Proceedings of the 4th International Conference on Mathematical Physics and Computational Simulation
DOI: 10.54254/2753-8818/2026.33535

Qwen3:4 Verification of the Feasibility of the PPO Strategy in Quantitative Trading

Haohan Zhang

GCTB-NSU Joint Institute of Technology at Guangzhou College of Technology and Business, Guangzhou College of Technology and Business, Foshan, China
zhh085266@outlook.com

Abstract. Single reinforcement learning strategies tend to be frequently traded and it's also not very good at risk control. This article puts forward some quantitative trading methods. This method uses the Qwen3:4b to lead the PPO. This paper is mostly going to be talking about the candidate trading action that comes out of a PPO. And it is hoped that this design will be able to run more smoothly. This paper will use the daily data of the publicly available SPYETF. The PPO is used to output initial trading actions. The Qwen3:4b model then makes a review of the said action in an offline setting and labels it either keep or drop. After that, this paper learns the rule by using a small agent model with these labels. And this small model would do the action filter thing when running the system online. Experimental results show that there are 467 action interceptions after using the new method and then the annualized return and its ratio will be reduced. That means it stopped some of the redundant actions but didn't do much for helping with key decisions. In summary, this paper shows that it can be done, technically speaking. The following will need more varied datasets to improve this system further.

Keywords: Qwen3:4b, Proximal Policy Optimization, Quantitative Trading, SPYETF

1. Introduction

Year after year, the use of reinforcement learning in financial transactions has been paid much attention [1-3]. It is different from supervised learning. Supervised is mainly about predicting how the price goes up and down. Reinforcement Learning: It emphasizes learning how to make decisions so as to make more money in the long term within a continuously operating market by continuously interacting among states, actions, and rewards. Therefore, the more appropriate problem for its use is when to buy, when to sell, and how much to hold, continuously making a decision on them, hence it has been very important in trading research.

Just using reinforcement learning-based trading is not without problems, too. The model gets swayed by the market's noise, local profit signals, it buys repeatedly, trades frequently, and its response isn't sufficient when there's a lot of price movement. Large language models are really good at knowing about rules and looking for conditions [1-3]. This article takes the role of Qwen3:4b, being a risk checker which does not exist online, and runs alongside the online execution

DOI: 10.54254/2753-8818/2026.33535

of the trading framework. The large model gives labels on the trading action, which is given by PPO. And this is the whole thing for qwen3:4b going to be using rl.

The achievements of the main part are as follows: Take, for instance, the trading framework formed by this article, which includes offline verification as well as online implementation, which can improve how the trading system runs. The article didn't guess the future price; it only wanted to know what action to keep for classification. This approach for just checking out actions, can dodge the very big risks connected with guessing the price and this way of joining rule filtering together with a proxy model was also proved and carried out. After all these are done, the problem of the trading strategies that keep on changing will be solved. And leaves a way for further changes on this system to do it this way.

2. Methods

2.1. Data sources and preprocessing

This paper uses the SPY ETF data, which is publicly available with daily market prices. Data were cut up by time order. The paper took January 1, 2022, to be the dividing line; the data prior to this line was used for training: The model trained on the train set learned some basic trading moves. After the dividing line, it became the test set. This paper chooses the first 500 trading days as the test data and uses it to check the true performance of each strategy.

Table 1. State characteristics and action definitions

Category	Name	Meaning
State Feature	ret1	One-step return
State Feature	ma_ratio	Moving-average ratio
State Feature	rsi	Relative strength index
State Feature	vol	Volatility
State Feature	position	Position state
State Feature	drawdown	Historical drawdown
Action Definition	0 = hold	Hold
Action Definition	1 = buy	Buy
Action Definition	2 = sell	Sell

In Table 1, the feature and action are what were used in this paper. The state features are mainly composed of single-step yield, Moving Average Ratio, Relative Strength Index, Volatility, Holding and Drawdown. The decisions were just hold, buy, or sell. It's pretty simple, this design. On the other hand, so that PPO can conveniently understand what kind of trading, at the same time, it is also very easy for Qwen3:4b to conveniently enter the last step of doing actions. In the article, the method used is simply modifying those transactions that are going to be carried out by using these states. Qwen3:4b is just doing a binary select job during the stage with no online tagging. The large model is deciding if it should keep going or stop now. When the system is really running in the environment, it will use this selected 0 or 1 to give an action according to ppo [4]. If it says keep, then the system will do what the PPO originally did. If the result is rejection, then it forces the original action to be held. Based on this conversion rule, the system can finally give us 3 real actions which are hold, buy and sell [5,6].

