

# News-Aware Direct Reinforcement Trading for Financial Markets

**Qing-Yu Lan, Zhan-He Wang, Jun-Qian Jiang,  
Yu-Tong Wang, and Yun-Song Piao**

School of Physical Sciences  
University of Chinese Academy of Sciences

Beijing, 100049, China

{lanqingyu19, wangzhanhe19, jiangjunqian21}@mails.ucas.ac.cn,  
{wangyutong, yspiao}@ucas.ac.cn

## Abstract

The financial market is known to be highly sensitive to news. Therefore, effectively incorporating news data into quantitative trading remains an important challenge. Existing approaches typically rely on manually designed rules and/or handcrafted features. In this work, we directly use the news sentiment scores derived from large language models, together with raw price and volume data, as observable inputs for reinforcement learning. These inputs are processed by sequence models such as recurrent neural networks or Transformers to make end-to-end trading decisions. We conduct experiments using the cryptocurrency market as an example and evaluate two representative reinforcement learning algorithms, namely Double Deep Q-Network (DDQN) and Group Relative Policy Optimization (GRPO). The results demonstrate that our news-aware approach, which does not depend on handcrafted features or manually designed rules, can achieve performance superior to market benchmarks. We further highlight the critical role of time-series information in this process.

## 1 Introduction

The inherent complexity and volatility of financial markets pose significant challenges to high-quality investment decision-making and undermine the reliability of traditional trading signals (Hambly et al. 2023), especially for cryptocurrency markets (Drozdz et al. 2023; Wei et al. 2023). In tasks such as stock portfolio management, each decision is usually driven by an integrated and diverse information flow with varying timeliness and forms, including market data, technical indicators, and market sentiment. Manual trading struggles to process these signals at scale and stay consistent under time pressure, which slows execution and weakens risk control. This motivates automated, data-driven quantitative systems that can fuse diverse signals and optimize returns while managing current market risk.

Reinforcement Learning (RL) serves as a powerful framework for building automated quantitative trading systems, enabling agents to explore complex market environments and continually update their policies to optimize strategies and maximize returns. However, markets are driven not only by past prices and volumes but also by news, which delivers extra information shocks and triggers market regime changes. Leaving out news makes the whole market condition only partially observable and increases non-stationarity. In early

studies, financial news was processed with dictionary-based methods (e.g. sentiment lexicons), the appearance of large language models (LLMs) (e.g. (Brown et al. 2020)) offers the possibility of automated unstructured textual information processing, rapidly extracting sentiment signals from news text, thereby enabling the development of RL agents that leverage news datasets in a reliable manner. Therefore, in recent years, LLMs have been investigated for financial trading and portfolio management, with strong results in sentiment extraction and explanation generation.

Despite the success of RL methods in data-driven decision making, in financial markets, successful RL approaches typically rely on well-designed handcrafted features. These technical factors or indicators were developed for equity markets and may not be applicable to the cryptocurrency markets. Moreover, handcrafted technical indicators often generalize poorly. For instance, the moving average feature can effectively capture trends but may incur substantial losses in mean-reversion markets (Poterba and Summers 1988). Similarly, the utilization of news information is also based on manually designed rules.

Based on these concerns, in this paper, we explore whether news data can be incorporated directly with raw price and volume as observable inputs to the RL agent without handcrafted features or manually designed rules. The overview of our pipeline is shown in Figure 1. We extract sentiment from finance-related news using an LLM, convert it into structured features (e.g. sentiment scores and risk scores) and then integrate them with raw market prices and volume. Within the RL agent, an LSTM or Transformer encoder is employed as the front-end network to process the merged time-series inputs. We tune hyperparameters on the validation set and conduct backtesting on the test set, finding that the framework can achieve competitive performance relative to market benchmarks and to agents without LLM-derived news sentiment or sequence-based network.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 details our methodology, including how we use LLM to conduct sentiment features extraction, the reinforcement learning architectures, and the procedures for dataset preprocessing and hyperparameters tuning. Section 4 presents the evaluation results for different RL agents and discusses the findings we seek to establish. Section 5 concludes the whole paper.

