# Curriculum Learning from Smart Retail Investors: Towards Financial Open-endedness

**Kent Wu, Ziyi Xia, Xiao-Yang Liu**
Columbia University
{hw2910,zx2325,xl2427}@columbia.edu

**Shuaiyu Chen**
Purdue University
chen4144@purdue.edu

## Abstract

The integration of data-driven supervised learning and reinforcement learning has demonstrated promising potential for stock trading. It has been observed that introducing training examples to a learning algorithm in a meaningful order or sequence, known as curriculum learning, can speed up convergence and yield improved solutions. In this paper, we present a financial curriculum learning method that achieves superhuman performance in automated stock trading. First, with high-quality financial datasets from smart retail investors, such as trading logs, training our algorithm through imitation learning results in a reasonably competent solution. Subsequently, leveraging reinforcement learning techniques in a second stage, we develop a novel curriculum learning strategy that helps traders beat the stock market.

## 1 Introduction

In the existing vast body of games and simulations Team et al. (2021), finance is a fertile yet formidable ground for open-ended learning applications. The less foreseeable nature of financial markets, exemplified by phenomena such as Black Swan events, poses challenges for agent learning. The markets are characterized by emergent complexity and sporadic patterns, making the prediction and management of financial events, even with abundant information, a daunting task. This raises a pivotal question: Can agents proficiently excel at tasks in the financial domain?

Asset management with financial advisors, which aligns with individual portfolio selections and risk preferences, is typically deemed wise for investors. Robo-advising, in particular, has gained popularity with financial reinforcement learning (FinRL) applications Hambly et al. (2023); Liu et al. (2020). However, several challenges currently impede its real-world adoption:

- **Environment:** The financial markets dynamics are often noisy, with a low signal-to-noise ratio (SNR).
- **Algorithm:** Deep reinforcement learning approaches are often considered black-box algorithms, lacking model interpretability. Many existing works are also subject to look-ahead bias and model overfitting, and thus less empirical and convincing.
- **Expert Demonstration:** There is a variety of traders in the market, such as swing traders, contrarian traders, systematic traders, etc. Valuable perspectives derived from trade logs often remain neglected, while understanding how these entities trade not only provides significant insights into the market dynamics but also informs valuable trading heuristics.
- **Community and Culture:** Financial institutions are conservative in embracing open-source benchmarks and dataset development in the industry, and user privacy concerns and confidentiality agreements surrounding trade logs data have made it difficult for research purposes.

To address these challenges, curriculum learning emerges as a promising approach, enabling the analysis of large-scale financial data by learning a series of incrementally complex tasks. In our context, we commence by emulating the strategies of human experts, subsequently employing reinforcement learning techniques to refine the model further.

**Our contributions** Financial curriculum learning, a two-stage **imitation**-and-**reinforcement** learning method, has the potential of surpassing inherent human performance. By imitating the trading behaviors exhibited in the preserved historical trade logs of alpha traders, an intelligent agent can learn from their expertise. Followed by reinforcement learning techniques, the agent is trained to perform trading decisions through market feedback overhead. This approach, rooted in open-ended learning, emphasizes adaptability and curriculum learning, offering innovative solutions in real-world financial applications.

## 2 Related Works

### 2.1 Curriculum Learning

In order to have an agent that could approach and finally beat the human traders, we adopt the idea of curriculum learning. This approach draws parallels to human education systems, where concepts are introduced gradually, building upon previously learned knowledge. Elman (1993) first proposed the idea of curriculum learning by demonstrating its effectiveness in teaching a recurrent network of simple grammar. Bengio et al. (2011) revisited the concept of curriculum learning by conducting experiments in vision and language tasks. They both draw a conclusion that by presenting examples in a meaningful order, the learner avoids wasting time on noisy or challenging examples that it is not yet prepared to handle effectively. The star product ChatGPT Ouyang et al. (2022) with GPT-3.5 is applying the similar idea with a two-stage framework, pretraining and fine-tuning using reinforcement learning from human feedback (RLHF). Compared to the GPT-3 baseline, it has a remarkable improvement in the accuracy and reliability of its answers. However, existing curriculum learning techniques have primarily been applied to language processing or games with full information. They have not been thoroughly tested to determine whether curriculum learning can achieve satisfactory performance in stochastic and information-hidden environments, such as the financial market.

We aim to validate the effectiveness of curriculum learning in the noisy finance environment. The first stage involves using imitation learning to capture the trading behaviors of smart retail investors. The second stage is to use deep reinforcement learning to further develop additional strategies.