DOI: 10.54254/2753-8818/2026.33535

2.2. Reinforcement learning trading environment and reward function

The trading environment of this work is based on the OpenAI Gym interface. The environmental state includes these six dimensions of the market features. Action space is set to be 3 types, which means not doing anything, which is also called hold (0), buy (1), sell (2). PPO is the main body of the reinforcement learning algorithm, which learns how to learn the candidate action generation policy in the training set.

In the trading environment, there are 3 main components: state, action and reward. The state is composed of market info+technical indicators, the action is the target position ratio decided on by the trading systems, the reward is how well the strategy does during the current round of trading (profits), and this paper also added a penalty for risky behavior as well.

The reward function is defined as:

$$reward = step_return - \lambda \times drawdown - \gamma \times turnover \tag{1}$$

In this function, step_return represents the current trading cycle's return, drawdown represents the strategy's drawdown amplitude, and turnover represents the trading frequency. By penalizing the drawdown and trading frequency, it is possible to reduce the excessive trading behavior of the strategy and enhance the risk control ability.

2.3. Qwen3:4b-guided PPO trading framework

This article has put the reinforcement learning algorithm and the guidance mechanism of the LLM in the same chapter. To put it simply, PPO makes out different trading actions at first and Qwen3:4b looks if there are any risks of them not being online. This system takes the labels it got from this check to teach a very small stand-in model. With respect to actually carrying out the online phase, however, this proxy model simply depends upon that in order to block what PPO does. Figure 1 shows the overall process of PPO quantitative trading guided by qwen3:4b.

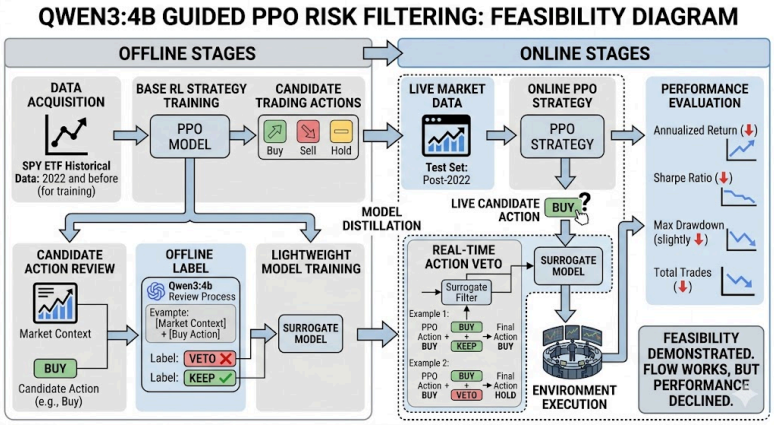

Figure 1. Overall process of PPO quantitative trading guided by Qwen3:4b (Picture credit: Original)

PPO learns some basic trading strategies here, and it outputs a set of candidates during the test time. Qwen3:4b doesn't actually do any trading online, it's more like a sort of offline risk reviewer. Its primary job is deciding if an action should stay or be thrown out [7-9]. This paper has also considered that it's very slow when want to calculate large local models on the go. What it gives out is often not too dependable. Therefore, all of the results from Qwen3:4b that were not online were

DOI: 10.54254/2753-8818/2026.33535

turned into a very small proxy model. So when you go online, you don't have to bring that big model along every time, which will save some costs in total.

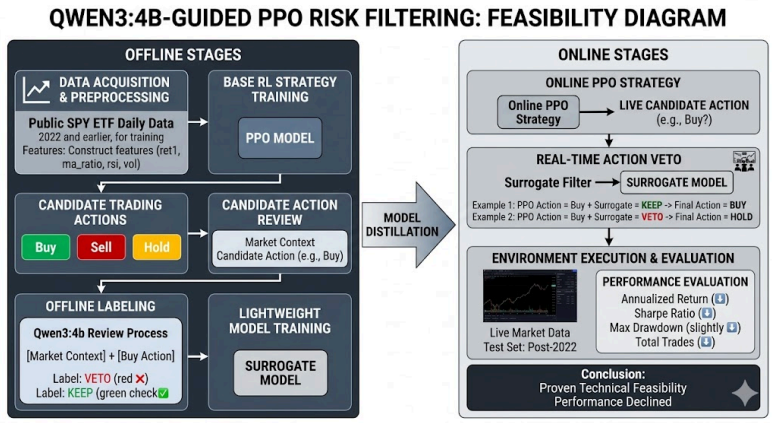

Figure 2. Qwen3:4b offline annotation and proxy distillation workflow diagram (picture credit: original)

The stage where the system was not online for inspection was actually a task for Qwen3:4b to make a binary choice - either to retain or to reject. Retaining meant continuing with the original action of PPO, while rejecting meant not performing this action and forcibly converting it into holding [10]. In this paper, in order to obtain more accurate labels, not all data was directly given to Qwen3:4b It first passed all the data through a normal process, and samples that seemed safe were directly classified as retention, while samples that seemed risky were directly classified as rejection. The remaining samples that could not be classified in either way were then given to Qwen3:4b to assign structured labels. Figure 2 illustrates the experimental design and the detailed process of these labeling rules.