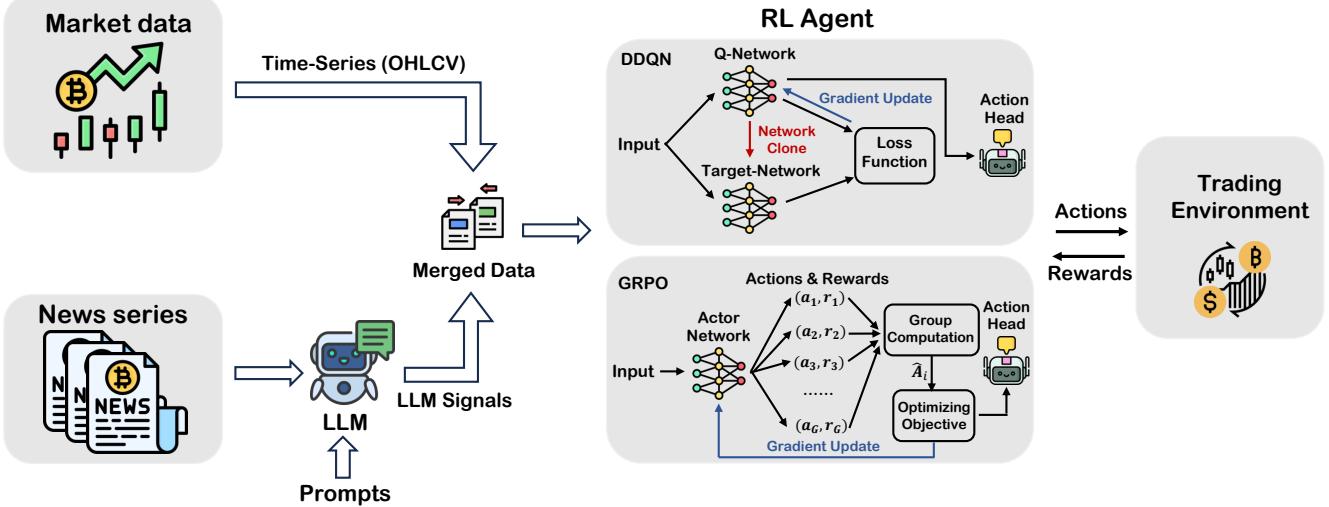


Figure 1: Schematic of the proposed news-aware reinforcement learning framework for financial trading. The system comprises an LLM that analyzes financial news; its output signals are combined with market data and then input to an RL agent. The RL agent utilizes DDQN/GRPO algorithms with various network architectures to process the integrated time-series data, and the action head’s output actions interact with the trading environment to generate rewards.

## 2 Related Work

Reinforcement Learning (RL) serves as a powerful framework for building automated quantitative trading systems, enabling agents to explore complex market environments and continually update their policies to optimize strategies and maximize returns. Neuneier (1995) was the first to introduce deep reinforcement learning into the financial domain, represented by the use of Q-learning. Subsequently, Moody et al. (1998); Moody and Saffell (2001) applied actor-based RL to finance and employed recurrent neural networks (RNNs). Since then, a wide variety of RL methods have been explored. These studies usually rely not only on historical price and volume data, but also on technical indicators and/or other handcrafted factors. For example, Zhang et al. (2019) compared algorithms such as Deep Q-learning Networks (DQN), Policy Gradients (PG) and Advantage Actor-Critic (A2C) incorporated both discrete and continuous action spaces, and carefully designed the reward function. Zou et al. (2024) adopted a combination of an LSTM network and the PPO algorithm, while Huang et al. (2024) proposed a novel BiLSTM-Attention architecture coupled with the RL SARSA algorithm. All of these models take technical indicators or factors as part of their inputs.

There are also some attempts in RL that do not rely on handcrafted features, but instead start directly from raw price and volume data. Deng et al. (2016) used the deep neural network to extract features, which were then input into an RL agent with RNN. Liang et al. (2018); Théate and Ernst (2021) input price and volume data into RL agents and applied different algorithms for RL. Taghian et al. (2021) employed an encoder-decoder architecture to extract features from raw data, which were then fed into the RL agent.

Financial news can significantly influence market price movements and should be considered as an additional state input, distinct from price information. Bollen et al. (2011) was the first to focus on the impact of news on financial price

movements, analyzing text sentiment using OpinionFinder and the Google Profile of Mood States (GPOMS). Loughran and McDonald (2011) analyzed financial text sentiment using a dictionary-based approach. Although these methods enable large-scale processing of text sentiment, they still exhibit significant manual involvement. In recent years, the development of large language models (LLMs) and their strong performance in Natural Language Processing (NLP) have provided an alternative approach for financial text sentiment analysis(Huang et al. 2023; Wu et al. 2023). Liu et al. (2020, 2022) incorporated the sentiment scores assigned by LLM to news together with price data as the state input to the reinforcement learning agent. Unnikrishnan (2024) constructed a sentiment-based reward for integration of sentiment analysis. Benhenda (2025) applied sentiment scores to perturbatively adjust the agent’s decision-making actions. Arshad et al. (2025) manually aligned the sentiment scores with price data using a market-aware module before inputting them into the agent. These studies incorporate news information into reinforcement learning through manually specified rules and/or also include technical indicators. In our work, we explore whether news information can be directly incorporated into a reinforcement learning framework without handcrafted features or manually designed rules.

## 3 Methodology

The core process of framework involves: (1) leveraging LLM to extract sentiment and risk signals from financial news and integrating them with historical market prices into a time-series input; (2) processing the integrated sequence through a LSTM or Transformer-based network within the RL agent to learn temporal patterns; and (3) generating trading actions via a policy head optimized with RL to maximize financial returns; (4) tuning model hyperparameters using the validation set and evaluating model via backtesting on the test set. In the following, we will present the detailed architecture of

our proposed framework.

## Sentiment Features Extraction

The first stage of the framework involves extracting sentiment scores and risk scores from financial news using LLMs with a robust and informative prompt following (Dong et al. 2024; Benhenda 2025) The prompt consists of a task specification that defines the analysis objective and a integer scoring system (i.e., 1–5 for sentiment and risk levels). The template of the prompt is illustrated in Figure 2. We input the prompt into the Gemini-2.5-flash model (Comanici et al. 2025) to assign sentiment and risk scores to the news items. To maximize the utilization of API resources, the news items are processed in batches for scoring. We have validated that, as long as the context length limit is not exceeded, the impact on the scoring results is negligible compared to the inherent randomness of the LLM itself.

