LOGIC-GUIDED DEEP REINFORCEMENT LEARNING FOR STOCK TRADING

Zhiming Li[1,*] **Junzhe Jiang**[3,*], **Yushi Cao**[1], **Aixin Cui**[4], **Bozhi Wu**[2], **Bo Li**[3], **Yang Liu**[2]
[1]Continental-NTU Corporate Lab, Nanyang Technological University, Singapore
[2]Nanyang Technological University, Singapore
[3]The Hong Kong Polytechnic University, Hong Kong
[4]The Chinese University of Hong Kong, Hong Kong

ABSTRACT

Previous state-of-the-art trading strategy proposes using ensemble reinforcement learning to combine the advantages of different subpolicies. Despite its improved performance, we observe that this policy is still quite sensitive to market volatility. In this work, we propose a novel framework called SYENS (Program Synthesis-based Ensemble Strategy) which aims to improve the trading strategy's robustness via the program synthesis by sketching paradigm. SYENS is a hierarchical strategy that uses a program sketch as the high-level strategy. The program sketch embeds human expert knowledge of market trends. And based on the program sketch, we adopt the program synthesis by sketching paradigm to synthesize the detailed ensemble strategy. Experimental results demonstrate that SYENS achieves the highest return while retaining low drawdown[1].

1 INTRODUCTION

Deep reinforcement learning (DRL) has achieved state-of-the-art performance on many quantitative trading tasks, e.g., stock trading (Ee et al., 2020; Nan et al., 2022), portfolio allocation(Guan & Liu, 2021; Cui et al., 2023), order execution Fang et al. (2021), etc. Despite its great success, it is found that the DRL model suffers from a lack of robustness due to the low signal-to-noise ratio (SNR) of the market (Liu et al., 2022).Specifically, in order to improve the DRL trading strategies' robustness, the previous state-of-the-art method proposes using the ensemble reinforcement learning paradigm (Yang et al., 2020).

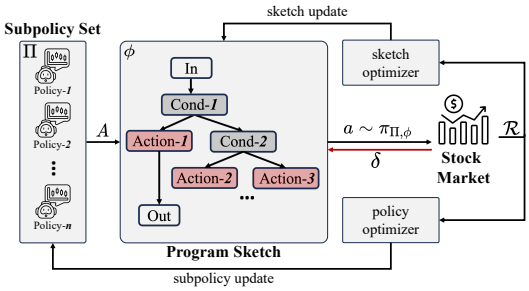

Figure 1: Overview of the SYENS model.

Though this ensemble approach manages to improve over the baselines, we observe that it is still quite sensitive to market turbulence and is likely to suffer from a large drawdown when dealing with market volatility. To improve the robustness of the trading strategy, in this paper, we propose a novel hierarchical, logic-guided ensemble method, called Program Synthesis-based Ensemble Strategy (SYENS). The idea is to embed human expert knowledge regarding the market environment so that the ensemble strategy can better adapt to different kinds of market trends. Specifically, to generate such a logic-guided strategy, we propose using the program synthesis by sketching paradigm (Solar-Lezama, 2008). We first construct a program sketch that embeds the fundamental expert knowledge. The program sketch serves as the backbone of the high-level strategy which describes different kinds of stock market trends and guides the ensemble. Finally, we incorporate the subpolicies as the low-level strategy for the ensemble and optimize the framework using program synthesis techniques.

[*]Equal contribution.
[1]The implementation is available at: `https://anonymous.4open.science/r/syens-7CD9`

Table 1: The comparative results of SYENS and the baseline methods.

Measurement	US Stock						
	DJIA	Original	**SYENS (Ours)**	SYENS w/o sketch	PPO	DDPG	A2C
Annual Return	9.28%	11.3% ± 6.2%	18.2% ± 0.7%	11.7% ± 1.3%	11.5% ± 0.3%	8.9% ± 0.5%	9.6% ± 2.8%
Cummulative Return	91.7%	130.2% ± 97.8%	239.6% ± 15.1%	121.1% ± 20.0%	88.3% ± 54.8%	86.7% ± 6.7%	98.7% ± 36.8%
Anual Volatility	18.9%	31.9% ± 1.3%	16.4% ± 0.2%	16.3% ± 0.5%	12.9% ± 0.1%	17.9% ± 0.4%	16.0% ± 0.8%
Max Drawdown	-38.0%	-49.9% ± 2.4%	-26.2% ± 1.5%	-26.0% ± 0.7%	-19.7% ± 0.3%	-33.6% ± 2.1%	-25.6% ± 4.4%
Sharpe	0.564	0.494 ± 0.17	1.097 ± 0.042	0.765 ± 0.088	0.905 ± 0.02	0.567 ± 0.03	0.653 ± 0.146