### 2.2 Imitation Learning

In complicated tasks such as Go and Atari games, imitation learning Hussein et al. (2017) is often used to initialize deep neural networks that approach human-level performance. Imitation learning involves training a model to imitate a human's behavior, typically using a dataset of expert demonstrations. This process provides a warm starting point for further refinement using reinforcement learning, which could learn through trial and error to find strategies that surpass human performance. AlphaGo Silver et al. (2016) also uses a two-stage training to learn from expert chess players. It first mimics moves from expert human players with a deep neural network. Afterwards, AlphaGo transitions to reinforcement learning through self-play. By changing the chess expert players into stock smart investors, we extend this meaningful framework in financial market trading. Given a board position, AlphaGo is trying to optimize the next move. Given a market position, our agent is trying to optimize the next trading action.

Therefore, we consider imitation learning as a specific form of supervised learning, and for simplicity, we will refer to them interchangeably in this paper.

### 2.3 Financial Reinforcement Learning

There are many existing researches that have already applied deep reinforcement learning (DRL) in financial applications. Liu et al. (2018) presented a comprehensive process of trading using DRL that involved converting historical market data into gym-style environments, and training using the DDPG algorithm. Zhang et al. (2020) utilized deep Q-learning networks (DQN), policy gradients (PG) and advantage actor-critic (A2C) algorithms for training. Liang et al. (2018) applied three DRL algorithms, including DDPG, PPO, and PG, to utilize an adversarial training method for portfolio management, and demonstrated promising results in backtesting. Hambly et al. (2023) surveyed popular financial applications that researchers have used DRL in, including equity trading, portfolio management, trade execution, market making, bid-ask optimization, and robo-advising.

FinRL Liu et al. (2020) is an open-source library that establishes the full pipeline of financial reinforcement learning: data processing, gym environment design, training DRL agents and backtesting. The work of FinRL-Meta Liu et al. (2022) provides dynamic datasets to produce high-quality market environments and benchmarks for financial reinforcement learning.

Nonetheless, no previous works resolve the instability convergence issue of deep reinforcement learning. Thus they barely have exceeding performance comparing to smart retail investors.

## 3 High-Frequency Retail Trading Activity and Information in the Stock Market

In this section, we will focus on our dataset and answer the following questions:

***Why do we focus on learning the behavior of retail investors?*** One important application of FinTech is the robo advisor, so it is important to understand the behavior of retail investors in order to provide them with advice on investment decisions. The availability of high-frequency financial data + DRL makes it possible to accomplish this task.

***Why is it possible to use a curriculum learning approach to study retail investor trading behavior?*** Typically, investor transaction data is not publicly available. However, a recent important finance study Boehmer et al. (2021) shows that we can obtain data on high-frequency retail trading activities from market orders. Also, we have a broad set of indicators constructed from high-frequency price and trading data to effectively capture real-time information in the constantly evolving stock market. These high-quality retail trading data lay the foundation of open-ended solutions with FinRL.

### 3.1 Dataset Overview

We identify marketable retail purchases and sales from a publicly available database of US equity transaction data (TAQ) following the novel method proposed in Boehmer et al. (2021). It shows that stocks with positive order imbalance (net buying) by retail investors outperform stocks with negative imbalances over the next 5 days. This empirical evidence suggests that retail marketable order imbalances contain information on the price movements of individual stocks and thus reflect ”the wisdom of the crowd” of retail investors.

Our sample consists of daily marketable orders for all US common stocks, with over 11 million daily stock observations from 2010 to 2021. This comprehensive dataset covers more than 6,700 publicly traded equities popular among retail investors in the US equity market. We rigorously validate our sample by replicating the main finding in Boehmer et al. (2021) and test the predictive power of imbalances in retail market order across various groups of stocks formed by sectors (e.g., technology, health care) or size and style (e.g., large cap and growth). We find that investment strategies tracking retail investor trading activities are particularly profitable in small-cap firms and specific sectors such as consumer goods, energy, technology, and healthcare. A diversified portfolio by combining the investment strategies formed by the retail market order imbalances across different size-sector groups delivers an annualized return of $20.5\%$ and a Sharpe ratio of $2.54$ over the sample period of 2010-2021. This return significantly outperforms major US market indices such as S&P 500 and Russell 1000.

The sheer scale and quality of this dataset also make it a valuable resource for researchers and analysts alike. More importantly, it is a highly informative source of alternative data for smart retail investors, serving as a target for imitation learning. Undertaking imitation learning holds the potential to achieve favorable performance, as real retail investors. Going one step ahead, the data of retail investors is valuable for robo advisor. Based on the investor’s historical trading behavior, the robo advisor can have a chance to understand the investor’s behaviors and tastes, and provide them with advice on investment decisions accordingly.

