Adaptive Confidence-Weighted LLM Infusion for Financial Reinforcement Learning

Emran Y. Alturki
Imperial College London
e.alturki24@imperial.ac.uk

Aydin Javadov
ETH Zürich
ajavadov@ethz.ch

Qiyang Sun
Imperial College London
q.sun23@imperial.ac.uk

Björn W. Schuller *Fellow, IEEE*
Imperial College London
Technical University of Munich
schuller@tum.de

***Abstract*—We propose an enhanced version of FinRL-DeepSeek for the "FinRL Contest 2025 Task 1", a framework that integrates Large Language Model (LLM) signals into Reinforcement Learning (RL) for stock trading. The original approach employed fixed LLM-based perturbations (e. g., on actions or trajectory returns), overlooking each signal's uncertainty and thereby risking the amplification of noise.**

To address these limitations, we introduce *Adaptive Confidence-Weighted LLM Infusion* for two types of infusions. First, trading-actions scores, which aligns each LLM-derived factor around a neutral baseline based on the LLM's self-reported confidence. High-confidence signals preserve their strength, while low-confidence signals are dampened. Second, controlling the exploration-exploitation trade-off by entropy regularisation.This dynamic adaptation is implemented in both PPO and CVaR-PPO, enabling more robust, context-aware policy updates. Empirical backtests on real market data demonstrate that our method consistently improves stability and risk-adjusted returns, underscoring the value of explicitly modeling confidence in LLM-augmented RL trading strategies. [1] [2]

***Index Terms*—FinRL, Reinforcement Learning, Prompting, LLM Infusion, Financial Trading, Algorithmic Portfolio Management**

I. Introduction

Algorithmic trading is a rapidly evolving domain, and Reinforcement Learning (RL) has emerged as a powerful approach within it. However, as highlighted by [1] [2], there remains a notable gap in effectively integrating text-based market information (e. g., financial news) and addressing risk management in RL-driven trading strategies. This paper, produced under the guidelines of the FinRL Contest 2025, serves as a complementary extension of the aforementioned work, with an emphasis on refining prompt design, LLM infusion, and hyperparameter tunings. By doing so, we aim to enhance the robustness and performance of RL-based trading agents, ultimately laying a stronger foundation for future developments in algorithmic trading research.

II. Data and Methodology

In this section, dataset, LLM prompts, and algorithms will be discussed. The new LLM-infusion method will be discussed in a separate section.

[1]https://github.com/emranyat/FinRL_DeepSeek
[2]https://huggingface.co/datasets/emranyat/FinRL_2025

A. Datasets

Main datasets include Yahoo Finance for the Nasdaq-100 stocks information (e. g., prices, volumes, indicators, etc) [3]. Additionally, the FINSPID dataset [4] which was published last year and contains over 15.7 Million news on the Nasdaq-100 stocks, is used. This dataset was shortened randomly by [2] to 2 Million news for the purpose of LLM requests expenses.

B. LLM Prompts

- **Prompting Method:** According to [5], effective prompting for Large Language Models (LLMs) often involves a combination of strategies, such as *probscore*, *advanced descriptions*, and *few-shot learning*.
  1) *Probscore* gauges how certain the LLM is about its own response.
  2) *Advanced descriptions* ensure thorough background context for the LLM, potentially improving answer quality.
  3) *Few-shot learning* supplies a small number of examples (e. g., five), guiding the LLM's outputs via *In-Context Learning (ICL)*.
  4) *Date Specification* may influence an LLM to disregard knowledge acquired after the date a news item was published, reducing the risk of look-ahead bias on the score's evaluation.

  Based on [6], properly engineered prompts can outperform naive or purely human-crafted queries. Consequently, we utilize a highly ranked LLM (GROK-3 [7]) [8] to generate our final prompt (see Figure 1 for the risk score example), then feed that prompt into another LLM (DeepSeek-V3 [9]) to obtain the final scores. This same approach also applies to sentiment scoring, using a prompt closely resembling Figure 1.
- **Confidence Probabilities:** As described above, we request the LLM to output a confidence probability (or integer rating) alongside each score, recognizing that not all news items carry equal significance. Leveraging this confidence factor in our RL training (see Section III) helps the agent weigh uncertain signals more cautiously and trust highly confident signals more strongly.

979-8-3315-9661-3/25/$31.00 ©2025 IEEE
DOI 10.1109/IDS66066.2025.00025

Fig. 1. Prompt for Risk Scores and Confidence Probabilities

C. Algorithms

We retain the two primary algorithms from the original paper [2], but enhance their integration with LLM signals:

a) Proximal Policy Optimization (PPO).: Originally introduced by OpenAI [10], PPO updates its policy by clipping the probability ratios between the new and old policies. This clipping mechanism penalizes excessive deviations, preventing unstable policy swings, and is typically yielding more robust performance in practice.

b) Conditional Value at Risk-Proximal Policy Optimization (CVaR-PPO): Proposed by [11], CVaR-PPO extends PPO by incorporating *risk-sensitive* optimization through Conditional Value at Risk (CVaR). This additional risk constraint helps mitigate large losses over adverse market conditions, often producing safer trading strategies than vanilla PPO. In our context, CVaR-PPO also benefits from the LLM-based signals, allowing us to more effectively manage and adapt to shifting risk profiles in the market.