## Reinforcement Learning Architectures

We consider both on-policy and off-policy reinforcement learning algorithms. For off-policy learning, we employ Double DQN (DDQN), which is one of the enhancements of Deep Q-Network algorithm. For on-policy learning, we use a variant algorithm of the Proximal Policy Optimization (PPO), the Group Relative Policy Optimization (GRPO) algorithm.

Traditional Q-learning maintains a Q-value table and iteratively updates the Q-value through the Bellman equation. This kind of algorithm will have the problem of dimensionality disaster when the state/action space becomes extremely large. DQN combines Q-learning methods with deep neural networks (DNN) to estimate Q-value, calculates the temporal-difference loss (TD-loss), and perform a gradient descent step to make an update. Among the enhancements of DQN, DDQN mitigates the overestimation bias of Q-values by decoupling action selection from value estimation. Specifically, it uses the current Q-network to select actions while employing the target network to evaluate the Q-values. The TD-loss used in DDQN is:

$$L(\theta) = \mathbb{E}[(r + \gamma \max_{a'} \hat{Q}(s', \max_a Q(s', a)) - Q(s, a))^2] \quad (1)$$

where  $\hat{Q}$  is the target network with weights parameter  $\theta$ ,  $s'$  denotes the next-time-step state, and  $a'$  denotes the action that maximizes  $\hat{Q}$ .

PPO is an RL algorithm introduced by OpenAI and widely used in financial trading tasks (Schulman et al. 2017; Lele et al. 2020), it builds upon the principles of Trust Region Policy Optimization (TRPO). PPO simplifies TRPO by replacing the constraint with a specialized clipped objective function. This function restricts the ratio between the probabilities of the new and old policies, preventing the policy from too rapid changes that could destabilize training. Specifically, the optimizing objective is:

$$L(\pi_\theta) = \mathbb{E}[\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)A_t)] \quad (2)$$

where  $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta, old}(a_t|s_t)$  denotes the probability ratio between new policy  $\pi_\theta$  and old policy  $\pi_{\theta, old}$ ,  $A_t$  is the advantage function at time  $t$ ,  $\epsilon$  is the clipping parameter that restricts the large changes between old and new policies.

GRPO is an improved version of PPO, proposed by the DeepSeek team (Shao et al. 2024). In GRPO algorithm,  $A_t$  is replaced with  $\hat{A}_{i,t}$  in Eq.(2), where  $\hat{A}_{i,t} = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$  represents the relative reward between different groups without the use of critic network. Since the value function used in PPO is usually implemented as a separate model of comparable size to the policy network, it imposes a significant memory and computational overhead.

We investigate two classes of network architectures in our proposed method: multilayer perceptron (MLP) and sequence-based architectures, specifically LSTM and Transformer networks (Graves 2012; Vaswani et al. 2017). We use MLPs to assess the impact of temporal sequences in our framework, in which data at a single time point is utilized. In sequence-based nets cases, we first feed the state into the sequence network, then input the output from the last time step into a single-layer MLP to obtain the action. Specifically, In the case of Transformer, we adopt an encoder stack structure with learnable positional encodings, a linear input projection maps input features to the model dimension before the encoder. It should be noted that when using different network architectures, the same MLP/sequence-based network is consistently applied across all components within a given algorithm (DDQN/GRPO). The AdamW optimizer (Loshchilov and Hutter 2017) is adopted for weight optimization throughout the training process.

## Dataset Preprocessing

The whole dataset, including market 1-minute OHLCV (i.e. open, high, low, close, volume) time-series sourced from Binance Exchange<sup>1</sup> for BTC/USDT and news text scrapped from Yahoo Finance related to Bitcoin<sup>2</sup>. We set the sentiment and risk scores for the interval between two successive news items to be governed by the preceding one. The time range is from 2019-12-31 00:00:00 to 2024-01-24 21:48:00. We divided the whole dataset into training, validation and test area using a chronological split to prevent look-ahead bias and ensure realistic performance evaluation, see Figure 3. The training set encompasses the initial 70% of the timeline, the validation set covers the subsequent segment from 70% to 85%, and the test set comprises the remaining portion from 85% to the end. During training, the agent learns from the training interval, while the validation set guides hyper-parameter tuning and agent selection, final performance is assessed on the test set.

## Hyper-parameters Tuning

It is well-known that reinforcement learning is sensitive to the choice of hyper-parameters. To make a fair comparison, we tune the hyper-parameters to efficiently explore the hyper-parameter space across all model configurations. Our tuning process is designed as follows: agents are trained on the training set, while performance on the validation set guides both hyper-parameter selection and early stopping. During training, for each hyper-parameter combination (i.e. trial), if the

<sup>1</sup><https://data.binance.vision/>

<sup>2</sup>[https://huggingface.co/datasets/edaschau/bitcoin\\_news](https://huggingface.co/datasets/edaschau/bitcoin_news)

