>
<td>Market crash</td>
<td>A huge drop of market price within a very short time</td>
</tr>
<tr>
<td>Volatility index (VIX) [51]</td>
<td>A market index that shows the market’s expectations for volatility</td>
</tr>
<tr>
<td>Liquidation, trade execution</td>
<td>An investor closes their position in an asset</td>
</tr>
</tbody>
</table>

A  Terminology of Reinforcement Learning, Data Science, and Finance

We provide a list of key terms and corresponding description in Table 3. For the terminology of reinforcement learning, interested users can refer to [45] or the classic textbook [46]. For the terminology of finance, interested users can refer to [11].

B  Accessibility, Accountability and Maintenance

FinRL-Meta is an open-source project on GitHub, with MIT License. Codes, market environments, benchmarks and documentations are available on the GitHub repository: https://github.com/AI4Finance-Foundation/FinRL-Meta. FinRL-Meta has been actively maintained by FinRL community that has over 10K members at the moment. On GitHub, we keep updating our codes, merging pull requests, and fix bugs and issues. We welcome contributions from community members, researchers and quant traders.
C Dataset Documentation and Usages

We organize our contents according to the suggested template of datasheets for datasets.

C.1 Motivation

• For what purpose was the dataset created?
As data is refreshing minute-to-millisecond, finance is a particularly difficult playground for deep reinforcement learning.
In academia, scholars use financial big data to obtain more complex and precise understanding of markets and economics. While industries use financial big data to refine their analytical strategies and strengthen their prediction models. To serve the rapidly growing FinRL community, we creates FinRL-Meta that provides data accessing from different sources and build the data to RL environments. We aim to provide dynamical RL environments that are manageable by users.

We aim to build a financial metaverse, a universe of near real-market environments, as a playground for data-driven financial machine learning.

• Who created the dataset?
FinRL-Meta is an open-source project created by the FinRL community. Contents of FinRL-Meta are contributed by the authors of this paper and will be maintained by members of FinRL community.

• Who funded the creation of the dataset?
AI4Finance Foundation, a non-profit open-source community that shares AI tools for finance, funded our project.

C.2 Composition

• What do the instances that comprise the dataset represent?
Instances of FinRL-Meta are financial data includes: stocks, securities, cryptocurrencies, etc. FinRL-Meta provides hundreds of market environments through an automatic pipeline that collects dynamic datasets from real-world markets and processes them into standard gym style market environments. FinRL-Meta also benchmarks popular papers as stepping stones for users to design new trading strategies.

• How many instances are there in total?
FinRL-Meta does not store data directly. Instead, we provide codes for data accessing, data cleaning, feature engineering, and building into RL environments. Table C.1 provides the supported data sources of FinRL-Meta.
At the moment, there are hundreds of market environments, tens of tutorials and demos, and several benchmarks provided.

• Does the dataset contain all possible instances or is it a sample of instances from a larger set?
With our provided codes, users could fetch data from the data source by properly specifying the starting date, ending date, time granularity, asset set, attributes, etc.

• What data does each instance consist of?
Now there are several types of financial data, as shown in Table C.1:
– Stocks
– Cryptocurrencies
– Securities
– ETFs
– Exchange rate:

Table 4: Supported data sources. OHLCV means open, high, low, and close prices; volume data.

- **Is there a label or target associated with each instance?**
  No. There is not label or preset target for each instance. But users can use our benchmarks as baselines.

- **Is any information missing from individual instances?**
  Yes. In several data sources, there are missing values and we provided standard preprocessing methods.

- **Are relationships between individual instances made explicit?**
  Yes. An instance is a sample set of the market of interest.

- **Are there recommended data splits?**
  We recommend users to follow our training-testing-training pipeline, as shown in Fig. 8. Users can flexibly choose their preferred settings, e.g., in stock trading task, our demo access Yahoo! Finance database and use data from 01/01/2009 to 06/30/2020 for training and data from 07/01/2020 to 05/31/2022 for backtesting.

- **Are there any errors, sources of noise, or redundancies in the dataset?**
  For the raw data fetched from different sources, there are noise and outliers. We provide codes to process the data and built them into standard RL gym environment.

- **Is the dataset self-contained, or does it link to or otherwise rely on external resources?**
  It is linked to external resources. As shown in Table C.1, FinRL-Meta fetch data from data sources to build gym environments.