III. Adaptive Confidence-Weighted LLM Infusion

In the original version of FinRL-DeepSeek [2], the LLM signals were injected into two main components of the RL agent:

1) A *recommendation factor* S_f, which amplifies or dampens each action a_t (e. g., pushing the policy to buy or sell).
2) A *risk factor* R_f, which modifies the trajectory returns $D(\pi_\theta)$ for risk-sensitive optimization (e. g., in CVaR-PPO).

A. Adaptive Confidence-Weighted Infusion

To incorporate the LLM's confidence explicitly, we propose a novel mechanism where each discrete LLM score (1–5) is accompanied by a confidence probability $C \in [0.00, 1.00]$ (where 0.00 indicates no confidence and 1.00 indicates a certain confidence in the score). We then replace the fixed factors with:

a) Adaptive Recommendation Factor: We define

$$\begin{aligned} S_f^{\text{mod}} &= 1 + (S_f - 1) * C_{\text{rec}} \cdot \alpha \\ a_t^{\text{mod}} &= S_f^{\text{mod}} \cdot a_t \,, \end{aligned} \tag{1}$$

where S_f is the *original* baseline factor from the LLM (e. g., $0.9, 0.95, 1.0, 1.05, 1.1$), and C_{rec} is the normalized confidence (ranging 0.00–1.00). The parameter $\alpha > 0$ is a hyperparameter controlling how strongly we amplify or dampen the recommendation based on confidence.

b) Adaptive Risk Factor: Similarly, for the risk component R_f^i, we introduce:

$$\begin{aligned} (R_f^i)^{\text{mod}} &= 1 + \left(R_f^i - 1\right) * C_{\text{risk}}^i \cdot \beta \\ R_f^{\text{mod}} &= \sum_i w_i \cdot \left(R_f^i\right)^{\text{mod}}, \end{aligned} \tag{2}$$

where each R_f^i is the baseline around 1.0, β is a tunable scaling parameter, and $C_{\text{risk}}^i \in [0.00, 1.00]$ is the confidence for each stock's risk assessment. The modified factor R_f^{mod} then replaces (**??**) in the CVaR or PPO objective:

$$D_{R_f^{\text{mod}}}(\pi_\theta) \;=\; R_f^{\text{mod}} \cdot D(\pi_\theta). \tag{3}$$

B. Entropy Regularisation

This is a different LLM infusion method, where the entropy function is injected with the LLM scores. The purpose of this method is to control the trade-off between exploration-exploitation of the policy, where if a stock got a high recommendation score, the policy will exploit this option and opposite if the score is low, and vice versa for risk scores.

a) Recommendation Scores: The policy loss in PPO with entropy regularization is given by:

$$\begin{aligned} \mathcal{L}_\pi = -\,\mathbb{E}_t \Bigg[\min \Bigg(& \frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} A_t, \\ & \text{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}, 1-\epsilon, 1+\epsilon \right) A_t \Bigg) \Bigg] \\ & + \beta \cdot \mathcal{H}(\pi_\theta(\cdot|s)) \end{aligned} \tag{4}$$

where:

- $\mathcal{H}(\pi_\theta(\cdot|s)) = -\sum_a \pi_\theta(a|s) \log \pi_\theta(a|s)$ is the **entropy**.
- β is the entropy coefficient (hyperparameter).

where the function of β is one of two options:

$$\beta = coef \cdot (5 - S_{\text{LLM}}) \cdot C_{\text{rec}} \tag{5}$$

Or;

$$\beta = S_{\text{LLM}} \cdot C_{\text{rec}}^2 \tag{6}$$

where S_{LLM} is the recommendation score that was assigned by the LLM (1-5).

b) Risk Scores: In the case of risk scores in CPPO:

$$\beta = coef \cdot R_{\text{LLM}} \cdot C_{\text{risk}} \tag{7}$$

Or;

$$\beta = R_{\text{LLM}} \cdot C_{\text{risk}}^2 \tag{8}$$

C. Adaptive Factors + Entropy Regularization

We propose the merging of both LLM infusion methods, where injection will occur at the policy level to amplify / damage the action and at the entropy level to mitigate the trade-off between exploration and exploitation.