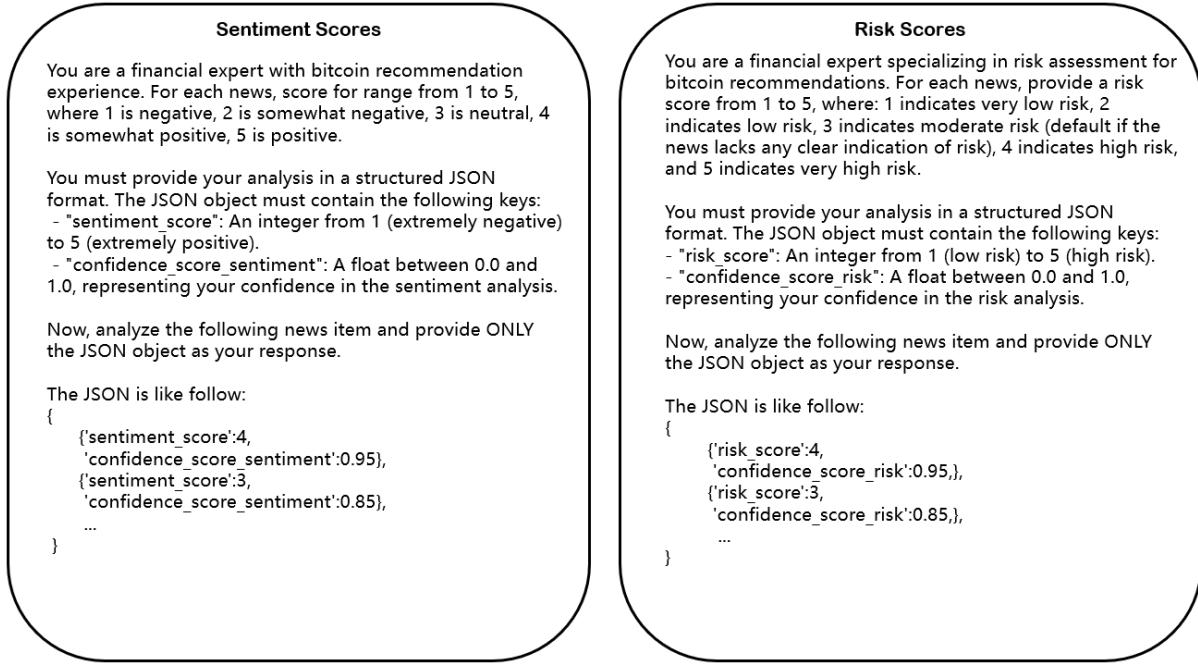


Figure 2: The format of input prompt that guides LLMs to generate sentiment scores and risk scores. It consists of the task specification, the scoring system and the output format example.

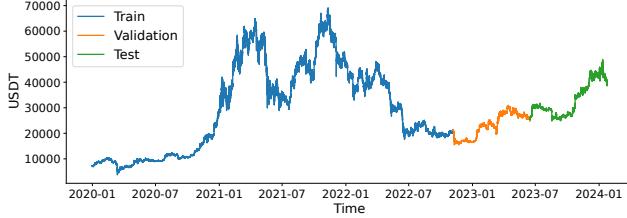


Figure 3: BTC (1-minute) Prices timeline division in our framework. Blue part: training set. Orange part: validation set. Green part: test set.

highest average return on the validation set does not increase for five continuous evaluations, the trial is terminated prematurely. These hyper-parameters including network architecture hyper-parameters (e.g. hidden dimensions, number of layers), sequential model hyper-parameters (e.g. window size, number of attention heads), agents training hyper-parameters (e.g., learning rate, batch size), and algorithm-specific hyper-parameters (e.g. exploration rate for DDQN, clip ratio for GRPO). The detailed hyper-parameters and their corresponding value ranges are summarized in Appendix A Table 3.

## 4 Performance Evaluations

As a proof-of-concept study, we consider a simple discrete action space with only three actions: short, long one BTC, and hold. We consider a stop-loss/take-profit threshold of 0.1% to simulate practical risk management constraints. We randomly sample 3,000-minute consecutive trading periods from the dataset for training and validation. Specifically, during hyper-parameter tuning, we use the statistical mean of cumulative

returns over 256 periods sampled from the validation set as the optimization target.

Finally, we similarly sample 256 periods from the test set to compute the average cumulative return over 3,000-minute trading periods. Our evaluation results are summarized in Table 1 as following, For each algorithm (DDQN, GRPO), we test different network architectures (MLP, LSTM, and Transformer). In addition to the full models that leverage LLM-derived news–sentiment signals, we conduct ablations for the LSTM and Transformer backbones in which the LLM signals are disabled and only market time-series inputs are provided, which isolates the contribution of news sentiment to trading performance. We consider two simple ways of utilizing the agents. The first is to use the agent that performs best on the validation set. The second is to use the top 10 agents with the best performance on the validation set and take the statistical average of their results to reduce the impact of randomness. In addition to evaluating the average results over 3,000-minute periods, we also perform a full backtest over the entire test period, as show in Figure 4, the cumulative returns are shown in Table 2, where the BTC market baseline return is 56% over the test period.

Our evaluation results shows that the proposed news-aware RL framework achieves higher cumulative returns than the BTC market benchmarks on the test set. This advantage holds for both DDQN and GRPO algorithms, indicating the robustness of our framework to model choice variance.

On the other hand, results in Figure 4, Table 1 and Table 2 show the architectural contrast. For LSTM-based RL agents, adding LLM-derived news sentiment consistently raises cumulative returns relative to the LSTM without news, indicating the contribution of news information. For Transformer-based agents , the contribution from news is weaker. This

Table 1: Averaged cumulative returns (in USDT) for different RL algorithms with various network architectures, the optimal performance of the agents are highlighted in bold font.