### 3.2 Trading Activity and Performance

One may effectively distinguish retail investors’ marketable orders from institutional ones, as institutions are typically not entitled to fractional penny price improvements with their trades. The authors in Boehmer et al. (2021) have identified marketable retail price-improved orders from the

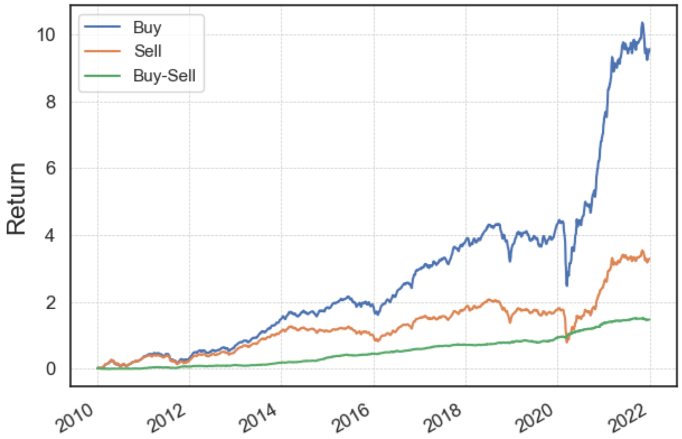

Figure 1: Expected cumulative returns for retail investor trading activity. The expected daily return rate is calculated as the geometric mean over the next 5 days.

publicly available information, which stems from all historical transactions for stocks listed on national exchanges in the U.S.

To largely eliminate potential noise within the market micro-structure, we categorize extreme signals into binary labels based on pre-established thresholds, as follows:

- Buy: Order imbalance greater than the 95th percentile.
- Sell: Order imbalance lower than the 5th percentile.
- Hold: The remaining data points, i.e., order imbalance between the above two marks.

Before these classifications, we have also applied a simple detection filter by considering the trade days before and after to capture such ”anomalies” (or spikes).

We now transform the trading activities into a multi-classification problem, and Figure 1 with our identified trade points reaffirms the effectiveness of smart retail investors. When considering fixed shares to trade each time, the buy portfolio yields approximately a 10-fold return. Conversely, the sell portfolio reflects pessimistic views, resulting in a much lower curve with 3 times return over the 12-year sample period. The significant discrepancy between the two curves highlights the strong trading behaviors exhibited by retail investors. Additionally, implementing a long-short strategy utilizing both portfolios demonstrates an expected return rate of close to 170%.

Note that we use the 5-day geometric mean return as our daily return rate, as buy portfolios tend to significantly outperform sell portfolios over the 5-day period. It may be intriguingly interpreted as investors managing 5 brokerage accounts of equivalent purchasing power, with each account engaging in weekly trading.

### 3.3 Trading Indicators

Human traders themselves are decision-makers upon fundamental stock analysis. Among market participants, retail investors are extremely passionate about the technical analysis over stock charts. Therefore, on top of known retail investors' trade activities, we have curated over 40 trading indicators per stock-day observation. We tend to investigate how these overarching stock indicators contribute to the stock trading decisions of retail investors.

## 4 Problem Formulation

We approach stock trading as a Markov Decision Process (MDP) and formulate it as an optimization problem. Next, we use a year-long snapshot of data from 2021 to merit further investigation into how to study these smart trade investors. The summary statistics can be found in Appendix A.

### 4.1 Assumptions

We facilitate several assumptions to augment the completeness of our problem formulation.

- Long-only strategy: Short selling is not allowed in the trading strategy, as retail investors typically seek undervalued securities.
- Free transaction costs: We assume that no fees are incurred for buying or selling stocks.
- Zero stock dividends: Dividends from stocks are not factored into the trading model.
- Market liquidity: Order execution is always successful at daily close prices without impacting the underlying stock price.
- Trading heuristics: Our imitation target, retail investors, employs informed trading heuristics based on various technical indicators. To capture all the public information available in the financial market that can potentially be used by retail investors, we construct a dataset of 43 high-frequency trading indicators from the Trade and Quote (TAQ) database Bogousslavsky et al. (2023). To the best of our knowledge, we are the first study to implement deep reinforcement learning with such a comprehensive dataset of trading indicators.

### 4.2 MDP Modeling and Market Environment

The secondary market allows investors to trade securities among themselves. However, the market dynamics often present significant noise, making alpha generation and future price prediction challenging. Consequently, we formulate our stock trading task as a concise Markov Decision Process (MDP), incorporating $\mathcal{D}$ stocks into our portfolio. Our objective is to maximize the cumulative return, treating each trading day as a discrete time step denoted by $t$, and making decisions based on the performance of individual stock $i$. Specific details of the MDP of our environment are outlined below:

- State Space $\boldsymbol{s_t} = [b_t, h_t, p_t, F_t]$ :
  - $b_t \in \mathbb{R}$: the remaining cash balance of our accounts and our purchasing power. It is initialized at 100,000 dollars by default.
  - $\boldsymbol{h_t} \in \mathbb{R}^D$: A vector that represents the current holdings in terms of the number of shares of each stock in our portfolio, indicating our position.
  - $\boldsymbol{p_t} \in \mathbb{R}^D$: A vector of the daily closing prices for each stock $i$. Combined with our holdings, this allows us to calculate the market values for each security.
  - $\boldsymbol{F_t} \in \mathbb{R}^{D\times 43}$: A vector of 43 technical indicators that we have curated for stock investors. These are considered stock features and can be used to mine alpha signals.
- Action Space $\boldsymbol{a_t} \in \mathbb{Z}^D$: A set of three discrete trading actions applicable to each underlying stock. Here, $a_t^i$ = 0 signifies a hold action, $a_t^i$ = 1 denotes buying one share, and $a_t^i$ = -1 indicates selling one share of stock $i$ on day $t$. Collectively, $\boldsymbol{a_t} \in \{-1, 0, +1\}^D$ forms an action vector that operates across $D$ stocks.
- Reward $r\left(s_t, a_t, s_{t+1}\right)$: the relative change in portfolio value after taking action $a_t$, transitioning from state $s_t$ to $s_{t+1}$. The reward is calculated as the return rate of portfolio value between day $t$ and day $t+1$, expressed as $\frac{h_{t+1}^T p_{t+1} - h_t^T p_t}{h_t^T p_t}$ in terms of holdings and prices.
- Policy $\pi(\cdot|s)$: the stock trading strategy at state $s$. The policy is determined by the probability $p$, which measures the agent's propensity to either long, short or hold the stocks.

Overall, Figure 2 summarizes our learning-based stock trading environment, where on each trading day, the agent adjusts its strategy based on the available market information and its current portfolio. The agent can choose from three possible actions: buy, sell with a limit of 1 unit per stock, or hold current positions.

Followed by selected actions, the agent receives feedback and observes the consequences on the next trading day. Feedback naturally comprises market rewards, which are defined as the rate of change in portfolio value. Apart from our environment and its rewarding property, we place an expert agent alongside the learning agent to incorporate human feedback, playing the role of supervisor on-site

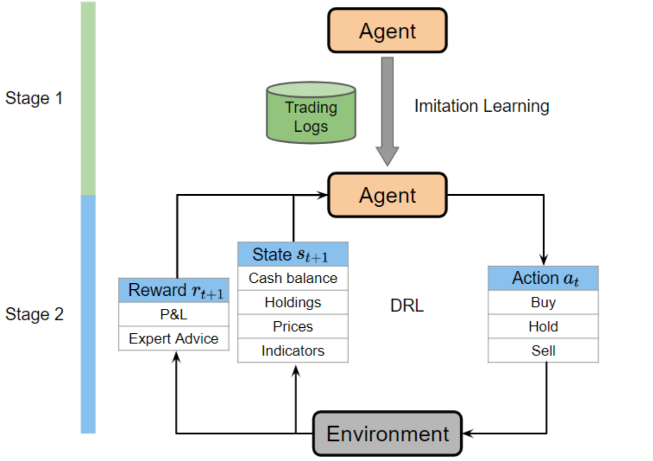

Figure 2: The overview of agent-and-environment interactions.

to prevent one from going astray. More crucially, the two types of existing feedback altogether consolidate the foundation of our curriculum learning solution whose details are disclosed in the next few sections.

# 5 Financial Curriculum Learning

Stock trading involves multiple stock assets, leading to a high-dimensional combinatorial problem. The sole use of either supervised learning or reinforcement learning presents several limitations in this context. Nevertheless, our two-stage financial curriculum learning solution integrates the strengths of both learning approaches. In Figure 2, the first stage, imitation learning, helps the agent approach retail traders fast. Then the second stage deep reinforcement learning makes it possible to have an exceeding performance.

## 5.1 Imitation Learning

Stock prices are known to be non-stationary time series, driven by market sentiments. Coincidentally, our dataset provides expert demonstrations of stock trading, which makes it possible to accomplish this task by imitation learning.

### 5.1.1 Objective Functions

Supervised learning has an upper bound on learning performance because it does not interact with the environment, and reinforcement learning can be unstable to use, due to the low SNR in financial data. Therefore, we employ the idea of curriculum learning, which combines the advantages of both methods, into the following objective:

$$(1-\lambda)\cdot L(a_t^A - a_t^E) + \lambda \cdot r(s_t, a_t^A, s_{t+1}), \tag{1}$$

where

$$L(a_t^A, a_t^E) = \frac{1}{D}\sum^{D} a_t^E \cdot \log\left(p(a_t^A)\right), \tag{2}$$

$$r(s_t, a_t^A, s_{t+1}) = \frac{PV_{t+1} - PV_t}{PV_t} = \frac{(h_t + a_t^A)^T p_{t+1} - h_t^T p_t}{h_t^T p_t}, \tag{3}$$

where

- $a_t^A, a_t^E$: Action of the training agent ($A$) and human expert ($E$) at time $t$;
- $PV_t$: The portfolio value worth on day t;

- $D$: The size of data points at each step. $D$ = 100 for our case;
- $\lambda$: A weight factor that increases over time, with $\lambda = \min\left(1, \frac{k}{T}\right)$ where $k$ being the current time step and $T$ being a hyper-parameter.