- **Does the dataset contain data that might be considered confidential?**
  No. All our data are from publicly available data sources.
Figure 8: Overview of the training-testing-trading pipeline in FinRL-Meta.

- Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?
  No. All our data are numerical.

### C.3 Collection Process

- How was the data associated with each instance acquired?
  FinRL-Meta fetch data from data sources. as shown in Table C.1.
- What mechanisms or procedures were used to collect the data?
  FinRL-Meta provides dynamic market environments that are built according to users’ settings. To achieve this, we provide software APIs to fetch data from different data sources. Note that some data sources requires accounts and passwords or have limitations on number or frequency of requests.
- If the dataset is a sample from a larger set, what was the sampling strategy?
  It is dynamic, depending on users’ settings, such as the starting date, ending date, time granularity, asset set, attributes, etc.
- Who was involved in the data collection process and how were they compensated?
  Our codes collect publicly available market data, which is free.
- Over what timeframe was the data collected?
  It is not applicable because the environments are created dynamically by running the codes to fetch data in real time.
- Were any ethical review processes conducted?
  No?

### C.4 Preprocessing/cleaning/labeling

- Was any preprocessing/cleaning/labeling of the data done?
  Yes. For the raw data fetched from different sources, there are noise and outliers. We provide codes to process the data and built them into standard RL gym environment.
- Was the raw data saved in addition to the preprocessed/cleaned/labeled data
  The raw data are hold by different data sources (data providers).
- Is the software that was used to preprocess/clean/label the data available?
  Yes. We use our own codes to do cleaning and preprocessing.

### C.5 Uses

- Has the dataset been used for any tasks already?
  Yes. Thousands of FinRL community members use FinRL-Meta for learning and research purpose. Demos and tutorials are mentioned in Section 3.
- Is there a repository that links to any or all papers or systems that use the dataset?
  1. Research papers that used FinRL-Meta are list here: https://github.com/Al4Finance-Foundation/FinRL/blob/master/tutorials/FinRL_papers.md
Our workshop version of FinRL-Meta \cite{29} appeared in NeurIPS 2021 Workshop on Data-Centric AI.

2. The following three repositories has incorporated FinRL-meta:

- FinRL-meta corresponding to the market layer of FinRL (5.6K stars):
  \url{https://github.com/AI4Finance-Foundation/FinRL}
- ElegantRL (2.1K stars) supports FinRL-Meta:
  \url{https://github.com/AI4Finance-Foundation/ElegantRL}
- FinRL-Podracer:
  \url{https://github.com/AI4Finance-Foundation/FinRL_Podracer}

• What (other) tasks could the dataset be used for?
   Besides the current tasks (tutorial, demo and benchmarks), FinRL-Meta will be useful for the following tasks:

   - **Curriculum learning for agents**: Based on FinRL-meta (a universe of market environments, say ≥ 100), one is able to construct a environment by sampling data samples from multiple market datasets, similar to XLand \cite{47}. In this way, one can apply the curriculum learning method \cite{47} to train a generally capable agent for several financial tasks.
   - To improve the performance for the large-scale markets, we are exploiting GPU-based massive parallel simulation such as Isaac Gym \cite{35}.
   - It will be interesting to explore the evolutionary perspectives \cite{19, 43, 24, 28} to simulate the markets. We believe that FinRL-Meta will provide insights into complex market phenomena and offer guidance for financial regulations.

• Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?
  We believe that FinRL-Meta will not encounter usage limit. Our data are fetched from different sources in real time when running the codes. However, there may be one or two out of ≥ 30 data sources (in Table C.1) change data access rules that may impact future use. So please refer to the rules and accessibility of certain data source when using.

• Are there tasks for which the dataset should not be used?
  No. Since there are no ethical problems for FinRL-Meta, users could use FinRL-Meta in any task as long as it does not violate laws.
  **Disclaimer**: Nothing herein is financial advice, and NOT a recommendation to trade real money. Please use common sense and always first consult a professional before trading or investing.

C.6 Distribution

• Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?
  No. It will always be held on GitHub under MIT license, for educational and research purpose.

• How will the dataset will be distributed?
  Our codes and existing environments are available on GitHub FinRL-Meta repository \url{https://github.com/AI4Finance-Foundation/FinRL-Meta}.