3. Experiment

3.1. Experimental design

This article specifically places Qwen3:4b at the place of not going online to tag it. It's like it is checking for any danger first, before letting the smaller proxy model take on the task of running online. Take advantage of the advantages of a big model's rule understanding and reduce the price of going online at the same time. This way is now probably the most important use, because it shows that it could really do everything.

As for this part of the experiment, this paper has presented 5 different strategies for comparison in this article. The buy-and-hold, double moving average is a common bottoming strategy. The pure PPO is just the building block for RL, the PPO with rule filtering is mostly to test if adding some rules to prevent has had an impact. The PPO with the large model proxy is the main method in the article. Specifically, it's to see if the big model not going online to lead and the little one doing so will still function properly. In this article, there are six indices selected in all to evaluate the ability of each strategy, such as Annualized Return Rate and Sharpe Ratio. How many times has it filtered, which is really counting how many times the small block stopped what PPO wanted to do. As can be seen from the experimental data, the first fifty were used, and thirty-nine were altered. Training the little model, and really used 46 samples, and its performance on the ten validation sets was not too bad.

DOI: 10.54254/2753-8818/2026.33535

3.2. Experimental results and analysis

Table 2 presents the main results of the five strategies on the test set. Buy-and-Hold and MA(10/20) currently only output the annualized return rate and volatility, PPO+Rule Filter, and PPO+Qwen3Surrogate use the complete results in the current experiment log.

Table 2. Comparison of different strategies' performance

Strategy	Annual Return	Volatility	Sharpe	Max Drawdown	Trades	filter_triggered
Buy-and-Hold	1.7722%	19.5305%	—	—	—	—
MA(10/20)	5.7729%	12.6347%	—	—	—	—
PPO	1.7562%	19.5501%	0.0898	-24.4955%	1	0
PPO + Rule Filter	1.7562%	19.5501%	0.0898	-24.4955%	1	344
PPO + Qwen3 Surrogate	1.4764%	19.5459%	0.0755	-24.4696%	1	467

From Table 2 can see that the comparison of Pure PPO against Buy-and-Hold shows no great advantage. And there is just 1 number of transactions. It shows that right now, the PPO can't really make good trade decisions when it gets tested. Although PPO + Rule Filter triggered 344 filter actions, its Annual Return Rate, Volatility, Sharpe Ratio, and Maximum Drawdown are basically the same as Pure PPO. This demonstrates that rules are being filtered out right now, which is just getting rid of some unnecessary actions and hasn't gotten rid of the main trading nodes that decide how the capital curve will go.

Let's have a look at PPO+Qwen3surrogate: And it used up 467 filters. The return rate per year decreased from 1.7562 % to 1.4764%, the Sharpe ratio decreased from 0.0898 to 0.0755%, and the max drawdown changed a little bit from -24.4955%to -24.4696%. It means that Qwen3's surrogate is now able to intervene on PPO candidates more often. But it is mostly on weak signals; it is more of a repetitive action than bringing an incredible amount of return to you. Overall, it's an improvement of some kind.

Table 3. Statistical table of different strategy actions distribution

Strategy	Hold Count	Buy Count	Sell Count	Effective Trades
RL	0	499	0	0
PPO + Qwen3 Surrogate	467	32	0	8

When combined with Table 3, it can be seen that pure PPO almost always outputs "buy" within the testing range, while PPO + Qwen3 surrogate changed many "buy" to "hold". This result indicates that the main function of the Qwen3 surrogate at present is to compress and filter candidate buying actions, rather than the entire holding path. Therefore, it is now more like an action filter rather than a complete re-decision maker.

DOI: 10.54254/2753-8818/2026.33535

3.3. Visual analysis

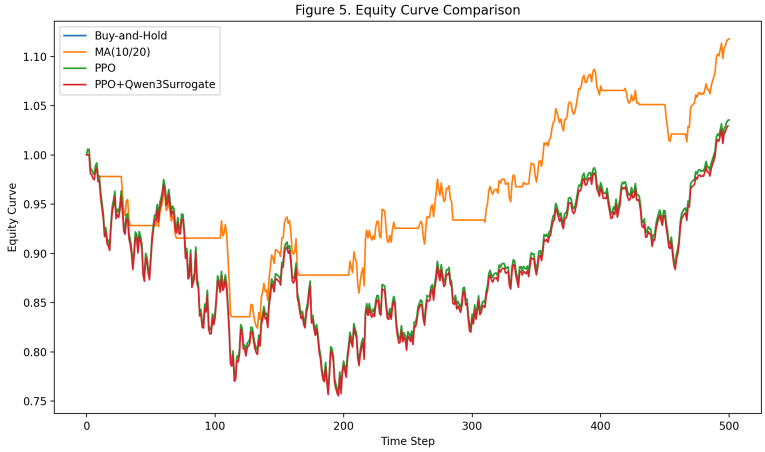

Figure 3. Comparison of fund curves for different strategies (picture credit: original)