Equation 2 is simply the cross-entropy loss as common in multi-classification tasks, whereas Equation 3 is the daily return rate to represent market feedback. With coefficient schedule $\lambda$, we give precedence over learning from smart traders, followed by fine-tuning with reinforcement learning. This allows the agent to adopt domain knowledge provided by human traders and to avoid getting stuck in local minima through the powerful search ability of reinforcement learning.

#### 5.1.2 Normalizing Rewards

The defined objective in Equation 1 is also exemplary of reward shaping. However, in the context of our MDP-based state transition in RL, the environment dynamics are considerably more volatile, particularly when multiple feedback types are considered.

Our environment reveals fast-paced and ever-changing dynamics, encompassing information from both the market and human feedback, as derived from Equations 2 and 3. The scale of these feedback signals may vary, and during environment exploration, the agent is exposed to a broader range of them. Additionally, neural networks are sensitive to extreme gradient values, causing gradient explosion or gradient vanishing. Consequently, for the purpose of dynamics normalization, it becomes imperative to re-balance the feedbacks, respectively.

We introduce a dynamic standardization scheme during training, maintaining running statistics to self-update the agent's distributional parameters based on historical visits. This technique not only contributes to stabler training but also accelerates convergence when neural networks are utilized. The estimators of running mean ($\hat{\mu}_n$), standard deviation ($\hat{\sigma}_n$), and sum of squared differences ($\mathrm{SSD}_n$) for n steps are as follows:

$$\hat{\mu}_n = \frac{(n-1)\cdot\hat{\mu}_{n-1} + x_n}{n}, \tag{4}$$

$$\hat{\sigma}_n = \sqrt{\frac{\mathrm{SSD}_n}{n-1}}, \tag{5}$$

where

$$\mathrm{SSD}_n = \mathrm{SSD}_{n-1} + (x_n - \hat{\mu}_{n-1})\cdot(x_n - \hat{\mu}_n). \tag{6}$$

### 5.2 Deep Reinforcement Learning (DRL)

We use a state-of-the-art actor-critic algorithm, PPO, with tailored engineering to work with our desired action space.

#### 5.2.1 Proximal Policy Optimization (PPO)

PPO Schulman et al. (2017) is an on-policy, actor-critic algorithm. The training procedure for PPO is an artifact of offline training and online searching. The agent collects experiences by interacting with the environment, and then learns and refines the policy after gathering sufficient data with a buffer. A few technical details regarding our PPO implementation are highlighted in Appendix B.

#### 5.2.2 Policy Network

The complex nature of stock trading presents numerous challenges for action simulation. First, with over 3,000 publicly listed securities on Nasdaq and over 8,000 on NYSE as of July 2023, stock selection resembles a multi-armed bandit problem. The number of available equities makes the selection process difficult. Second, the crucial task of deciding which equities to trade—whether to sell, buy, or hold—depends heavily on market timing, significantly impacting the outcomes. In summary, both perspectives contribute to the high-dimensional characteristics of stock trading.

A novel idea is to parametrize our action space. We focus on the top 100 most traded stocks by retail investors and narrow down the action space to represent the three distinct behaviors (buy, sell, and hold) with the smallest possible unit (1 share per stock) at each step. By combining these insights, we

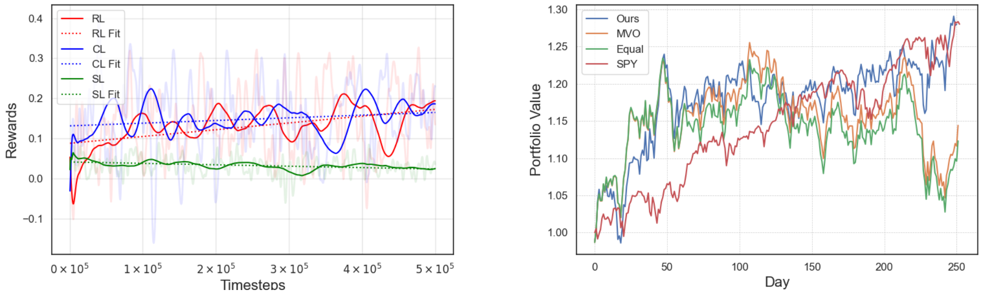

Figure 3: a) Training curves of different methods: supervised learning (SL), reinforcement learning (RL), and curriculum learning (CL). All are smoothed by rolling windows of size 100 and 10, respectively. b) Portfolio comparison over the year 2021. The environment is reset randomly to validate model robustness.

simplify and standardize our problem setting by introducing a multivariate probability distribution $P$ over the 100 stocks with a neural network. Our space dimensionality is largely reduced to $\mathcal{M}^{100\times3}$.