Networks	DDQN		GRPO	
	Top1	Top10	Top1	Top10
MLP	80.6	153	203.2	151.5
LSTM	<b>329.8</b>	<b>338</b>	<b>447.5</b>	<b>289.5</b>
Transformer	307.1	223.8	227	219.4
LSTM (Without LLM signal)	201.9	118.1	135.4	265.9
Transformer (Without LLM signal)	283.8	199.3	272.1	224.9

Table 2: Full backtest cumulative returns (percentage change) for different RL algorithms with various network architectures, the optimal performance of the agents are highlighted in bold font.

Networks	DDQN		GRPO	
	Top1	Top10	Top1	Top10
MLP	114.9%	91 %	59.9%	83.7%
LSTM	124.5%	<b>119%</b>	<b>124.5%</b>	<b>106.8%</b>
Transformer	112%	95.8%	79.1%	92.2%
LSTM (Without LLM signal)	47%	67.8%	68.3%	89%
Transformer (Without LLM signal)	<b>131.8%</b>	66.7%	54.4%	69.2%

suggests that Transformer encoder may be less sensitive than LSTM at dealing with news information and capturing temporal dependencies between time-series data.

Moreover, sequence-based agents (LSTM, Transformer) consistently outperform the MLP-based agents for both DDQN and GRPO algorithms, confirming the value of modeling prices and sentiment signals as continuous time-series rather than isolated inputs. Between the two sequence models, LSTM-based agents outperform Transformer-based agents in our setting. One potential reason is that Transformers we used are not explicitly tailored for time-series modeling and we did not introduce time-series specific adaptations to the Transformer architecture in our framework, whereas LSTM can naturally process time-series data with inherent causal structures.

Overall, the aforementioned evalution results illustrate that our proposed framework can exceed the BTC market benchmarks. Incorporating LLM-derived news-sentiment inputs yields higher returns than the corresponding model without LLM signals, confirming the value of news analysis. Moreover, sequence-based agents (LSTM and Transformer) consistently outperform MLP agents, indicating that modeling prices and sentiment as continuous sequences rather than isolated inputs is essential for effective algorithmic trading.

## 5 Conclusion

This paper introduces a news-aware RL framework for financial trading that leverages sequence-based networks to process raw market prices and volume with news sentiment features directly. We show that even without handcrafted features or manually designed rules, RL agents can feasibly utilize news sentiment features derived from large language

models, together with price data, as time-series inputs processed by LSTM or Transformer architectures. Evaluation results show that our proposed framework can outperform the market baseline in both DDQN/GRPO algorithms, they also demonstrate the importance of news events utilization and the effectiveness of leveraging sequence-based network architectures to capture temporal dependencies between time-series data. It opens a promising direction for future research on incorporating news information into financial market trading with minimal or no reliance on manual intervention.

This work serves as a proof of concept, focusing on evaluating the feasibility of incorporating LLM-derived news sentiment signals and sequence-based network architectures without handcrafted features. Further research can develop fully optimized trading strategies for practical deployment and return maximization. For instance, rather than the simple averaging of top-performing agents adopted in our study for evaluation, the more efficient utilization of hierarchical multi-agent frameworks probably be required for enhanced collective decision-making in the future. Furthermore, the action space and risk management employed in this study are simplified, and practical trading applications would require more advanced designs. In addition, future work could explore architectures better suited to time-series modeling to capture market prices and sentiment dynamics more effectively.

## References

- Hambly B, Xu R, Yang H, et al. 2023. Recent advances in reinforcement learning in finance *Mathematical Finance*, 2023, 33(3): 437-503.