2 METHODOLOGY

In this section, we introduce the details of our novel logic-guided trading framework: SYENS (Program Synthesis-based Ensemble Strategy). Figure 1 shows the overview of the SYENS framework. The framework contains two major components: ❶ program sketch, ❷ logic-guided strategy optimization and trading method. The program sketch serves as the high-level strategy that embeds the domain expertise. The detailed program sketch used in this work is shown in Figure 2. Concretely, it consists of multiple conditionals (i.e., `If` statement), the detailed tensor/coefficient to be synthesized are left as *hole* construct: `??`. Each `If` condition describes a different market trend using three kinds of market indicators (i.e., volatility, downside risk and growth rate). The `If` body is the corresponding ensemble strategy of different trends, which adopts the bagging ensemble of multiple subpolicies using weight tensor to be synthesized: `aggregation(actions * softmax(??_n))`. Intuitively, the high-level strategy first decides the current market trend and then selects the corresponding ensemble strategy for the low-level subpolicies. To dynamically update this logic-guided ensemble strategy and use it for trading, we propose a logic-guided strategy optimization and trading method. As shown in Figure 1, we use two distinct optimizers to update the high-level program sketch and the low-level subpolicies respectively. Given a training time period, we first train all the subpolicies independently. Then we use a validation time period for the program sketch optimization, we use the bayesian optimization in this work. Finally, during the test time period (actual trading), given an observation of the market environment, the high-level strategy is executed to decide the market trend and the corresponding ensemble strategy to use.

```
if (volatility(t) < ??_0) and (downside(t) > ??_1):
    then aggregation(actions * softmax(??_2))
else if (volatility(t) < ??_3) and (downside(t) < ??_4):
    then aggregation(actions * softmax(??_5))
else if volatility(t) > ??_6 and growth_rate(t) < ??_7:
    then aggregation(actions * softmax(??_8))
else if volatility(t) > ??_9 and growth_rate(t) > ??_10:
    then aggregation(actions * softmax(??_11))
else
    then aggregation(actions * softmax(??_12))
```

Figure 2: The program sketch used in this work. ??_n denotes a hole (tensor/coefficient) to be synthesized.

3 EXPERIMENTS

We follow Yang et al. and use three actor-critic based policies for the ensemble, namely Advantage Actor Critic (A2C), Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG). We conduct all experiments on the US stock market under the cash trading setting. We compare our method with the state-of-the-art ensemble trading strategy, PPO, DDPG, A2C and the Dow Jones Industrial Average (DJIA)[2] (The DJIA is a stock market index of 30 prominent companies listed on stock exchanges in the United States). We further conduct an ablation study by removing the program sketch, which degenerates the strategy into the vanilla bagging-based ensemble method (denoted as SYENS w/o sketch). For the trading environment, we use the historical daily data from 01/01/2009 to 06/01/2023. We use the data from 01/01/2009 to 2015/09/26 as the in-domain training data and the data from 09/26/2015 to 06/01/2023 as the validation and test data. Please refer to the appendix for concrete implementation details. The results are shown in Table 1. It is obvious that by introducing logical knowledge using the program sketch, the SYENS model manages to achieve state-of-the-art cumulative return while maintaining low maximum drawdown.

4 CONCLUSIONS

In this paper, we propose a novel logic-guided trading framework, called SYENS. Our proposed framework introduces utilizing the program synthesis by sketching paradigm. It is able to generate a hierarchical ensemble strategy that synergistically combines multiple subpolicies with explicit logical conditions. Experimental results under the cash trading setting validate that our method is more robust to market volatility.

[2] `https://en.wikipedia.org/wiki/Dow_Jones_Industrial_Average`

ACKNOWLEDGEMENT

This research is supported by the National Research Foundation, Singapore, and the Cyber Security Agency under its National Cybersecurity R&D Programme (NCRP25-P04-TAICeN) and NRF Investigatorship NRF-NRFI06-2020-0001, the National Natural Science Foundation of China (Grant No. 62106172), the Science and Technology on Information Systems Engineering Laboratory (Grant Nos. WDZC20235250409, 6142101220304), the Xiaomi Young Talents Program of Xiaomi Foundation, and the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contributions from the industry partner(s). Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore, and Cyber Security Agency of Singapore.

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

REFERENCES

Tianxiang Cui, Shusheng Ding, Huan Jin, and Yongmin Zhang. Portfolio constructions in cryptocurrency market: A cvar-based deep reinforcement learning approach. *Economic Modelling*, 119: 106078, 2023.

Yeo Keat Ee, Nurfadhlina Mohd Sharef, Razali Yaakob, and Khairul Azhar Kasmiran. Lstm based recurrent enhancement of dqn for stock trading. In *2020 IEEE Conference on Big Data and Analytics (ICBDA)*, pp. 38–44. IEEE, 2020.

Yuchen Fang, Kan Ren, Weiqing Liu, Dong Zhou, Weinan Zhang, Jiang Bian, Yong Yu, and Tie-Yan Liu. Universal trading for order execution with oracle policy distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 107–115, 2021.

Mao Guan and Xiao-Yang Liu. Explainable deep reinforcement learning for portfolio management: an empirical approach. In *Proceedings of the Second ACM International Conference on AI in Finance*, pp. 1–9, 2021.

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.

Abhishek Nan, Anandh Perumal, and Osmar R Zaiane. Sentiment and knowledge based algorithmic trading with deep reinforcement learning. In *Database and Expert Systems Applications: 33rd International Conference, DEXA 2022, Vienna, Austria, August 22–24, 2022, Proceedings, Part I*, pp. 167–180. Springer, 2022.