Notably, even with a limited number of discrete actions applicable to individual stocks, considering a pool of 100 stocks would have resulted in an astronomical number of possible combinations—$3^{100}$ actions are infinitely many to search iteratively, which is akin to the complexity encountered in games like Go Silver et al. (2016). Over again, this emphasizes the necessity of employing an actor-critic framework, where the actor network can better approximate the next moves.

## 6 Performance Evaluation

Our proposed curriculum learning method not only facilitates a smoother onboarding process for ”rookie” trading agents but also has the potential to achieve exceptional performance—the student agent surpasses its master.

### 6.1 Experiment Design

We conducted experiments using various approaches for agent training, denoted as supervised learning (SL), reinforcement learning (RL), and our proposed curriculum learning (CL). All these training schemes follow a fashion of offline training and online searching. The experiment configurations and algorithm parameters are specified as follows, and in Appendix C.1,

- SL: $\lambda = 0$ for all timestep $k$;
- RL: $\lambda = 1$ for all timestep $k$;
- CL: $\lambda = \min\left(1, \frac{k}{10^4}\right)$ for all timestep $k$.

### 6.2 Results

We are analyzing our results from two perspectives, machine learning and portfolio performance.

#### 6.2.1 Model Training

In our sample run, we observe distinct training patterns among the different types of agents. The pure RL agent initiates its training with completely random exploration and progressively evolves to master the art of stock trading. On the other hand, the SL agent exhibits an upper limit to their performance, reflecting the constraints of the knowledge imparted by the expert.

Most notably, our CL agent displays stable characteristics during training, as well as convergence speed-up. It leverages the expertise of the trading expert at the early stage of training, resulting in a higher rate of convergence compared to RL. Subsequently, while both RL and CL agents experience diminishing convergence rates after $1 \times 10^5$ time steps, the latter shows higher stability with less

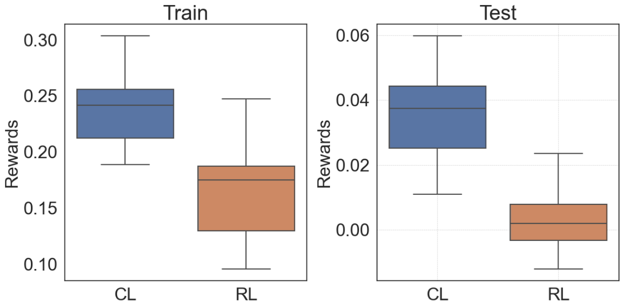

Figure 4: Distribution of total rewards in the defined train (first 250 days) and test (last 52 days) environments over 10 episodes.

oscillation in its training curve. It appears as if the CL agent could fine-tune its own strategy as it gains a deeper understanding of the environment. Although both achieve comparable rewards in our task, the CL agent demonstrates the ability to quickly find high-quality solutions within a significantly reduced timeframe of about 40000 time steps, as illustrated in Figure 3-a).

### 6.2.2 Portfolio Performance

We explore the investment strategy constructed by our CL agent. In Figure 3-b), we compare our best-trained CL agent against common benchmarks, such as S&P 500 ETF Trust (SPY), mean-variance optimization (MVO), and equal weight (also known as buy-and-hold).

Under the exact same portfolio of 100 stock constituents, CL significantly outperforms the equally weighted and mean-variance methods with a higher Sharpe ratio, as well as a lower max drawdown. Moreover, it shows a highly competitive performance compared to SPY, surpassing the latter by almost 2%. See Appendix C.2 for the full results in financial metrics.

We also conduct a traditional train-test evaluation, particularly for CL and RL. The first 200 trade days are defined as the training environment, whereas the latter 52 days are used as the test environment. Both agents are trained with the previous setting until $2 \times 10^5$ timesteps, where their fitted performance becomes close to one another. Figure 4 presents rewards distributions, confirming that the CL agent's performance demonstrates enhanced performance even in out-of-sample back-testing when compared to RL.