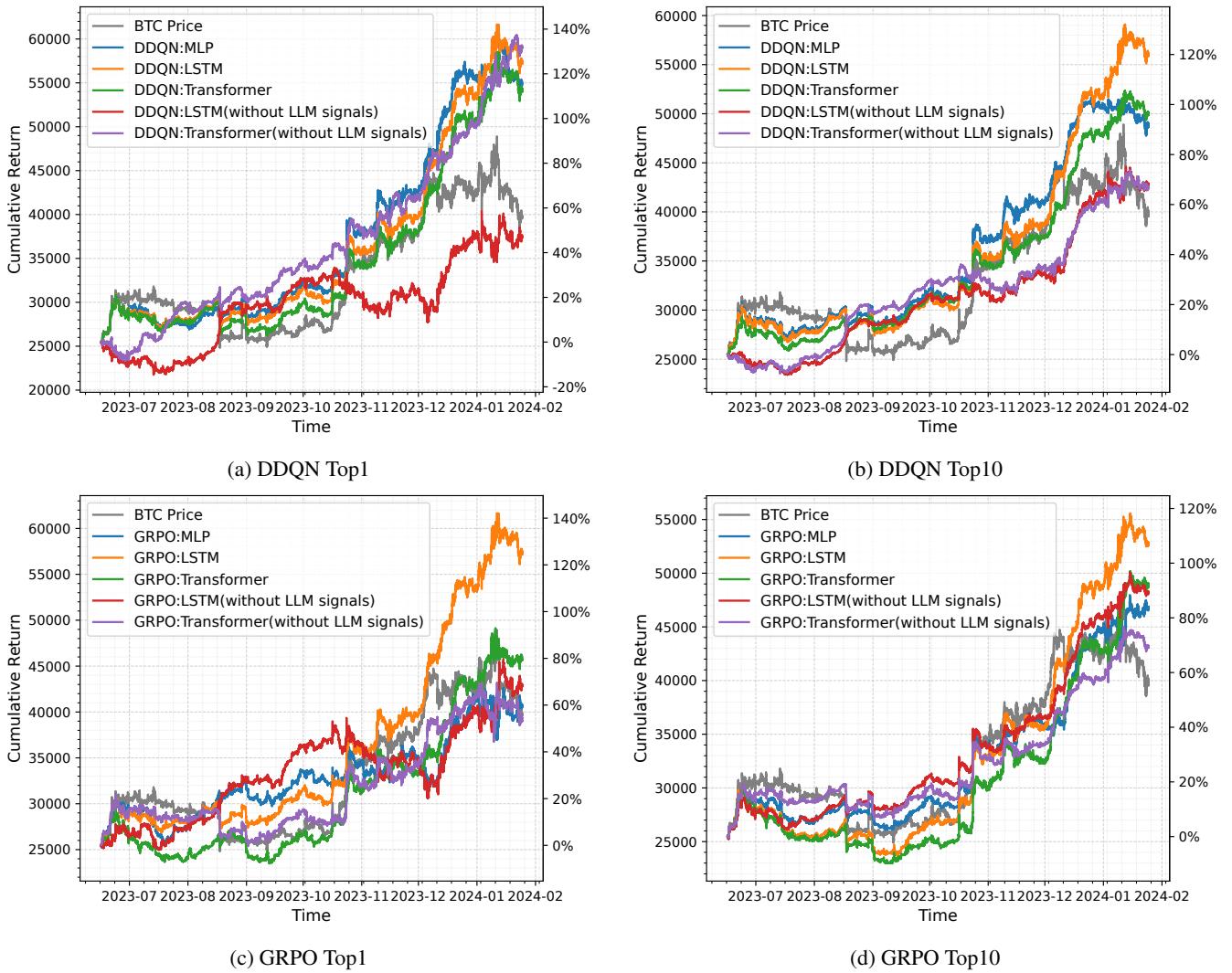


Figure 4: Generalization performance of validation-selected Top-K agents on the BTC test set: DDQN (top row) and GRPO (bottom row). Each panel marks cumulative return on the test set in USDT terms (left-side axis) and percentage terms (right-side axis). Columns correspond to  $K \in \{1, 10\}$ ; for  $K = 10$ , curves are averaged across the top- $K$  agents ranked by validation performance. The gray curve denotes the BTC prices baseline, while colored curves indicate distinct network architectures (MLP, LSTM, Transformer) and the presence/absence of an LLM-derived news-sentiment signals.

Drozd S, Kwapien J, Watorek M. 2023. What is mature and what is still emerging in the cryptocurrency market? *Entropy*, 2023, 25(5): 772.

Wei et al.(2023) Y, Wang Y, Lucey B M, et al. 2023. Cryptocurrency uncertainty and volatility forecasting of precious metal futures markets *Journal of Commodity Markets*, 2023, 29: 100305.

Moody J, Saffell M. 2001. Learning to trade via direct reinforcement *IEEE Transactions on Neural Networks*, 2001, 12(4): 875-889.

Deng Y, Bao F, Kong Y, et al. 2016. Deep direct reinforcement learning for financial signal representation and trading *IEEE Transactions on Neural Networks and Learning Systems*, 2016, 28(3): 653-664.

Brown T, et al. 2020. Language models are few-shot learners *Advances in Neural Information Processing Systems*, 2020, 33: 1877-1901.

Liu X Y, et al. 2022. FinRL-meta: Market environments and benchmarks for data-driven financial reinforcement learning *Advances in Neural Information Processing Systems*, 2022, 35: 1835-1849.

Théate T, Ernst D. 2021. An application of deep reinforcement learning to algorithmic trading *Expert Systems with Applications*, 2021, 173: 114632.

Taghian M, Asadi A, Safabakhsh R, et al. 2021. A reinforcement learning based encoder-decoder framework for learning stock trading rules *arXiv preprint arXiv:2101.03867*, 2021. Zou J, et al. 2024. A novel deep reinforcement learning based