• When will the dataset be distributed?
  FinRL-Meta is publicly available since February 14th, 2021.

• Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?
  FinRL-Meta is distributed under MIT License, for educational and research purpose.

• Have any third parties imposed IP-based or other restrictions on the data associated with the instances?
  No.
• Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?
No. Our data are fetched from different sources in real time. However, there may be one or two out of ≥30 data sources (in Table 4) change data access rules that may impact future use. So please refer to the rules and accessibility of certain data source when using.

C.7 Maintenance

• Who will be supporting/hosting/maintaining the dataset?
FinRL-Meta has been actively maintained by FinRL community (including the authors of this paper) that has over 10K members at the moment. We are still actively updating market environments, to serve the rapidly growing FinRL community.

• How can the owner/curator/manager of the dataset be contacted?
To contact the main developers, we encourage users join our Slack channel: https://join.slack.com/t/ai4financeworkspace/shared_invite/zt-v6701ljm-dz7gI19fhZ1j1qrrY0kg or our mailing list: https://groups.google.com/u/1/g/ai4finance.

• Is there an erratum?
Users can use GitHub to report issues/bugs, and use Slack channel or mailing list to discuss solutions. FinRL community is actively improving the codes, say extracting technical indicators, evaluating feature importance, quantifying the probability of backtesting overfitting, etc.

• Will the dataset be updated?
Yes, we are actively updating FinRL-Meta’s codes and data sources. Users could get information and the newly updated version through our GitHub repository, or join the mailing list: https://groups.google.com/u/1/g/ai4finance.

• If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances
The data of FinRL-Meta do not relate to people.

• Will older versions of the dataset continue to be supported/hosted/maintained?
Yes. All versions can be found on our GitHub repository.

• If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?
We maintain FinRL-Meta on GitHub. Users can use GitHub to report issues/bugs, and use Slack channel or mailing list to discuss solutions. We welcome community members to submit pull requests through GitHub.
D Benchmarks

We have reproduced several papers as users’ stepping stones. Our codes are publicly available for research usage. Users can utilize them as stepping stones for designing trading strategies.

The codes can be found at 
https://github.com/AI4Finance-Foundation/FinRL/tree/master/tutorials

Each notebook gives a full tutorial on how to reproduce the experiment of the paper.

D.1 Stock Trading

We access Yahoo! Finance database and select the 30 constituent stocks (accessed at 07/01/2020) in Dow Jones Industrial Average (DJIA). We use data from 01/01/2009 to 06/30/2020 for training and data from 07/01/2020 to 05/31/2022 for testing. We use technical indicators in our state space, e.g., Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), Commodity Channel Index (CCI), Average Directional Index (ADX), etc.

Code available at: 
https://github.com/AI4Finance-Foundation/FinRL/blob/master/tutorials/1-Introduction/FinRL_StockTrading_Fundamental.ipynb

D.2 Liquidation Analysis

We build a simulated environment of stock prices according to the Almgren and Chriss model. Then we implement the multi-agent DRL algorithms for both competing and cooperative liquidation strategies. This benchmark demonstrates the trade execution task using deep reinforcement learning algorithms. When trading, traders want to minimize the expected trading cost, which is also called implementation shortfall.

Code available at:  
https://github.com/AI4Finance-Foundation/FinRL-Meta/tree/master/tutorials/2-Advance/execution_optimizing

D.3 Explainable Financial Reinforcement Learning

We reproduce [16] that compares the performance of DRL algorithms with machine learning (ML) methods on the multi-step prediction in the portfolio allocation task. We use four technical indicators MACD, RSI, CCI, and ADX as features. Random Forest (RF), Decision Tree Regression (DT), Linear Regression (LR), and Support Vector Machine (SVM) are the ML algorithms in comparison. We use data from Dow Jones 30 constituent stocks to construct the environment. We use data from 04/01/2009 to 03/31/2020 as the training set and data from 04/01/2020 to 05/31/2022 for backtesting.

Code available at: 
https://github.com/AI4Finance-Foundation/FinRL/blob/master/tutorials/2-Advance/FinRL_PortfolioAllocation_Explainable_DRL.ipynb

D.4 Podracer on the Cloud

We reproduce cloud solutions of population-based training, e.g., generational evolution and tournament-based evolution. If GPUs are abundant, users can take advantage of this benchmark to meet the real-time requirement of high-frequency trading tasks. Detailed instructions are provided on our website.