## 7 Future Work and Conclusion

Our curriculum learning solution bridges the multi-stage training with a simple yet concise implementation, leading to convergence speed-up and stable training, as well as high-quality portfolio performance. Moreover, we hope to improve the investment skills of retail investors, with the ultimate goal of fostering an efficient, fair market system, where more investors may benefit from this collective good. In the future, we would like to continue our work in the following aspects:

**Institutional Trading** In contrast to retail investors, investigating macro, low-frequency trading heuristics from institutional investors is also promising. Institutional investors are regulated to disclose their quarter-end holdings, coupled with publicly available earnings reports for thousands of companies, which can serve as valuable data sources.

**Multi-agent System** The stock market is also typically regarded as a partially observable Markov decision process (POMDP) involving multiple market participants who adopt diverse trading strategies, frequencies, and underlying assets. Fujimoto & Gu (2021) introduces a minimal offline approach to learning an agent that leverages the advantages of different experts. However, a multi-agent system is another natural approach to model such a multi-strategy mechanism, resembling a hedge fund's business model. Agents may cooperate or compete against each other to maximize collective or individual rewards, respectively.

## REFERENCES

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. *Machine Learning*, 83(1-2):7–49, 2011. doi: 10.1007/s10994-011-5278-7.

Ekkehart Boehmer, Charles M Jones, Xiaoyan Zhang, and Xinran Zhang. Tracking retail investor activity. *The Journal of Finance*, 76(5):2249–2305, 2021.

Vincent Bogousslavsky, Vyacheslav Fos, and Dmitriy Muravyev. Informed trading intensity. *Journal of Finance, Forthcoming*, 2023.

Jeffrey L. Elman. Learning and development in neural networks: the importance of starting small. *Cognition*, 48(1):71–99, 1993. doi: 10.1016/0010-0277(93)90058-4.

Scott Fujimoto and Shixiang (Shane) Gu. A minimalist approach to offline reinforcement learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 20132–20145. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/a8166da05c5a094f7dc03724b41886e5-Paper.pdf`.

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

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

Zhipeng Liang, Hao Chen, Junhao Zhu, Kangkang Jiang, and Yanran Li. Adversarial deep reinforcement learning in portfolio management. *arXiv preprint arXiv:1808.09940*, 2018.

Xiao-Yang Liu, Zhuoran Xiong, Shan Zhong, Hongyang Bruce Yang, and Anwar Walid. Practical deep reinforcement learning approach for stock trading. 2018.

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. *NeurIPS Workshop on Deep Reinforcement Learning*, 2020.

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. *Advances in Neural Information Processing Systems*, 35:1835–1849, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. Trust region policy optimization. In *The 31st International Conference on Machine Learning*, 02 2015.

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

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, Nat McAleese, Nathalie Bradley-Schmieg, Nathaniel Wong, Nicolas Porcel, Roberta Raileanu, Steph Hughes-Fitt, Valentin Dalibard, and Wojciech Marian Czarnecki. Open-ended learning leads to generally capable agents, 2021.

Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep reinforcement learning for trading. *The Journal of Financial Data Science*, 2(2):25–40, 2020.

# A Dataset Statistics

Table 1: Summary statistics of top 100 traded stocks data by retail investors (from 2020-12-31 to 2021-12-31).

| Data Source | Sector | Stocks (Prop) | Annual Return | Volatility | Avg. Position |
|---|---|---|---|---|---|
| Retail Trade Activity | XLB | 2 (2.0%) | -17.4% | 93.1% | 0.044 |
| | XLC | 4 (4.0%) | 24.3% | 47.3% | 0.072 |
| | XLE | 8 (7.9%) | 42.4% | 80.7% | 0.044 |
| | XLF | 6 (5.9%) | 34.7% | 36.1% | 0.071 |
| | XLI | 15 (14.9%) | -2.6% | 48.9% | 0.049 |
| | XLK | 15 (14.9%) | 4.9% | 42.8% | 0.048 |
| | XLP | 5 (5.0%) | 27.8% | 44.7% | 0.067 |
| | XLR | 4 (4.0%) | 0.8% | 59.3% | 0.082 |
| | XLU | – | – | – | – |
| | XLV | 27 (27.0%) | -8.2% | 60.1% | 0.047 |
| | XLY | 14 (13.9%) | 25.7% | 47.5% | 0.045 |

| Data Source | Russell Group | Stocks (Prop) | Annual Return | Volatility | Avg. Position |
|---|---|---|---|---|---|
| Retail Trade Activity | Large | – | – | – | – |
| | Mid | 3 (2.4%) | 43.8% | 32.8% | 0.067 |
| | Small | 36 (29.0%) | 15.9% | 40.5% | 0.005 |
| | Micro | 74 (59.2%) | 6.3% | 55.2% | 0.005 |
| | Nano | 11 (8.8%) | 0.7% | 85.6% | 0.019 |