- automated stock trading system using cascaded LSTM networks *Expert Systems with Applications*, 2024, 242: 122801.
- Li Y, et al. 2023. Large language models in finance: A survey *Proceedings of the Fourth ACM International Conference on AI in Finance*, 2023.
- Shen Y, Zhang P K. 2024. Financial sentiment analysis on news and reports using large language models and finbert *2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS)*. IEEE, 2024: 717-721.
- Wu S, Irsoy O, Lu S, et al. 2023. Bloomberggpt: A large language model for finance *arXiv preprint arXiv:2303.17564*, 2023.
- Huang A H, Wang H, Yang Y, et al. 2023. FinBERT: A large language model for extracting information from financial text *Contemporary Accounting Research*, 2023, 40(2): 806-841.
- Konstantinidis T, et al. 2024. Finllama: Financial sentiment classification for algorithmic trading applications *arXiv preprint arXiv:2403.12285*, 2024.
- Lopez-Lira A, Tang Y. 2023. Can ChatGPT forecast stock price movements? Return predictability and large language models *arXiv preprint arXiv:2304.07619*, 2023.
- Yu Y, Yao Z, Li H, et al. 2024. Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making *Advances in Neural Information Processing Systems*, 2024, 37: 137010-137045.
- Yang H, Zhang B, Wang N, et al. 2024. Finrobot: An open-source ai agent platform for financial applications using large language models *arXiv preprint arXiv:2405.14767*, 2024.
- Xiao Y, et al. 2024. TradingAgents: Multi-agents LLM financial trading framework *arXiv preprint arXiv:2412.20138*, 2024.
- Liu X Y, Wang G, Yang H, et al. 2023. FinGPT: Democratizing internet-scale data for financial large language models *arXiv preprint arXiv:2307.10485*, 2023.
- Sutton R S, Barto A G. 1998. Reinforcement learning: An introduction Cambridge: MIT Press, 1998.
- Kim A, Muhn M, Nikolaev V, et al. 2024. Financial statement analysis with large language models *arXiv preprint arXiv:2407.17866*, 2024.
- Darmanin A, Vella V. 2025. Language Model Guided Reinforcement Learning in Quantitative Trading *arXiv preprint arXiv:2508.02366*, 2025.
- Unnikrishnan A. 2024. Financial news-driven llm reinforcement learning for portfolio management *arXiv preprint arXiv:2411.11059*, 2024.
- Arshad S, Ameer H, Azhar N, et al. 2025. FinRL Contest 2025 Task 1: Market-Aware In-Context Learning Framework for Proximal Policy Optimization in Stock Trading Using DeepSeek *2025 IEEE 11th International Conference on Intelligent Data and Security (IDS)*. IEEE Computer Society, 2025: 76-78.
- Liu X Y, Yang H, Chen Q, et al. 2020. FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance *arXiv preprint arXiv:2011.09607*, 2020.
- Xiong G, Deng Z, Wang K, et al. 2025. FLAG-Trader: Fusion LLM-Agent with Gradient-based Reinforcement Learning for Financial Trading *arXiv preprint arXiv:2502.11433*, 2025.
- Liang Z, Chen H, Zhu J, et al. 2018. Adversarial deep reinforcement learning in portfolio management *arXiv preprint arXiv:1808.09940*, 2018.
- Zhang Z, Zohren S, Roberts S. 2019. Deep reinforcement learning for trading *arXiv preprint arXiv:1911.10107*, 2019.
- Cheng J, Chin P. 2024. Sociodojo: Building lifelong analytical agents with real-world text and time series *The Twelfth International Conference on Learning Representations*, 2024.
- Rajpoot P K, Parikh A. 2023. GPT-FinRE: In-context learning for financial relation extraction using large language models *arXiv preprint arXiv:2306.17519*, 2023.
- Wang S, et al. 2023. Alpha-GPT: Human-AI interactive alpha mining for quantitative investment *arXiv preprint arXiv:2308.00016*, 2023.
- Cheng Y, Tang K. 2024. GPT's idea of stock factors *Quantitative Finance*, 2024, 24(9): 1301-1326.
- Huang Y, Wan X, Zhang L, et al. 2024. A novel deep reinforcement learning framework with BiLSTM-Attention networks for algorithmic trading *Expert Systems with Applications*, 2024, 240: 122581.
- Sarlakifar F, et al. 2025. A Deep Reinforcement Learning Approach to Automated Stock Trading, using xLSTM Networks *arXiv preprint arXiv:2503.09655*, 2025.
- Nassirtoussi A K, Aghabozorgi S, Wah T Y, et al. 2014. Text mining for market prediction: A systematic review *Expert Systems with Applications*, 2014, 41(16): 7653-7670.
- Yang Y, Tang Y, Tam K Y, et al. 2023. Investlm: A large language model for investment using financial domain instruction tuning *arXiv preprint arXiv:2309.13064*, 2023.
- Tan M, et al. 2024. Are language models actually useful for time series forecasting? *Advances in Neural Information Processing Systems*, 2024, 37: 60162-60191.
- Yu X, et al. 2023. Temporal data meets LLM-explainable financial time series forecasting *arXiv preprint arXiv:2306.11025*, 2023.
- Jin M, Wang S, Ma L, et al. 2023. Time-lm: Time series forecasting by reprogramming large language models *arXiv preprint arXiv:2310.01728*, 2023.
- Benhenda M. 2025. FinRL-DeepSeek: LLM-infused risk-sensitive reinforcement learning for trading agents *arXiv preprint arXiv:2502.07393*, 2025.
- Wang Y, Xu J, Ma F, et al. 2025. FinZero: Launching Multi-modal Financial Time Series Forecast with Large Reasoning Model *arXiv preprint arXiv:2509.08742*, 2025.
- Schulhoff S, Ilie M, Balepur N, et al. 2024. The prompt report: a systematic survey of prompt engineering techniques *arXiv preprint arXiv:2406.06608*, 2024.
- Gu J, et al. 2024. Adaptive and explainable margin trading via large language models on portfolio management *Proceedings of the 5th ACM International Conference on AI in Finance*, 2024.