Armando Solar-Lezama. *Program synthesis by sketching*. University of California, Berkeley, 2008.

Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. Deep reinforcement learning for automated stock trading: An ensemble strategy. In *Proceedings of the first ACM international conference on AI in finance*, pp. 1–8, 2020.

5 APPENDIX

5.1 IMPLEMENTATION DETAILS

In this section, we concretely introduce the implementation details of our approach and the baselines. We first introduce the state and action space of the reinforcement learning agent. Then we illustrate the detailed training setup.

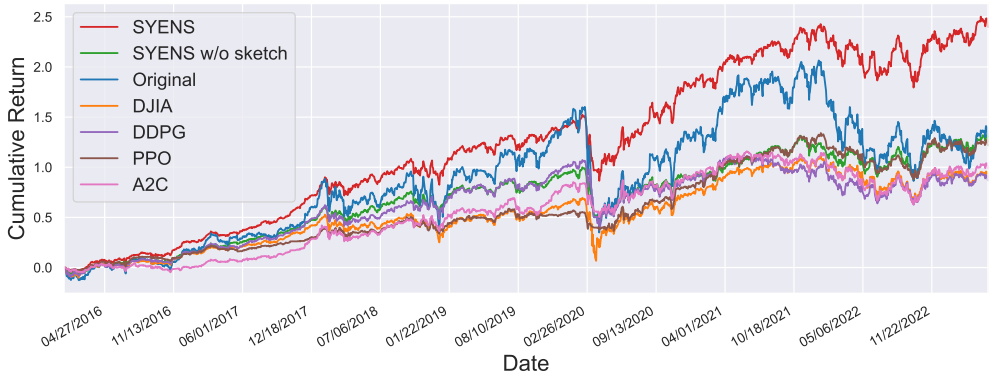

Figure 3: Cumulative return curve of different methods (the results are averaged over 5 random seeds).

State. Our state representation $s \in \mathbb{R}^{331}$ contains 9 major indicators, namely, available balance at the current time step, adjusted close price of each stock, shares owned of each stock, Moving Average Convergence Divergence (`macd`, `macds`), Bollinger Bands (`boll_ub`, `boll_lb`), Relative Strength Index (`rsi_30`), Commodity Channel Index (`cci_30`), Directional Index(`dx_30`), and Simple Moving Average (`close_30_sma`, `close_60_sma`). The concise definitions of the last 6 types of technical indicators are as follows[3]:

- **Moving Average Convergence Divergence** (`macd`, `macds`): MACD measures the difference between two exponential moving averages. MACDS stands for Moving Average Convergence Divergence Signal, which calculates the nine-day exponential moving average of the MACD line.
- **Bollinger Bands** (`boll_ub`, `boll_lb`): Bollinger Bands measures the volatility of the asset. The upper band `boll_ub` and the lower band `boll_lb` can identify potential overbought and oversold levels.
- **Relative Strength Index** (`rsi_30`): Relative Strength Index measures the magnitude of recent price changes to evaluate overbought or oversold conditions in the asset. We use a window size of 30.
- **Commodity Channel Index** (`cci_30`): Commodity Channel Index measures the deviation of the asset's price from its statistical average. We use a window size of 30 days.
- **Directional Index** (`dx_30`): Directional Index measures the strength of a trend in the asset. We use a window size of 30 days.
- **Simple Moving Average** (`close_30_sma`, `close_60_sma`): Simple Moving Average measures the average price of the asset. We use both the window size of 30 days (`close_30_sma`) and 60 days (`close_60_sma`).

Action. For a single stock, the action space is defined as $\{-k, \ldots, -1, 0, 1, \ldots, k\}$, where k and $-k$ represent the maximum number of shares we can buy and sell, we set it to be 100 in this work.

Training setup. We follow Yang et al. and use three actor-critic based policies for the ensemble, namely Advantage Actor Critic (A2C), Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG). For all the subpolicies, we use a 2-layer multilayer perception (MLP) with 64 neurons within each layer. We train all subpolicies with the Adam optimizer and the reward scaling factor is set as $1\mathrm{e}{-10}$. The replay buffer size is set to be $10,000$ for the three policies. We follow Yang et al. Yang et al. (2020) to recursively update the training-validation-test time step t by $\gamma = 3$ months forward and retrain all the subpolicies during each quarter. We set the initial portfolio value to be \$1, 000, 000, the transaction costs to be 0.1% of the value of each trade (either buy or sell).

[3]Please refer to `https://pypi.org/project/stockstats/` for the concrete details.

5.2 CUMULATIVE RETURN CURVE VISUALIZATION

We further show the visualization of the cumulative return curves of different methods during the learning process, the results are shown in Figure 3. It is obvious that due to the market trend-aware design, SYENS is (1) much better in capturing good trading opportunity and achieve the highest cumulative return, (2) much more robust to market turbulence and presents the lowest drawdown when encountering a market crash (e.g., 10/18/2021 to 05/06/2022).