In Figure3 can see the capital curves for just PPO and PPO with an agent model. From the graph can be seen that there is little difference between these 2 curves: Strategy with the Agent model had a slightly smaller total return at the end of testing and this is also in line with the change in the revenues that were recorded from table 2. In the above graph, and can see that the result is correct. After making an all-around analysis of the table result and the capital curve could see that the current agent filter is kind of stable in executing trade action. The main job of this block is to stop any unneeded trading action. And it does not really help much with final returns. According to the data showing that there are 39 out of 50 actions that have been modified during the offline annotation phase, and can confirm that the technical route feasibility of offline big model annotation + light agent online execution is correct. Because currently have a small number of samples, it's too early to see how well Qwen3:4b can give out information. This objective fact should also be stated as true in the conclusion.

4. Conclusion

This article specifically looks at how to apply the Qwen3:4b-guided PPO strategy for quantitative trading on the SPY ETF. The study has made a more complete transaction method process, including getting public history data, and making some preparations for PPO. And then hand it off to Qwen3:4b for an offline risk check and have the proxy model do its thing for on-the-fly interception. I'm not predicting what the prices will do in the future, But rather it focuses on the binary classification of reviews, which are kept or rejected. The system to map the results of reviews into a final trading decision is actually what this article is talking about for its main content.

From the existing experiments can be seen that the whole process in which Qwen3:4b took part in Offline Annotation and Proxy Model Distillation was carried out. Now, it is also able to do the interception and filtering tasks for preliminary actions quite steadily. Compared to using just the PPO strategy, the number of times the PPO with proxy model caused an interception among the tests was 467 times. Its annual return rate is now down from 1.752% to 1.4764%, and its Sharpe ratio too, which is now at 0.0755. And found that in case it's mostly about suppressing redundancies and very high-risk weak signals. This doesn't really do much to make the final return any better. This article is just in this phase to conclude that it can be done. It's not like can say for sure that it works better as a

DOI: 10.54254/2753-8818/2026.33535

whole. Future work will need to increase the amount of offline annotation for Qwen3:4b Also, future research will have to make clearer definitions about which samples are higher risk and lower risk. It needs to have even more market characteristics and info. The only way for it to get better at predicting things in really complicated situations, and getting its predictions right on the important spots where people do business, is if this goes somewhere with real returns in the future.

References

[1] Li, Y., Wang, S., Ding, H., et al. (2023). Large language models in finance: A survey. In Proceedings of the 4th ACM International Conference on AI in Finance, 374-382.
[2] Ding, H., Li, Y., Wang, J., Chen, H., Guo, D., & Zhang, Y. (2024). Large language model agent in financial trading: A survey. arXiv preprint arXiv: 2408.06361.
[3] Gupta, R., & Sharma, V. (2024). Finance-specific large language models: Applications and challenges. Decision Support Systems, 176, 114069.
[4] Benhenda, M. (2025). FinRL-DeepSeek: LLM-infused risk-sensitive reinforcement learning for trading agents. arXiv preprint arXiv: 2502.07393.
[5] Xiong, G., Deng, Z., Wang, K., Cao, Y., Li, H., Yu, Y., ... & Xie, Q. (2025, July). Flag-trader: Fusion llm-agent with gradient-based reinforcement learning for financial trading. In Findings of the Association for Computational Linguistics: ACL 2025 (pp. 13921-13934).
[6] Lopez-Lira, A. (2025). Can large language models trade? testing financial theories with llm agents in market simulations. arXiv preprint arXiv: 2504.10789.
[7] Darmanin, A., & Vella, V. (2025, November). Language model guided reinforcement learning in quantitative trading. In 2025 3rd International Conference on Foundation and Large Language Models (FLLM) (pp. 405-412). IEEE.
[8] Long, W., Zeng, W., Zhang, X., & Zhou, Z. (2025). Integrating Large Language Models and Reinforcement Learning for Sentiment-Driven Quantitative Trading. arXiv preprint arXiv: 2510.10526.
[9] Cao, B., Wang, S., Lin, X., Wu, X., Zhang, H., Ni, L. M., & Guo, J. (2025). From deep learning to LLMs: a survey of AI in quantitative investment. arXiv preprint arXiv: 2503.21422.
[10] Kou, Z., Yu, H., Luo, J., Peng, J., Li, X., Liu, C., ... & Guo, Y. (2024). Automate strategy finding with llm in quant investment. seed, 1, 3.