| Variables | N | Mean | Std | Min | 25% | 50% | 75% | Max |
|---|---|---|---|---|---|---|---|---|
| Price | 25,200 | 45.94 | 87.03 | 0.56 | 10.03 | 23.04 | 44.81 | 729.80 |
| Market Cap | 25,200 | 1,062.26 | 1,309.57 | 9.87 | 369.80 | 678.27 | 1,328.83 | 10,505.10 |
| ret_1 | 25,200 | 0.1% | 3.5% | -75.5% | -1.6% | 0.0% | 1.6% | 87.3% |
| ret_5 | 25,200 | 0.2% | 7.8% | -76.3% | -3.5% | 0.0% | 3.5% | 181.2% |
| Russell Group | 25,200 | 3.78 | 0.64 | 2.00 | 3.00 | 4.00 | 4.00 | 5.00 |
| Position Indicator | 25,200 | 0.05 | 0.38 | -1.00 | 0.00 | 0.00 | 0.00 | 1.00 |

| Categories | 43 high-frequency trading indicators (all lagged by one trading day) |
|---|---|
| Returns | returns over different time zones (daily, overnight, intraday, morning, afternoon) |
| Liquidity | effective spread, realized spread, Kyle lambda, intraday and overnight price impacts |
| Volatility | absolute return and realized volatility (daily, intraday, overnight), variance ratios, 30-min return autocorrelation |
| Trading | trade volumes (daily, intraday, overnight, morning, afternoon), Herfindahl index, the number of 5, 15, 30-min time intervals with trade, buy-minus-sell share/dollar volume, value-weighted average trade price of buys minus sells |

## B Implementation Notes

PPO extends beyond the Trust Region Policy Optimization (TRPO) algorithm Schulman et al. (2015), with a novel clipped objective function to be optimized as:

$$L^{CLIP}(\theta) = \hat{E}_t \left[ \min \left( r(\theta)\hat{A}_t, \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{7}$$

The clip() operation is indicating that $r(\theta)$ is clipped within $[1-\epsilon, 1+\epsilon]$ to control update size. Furthermore, the probability ratio $r(\theta)$ is the ratio between the current policy and the prior policy, and the advantage $\hat{A}_t$ is computed by Monte Carlo estimation in our implementation for simplicity. Both expressions are shown as follows:

$$r(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}, \tag{8}$$

$$\hat{A}(s_t, a_t) = R_t - V(s_t), \tag{9}$$

where $R_t$ is the return from state $s_t$ to the terminal state of the episode, and $V(s_t)$ is the estimated value of state $s_t$ from the critic network.

The final loss is a multi-objective, combining the clipped surrogate loss, the value function loss, and a distributional entropy over $\pi_\theta$, which can be written as:

$$L_{\text{Final}}(\theta) = \mathbb{E}\left[ L^{CLIP}(\theta) - c_1 MSE(V(s_t), R_t) + c_2 H(\pi_\theta) \right] \tag{10}$$

# C Experiment Details

## C.1 Algorithm Hyperparameters

Table 2: PPO Hyperparameters. Our actor and critic are 3-linear-layer networks with 256 hidden neurons. The update frequency is 4 times the maximum length of an episode.

| **Parameter** | **Value** |
|---|---|
| input size | 4501 |
| output size | $100 \times 3$ |
| hidden_units | [256, 256, 256] |
| activation | Tanh() |
| total_timestep | $5 \times 10^5$ |
| update_frequency | 1004 |
| buffer_size | 1004 |
| K_epochs_update | 40 |
| eps_clip | 0.2 |
| gamma | 0.99 |
| lr_actor | 0.0003 |
| lr_critic | 0.001 |
| $c_1$ | 0.5 |
| $c_2$ | 0.01 |

## C.2 Portfolio Performance

Table 3: Portfolio comparison over the year 2021 in terms of financial metrics.

| **Measure** | **Ours** | **Equal** | **SPY** | **MVO** |
|---|---|---|---|---|
| Annual return | 29.6% | 13.8% | 27.9% | 15.9% |
| Annual volatility | 22.3% | 22.1% | 13.1% | 22.0% |
| Sharpe ratio | 1.282 | 0.697 | 1.950 | 0.786 |
| Calmar ratio | 3.247 | 0.825 | 5.146 | 0.959 |
| Stability | 0.546 | 0.004 | 0.931 | 0.008 |
| Max drawdown | -9.1% | -16.8% | -5.4% | -16.6% |
| Omega ratio | 1.231 | 1.116 | 1.381 | 1.132 |
| Sortino ratio | 2.011 | 1.036 | 2.889 | 1.171 |
| Tail ratio | 1.241 | 1.175 | 1.068 | 1.175 |
| Daily value at risk | -0.027 | -0.027 | -0.015 | -0.027 |