- Lima Paiva F C, et al. 2021. Intelligent trading systems: A sentiment-aware reinforcement learning approach *Proceedings of the Second ACM International Conference on AI in Finance*, 2021.
- Chen Q, Kawashima H. 2025. Adaptive Alpha Weighting with PPO: Enhancing Prompt-Based LLM-Generated Alphas in Quant Trading *arXiv preprint arXiv:2509.01393*, 2025.
- Yu Y, et al. 2025. Finmem: A performance-enhanced LLM trading agent with layered memory and character design *IEEE Transactions on Big Data*, 2025.
- Schulman J, et al. 2017. Proximal policy optimization algorithms *arXiv preprint arXiv:1707.06347*, 2017.
- Lele S, et al. 2020. Stock market trading agent using on-policy reinforcement learning algorithms *Available at SSRN 3582014*, 2020.
- Benhenda M. 2025. FinRL-DeepSeek: LLM-infused risk-sensitive reinforcement learning for trading agents *arXiv preprint arXiv:2502.07393*, 2025.
- Shao Z, Wang P, Zhu Q, Xu R, Song J, Bi X, Zhang H, Zhang M, Li YK, Wu Y, Guo D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models *arXiv preprint arXiv:2402.03300*, 2024.
- Moody J, Wu L, Liao Y, Saffell M. 1998. Performance functions and reinforcement learning for trading systems and portfolios *Journal of Forecasting* 17(5–6):441–470.
- Neuneier R. 1995. Optimal asset allocation using adaptive dynamic programming *Advances in Neural Information Processing Systems* 8.
- Poterba JM, Summers LH. 1988. Mean reversion in stock prices: Evidence and implications *Journal of Financial Economics* 22(1):27–59.
- Bollen J, Mao H, Zeng X. 2011. Twitter mood predicts the stock market *Journal of Computational Science* 2(1):1–8.
- Loughran T, McDonald B. 2011. When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks *The Journal of Finance* 66(1):35–65.
- Huang AH, Wang H, Yang Y. 2023. FinBERT: A large language model for extracting information from financial text *Contemporary Accounting Research* 40(2):806–841.
- Wu S, Irsoy O, Lu S, Dabrowski V, Dredze M, Gehrmann S, Kambadur P, Rosenberg D, Mann G. 2023. Bloomberggpt: A large language model for finance *arXiv preprint arXiv:2303.17564*
- Graves A. 2012. Long short-term memory *Supervised sequence labelling with recurrent neural networks*, 2012: 37–45.
- Vaswani A, et al. 2017. Attention is all you need *Advances in Neural Information Processing Systems*, 2017, 30.
- Loshchilov I, Hutter F. 2017. Decoupled weight decay regularization *arXiv preprint arXiv:1711.05101*, 2017.
- Bergstra J, et al. 2011. Algorithms for hyper-parameter optimization *Advances in Neural Information Processing Systems*, 2011, 24.
- Dong Z, Fan X, Peng Z. 2024. Fnspid: A comprehensive financial news dataset in time series *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*:4918–4927.
- Comanici G, Bieber E, Schaeckermann M, Pasupat I, Sachdeva N, Dhillon I, Blistein M, Ram O, Zhang D, Rosen E, others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities *arXiv preprint arXiv:2507.06261*

## A Supplementary for hyper-parameters tuning

This appendix section provides the supplementary materials for subsection 3 in Section 3. In this study, we tune the hyper-parameters using optuna with TPE (Tree-structured Parzen Estimator) algorithm (Bergstra et al. 2011). Table 3 shows the tuned hyper-parameters for RL structure with different algorithms and network architectures and their corresponding value ranges that we used in this paper.

Table 3: Hyper-parameters and tuning ranges/values.

Description	Range/Values
Window size for sequence models	10 to 50 (log scale)
Hidden dimension size for LSTM	[32, 64, 128]
Number of layers for LSTM/Transformer	[1, 2] (LSTM), 1 to 3 (Transformer)
Position encoder standard deviation in Transformer	0.02 to 1 (log scale)
Number of attention heads in Transformer	[2, 4]
Feedforward dimension in Transformer	[32, 64, 128]
First hidden layer size in MLP	[32, 64, 128]
Second hidden layer size in MLP	[32, 64, 128]
Discount factor for future rewards	0.90 to 0.995 (log scale)
Gradient clipping norm	0.1 to 4.0 (log scale)
State value target update rate	[0, 0.01]
Learning rate for optimizer	2e-6 to 1e-3 (log scale)
Weight decay coefficient	1e-5 to 1e-2 (log scale)
Training batch size	[32, 128, 512]
Policy update repetition times	[1, 2] (off-policy), [4, 8] (on-policy)
Horizon length for training	max_step × [2, 4, 8]
Replay buffer size	horizon_len × [2, 4, 8]
Exploration rate for DDQN	0.005 to 0.125 (log scale)
Epsilon decay rate for DDQN	[0.99995, 0.99999, 0.999999]
Soft update rate for target network	1e-3 to 1e-2 (log scale)
GAE parameter for PPO	0.9 to 0.99
Clipping ratio for PPO	0.1 to 0.2
Target KL divergence for PPO	0.005 to 0.02 (log scale)
Entropy coefficient for PPO	0.001 to 0.1 (log scale)
Value function coefficient for PPO	0.1 to 1.0 (log scale)