Code available at: 
https://github.com/AI4Finance-Foundation/FinRL_Podracer

D.5 Ensemble Strategy

The ensemble method combines different agents to obtain an adaptive one, which performs remarkably well in practice. We consider three component algorithms, Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), and Deep Deterministic Policy Gradient (DDPG), which have different strengths and weaknesses. Using a rolling window, an ensemble agent automatically selects the best model for each test period. Again on the 30 constituent stocks of the DJIA index, we use data from 04/01/2009 to 06/30/2019 for training, and data from 07/01/2020 to 05/31/2022 for validation and testing through a quarterly rolling window.
D.6 Market Simulator

We simulate the market from the playback of historical limit-order-book-level data and the order matching mechanism. Currently, we simulate at the minute level (i.e. one time step = one minute), which is changeable. The state is a stack of market indicators and market snapshots from the last few time steps. The action is to place an order. We support market orders and limit orders. We also provide several wrappers to accept typically discrete or continuous actions. Rewards can be configured by the participants with the aim of generating policies that optimize pre-specified indicators. In our simulator, we take into account the following factors: 1) temporary market impact; 2) order delay. We do not consider the following factors in our simulator: 1) permanent market impact of limit orders; 2) non-resiliency limit order book.

Code available at: https://github.com/AI4Finance-Foundation/Market_Simulator

E Open Source FinRL-Meta and DAO, DeFi, NFT, Web3

FinRL [31,32] is the first open-source framework to demonstrate the great potential of financial reinforcement learning. It has evolved into an ecosystem, FinRL-Meta, which is a playground for financial reinforcement learning. We believe FinRL-Meta will reshape our financial lives and our open-source community will make sure it is for the better.

Next, we discuss the potential of combining our open-source FinRL-Meta project with the merging technologies of DAO, DeFi, NFT, and Web3.

Decentralized Autonomous Organizations (DAO) has three features: 1). Member-owned communities without centralized leadership; 2). A safe way to collaborate with Internet strangers; and 3). A safe place to commit funds to a specific cause. In FinRL-Meta community, we would like to encourage community members to organize into a DAO fund (essentially a distributed fund) in an ad hoc manner. Members of such a DAO fund will employ smart contracts to crowdfunding, design strategy, implement algorithmic trading, share profit and loss, etc.

Decentralized Finance (DeFi) has three features: 1). A global, open alternative to the current financial system; 2). Products that let you borrow, save, invest, trade, and more; and 3). Based on open-source technology that anyone can program with. We would like to encourage community members to actively explore the potential of financial reinforcement learning technologies in the emerging DeFi based financial markets. For example, using the automated environment layer in Section 4.2, we will connect DeFi trading systems into gym-style market environments.

Non-Fungible Token (NFT) has three features: 1). A way to represent anything unique as an Ethereum-based asset; 2). NFTs are giving more power to content creators than ever before; and 3). Powered by smart contracts on the Ethereum blockchain. In FinRL-Meta community, we would like to encourage community members to release trading strategies and codes in the NFT forms, while a DAO fund will buy these NFTs via smart contracts and then trade in a DeFi system.

Third-Generation of WWW (Web3) incorporates concepts such as decentralization, blockchain technologies, and token-based economics, which is essentially community-ran Internet. Our open-source FinRL-meta community will serve Web3 users a unique infrastructure and playground, where Web3 users can establish DAO funds.

A near-term development in FinRL-Meta community would embrace the Massive Open Online Courses (MOOC) paradigm. For demonstrative and education purposes, our community members (over 10K Github users) are actively creating notebooks, blogs and videos, which can be released in the form of NFTs. Web3MOOC guarantees authorship and ownership (a right to profit) of those NFTs, thus provides a strong incentive for our community members.

---

7Ethereum: https://ethereum.org/en/
8Web3 Foundation: https://web3.foundation/
F  Rights

As the authors of this paper and core developers of FinRL-Meta, we bear all responsibility in case of violation of rights.

Disclaimer: Nothing in this paper and FinRL-Meta repository is financial advice, and NOT a recommendation to trade real money. Please use common sense and always first consult a professional before trading or investing.
Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] More computational cost.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] May lead to future works with higher computational cost.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]