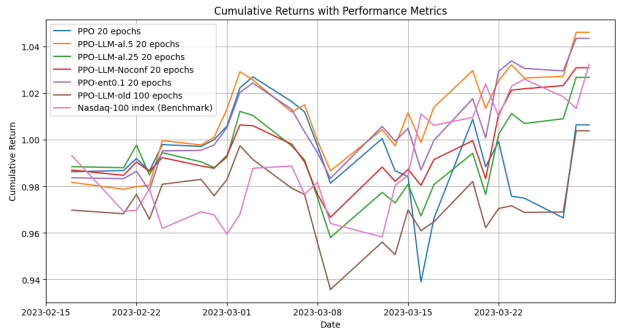

Fig. 2. Backtesting of PPO after 400K training (20 epochs, 20k steps), 14 months training, 2 months trading

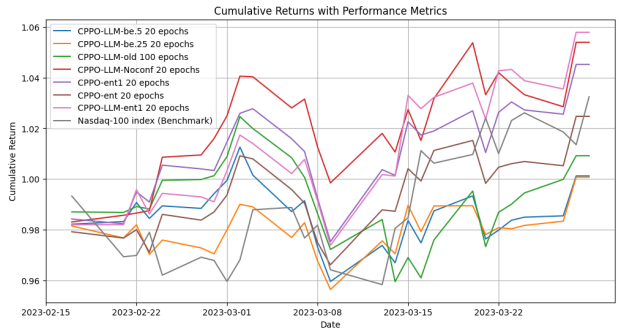

Fig. 3. Backtesting of CPPO after 400K training (20 epochs, 20k steps), 14 months training, 2 months trading

IV. Results

A. Early Stopping

Experiments were conducted to a small subset of the whole dataset (2013-2023) to reduce computational costs and explore different pathways. Specifically, we tried a 10% amplifier/damper of the score for both PPO and CPPO.

Training Period: 2022 - 2023

Trading Period: 2023/2 - 2023/3

Figure 2,3 and table I present the preliminary results of early stopping (400k training steps). Results indicate that the use of CPPO with both LLm infusions (factors and entropy) got a higher cumulative returns than the Nasdaq-100 benchmark, indicating that the infusion of recommendation/risk scores and its confidences have positively contributed to the model.

B. Full Training

Owing to limited computational resources, we have postponed the full-scale experiments to a later stage. Specifically, we plan to run the model on the entire dataset with a more extensive training schedule (e. g., 100 epochs and 20k steps). These results, along with the final trained model, will be submitted once the extended experiments are completed.

V. Conclusion

We introduced an enhanced version of FinRL-DeepSeek by refining its LLM prompt design and incorporating an adaptive LLM-infusion approach. Empirical evidence suggests our improvements can significantly stabilize performance and better leverage text-based financial information. In future work, we aim to:

- **Expand News Sources:** Incorporate diverse media (e. g., X-posts, YouTube videos) to capture a broader market sentiment spectrum and enable more robust decision-making.
- **Pipeline Optimization:** Streamline the data processing and request management pipeline for LLM interactions. In our initial trials, parallelizing the requests to the LLM provider reduced average latency by up to 90%, minimizing dropped queries and missing data.

TABLE I
Backtesting Results

Model	Cumulative Returns	Rachev Ratio	Max Drawdown	Outperf. Frequency	Downturn Outperf.
PPO	0.0065	1.2292	-0.0610	48.1481	83.3333
PPO-LLM al0.5	0.0462	1.1154	-0.0212	55.5556	91.6667
PPO-LLM al0.25	0.0269	1.3214	-0.0419	51.8519	83.3333
PPO-LLM Noconf	0.0310	1.6571	-0.0333	55.5556	91.6667
PPO-ent0.1	0.0436	1.4952	-0.0217	55.5556	91.6667
CPPO-LLM be0.5	0.0012	0.9164	-0.0405	55.5556	91.6667
CPPO-LLM be0.25	0.0007	1.1748	-0.0437	48.1481	83.3333
CPPO-LLM Noconf	0.0539	1.2210	-0.0170	55.5556	91.6667
CPPO-ent0.1	0.0452	1.4174	-0.0248	55.5556	91.6667
CPPO-ent	0.0247	1.1145	-0.0339	55.5556	91.6667
CPPO-LLM & ent0.1	0.0579	1.6859	-0.0262	51.8519	83.3333

References

[1] K. Wang, K. Xiao, and X.-Y. L. Yanglet, "Parallel market environments for finrl contests," *arXiv preprint arXiv:2504.02281*, 2025.

[2] M. Benhenda, "FinRL-DeepSeek: LLM-infused risk-sensitive reinforcement learning for trading agents." [Online]. Available: http://arxiv.org/abs/2502.07393

[3] Yahoo Finance. Stock market data, financial news, and analysis. Accessed: 2025-04-01. [Online]. Available: https://finance.yahoo.com

[4] Z. Dong, X. Fan, and Z. Peng, "FNSPID: A comprehensive financial news dataset in time series." [Online]. Available: http://arxiv.org/abs/2402.06698

[5] D. Yang, Y.-H. H. Tsai, and M. Yamada, "On verbalized confidence scores for LLMs." [Online]. Available: http://arxiv.org/abs/2412.14737

[6] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba, "Large language models are human-level prompt engineers," 2023. [Online]. Available: https://arxiv.org/abs/2211.01910

[7] xAI, "Grok-3," 2024, accessed: 2025-04-01. [Online]. Available: https://x.ai

[8] W.-L. Chiang, L. Zheng, Y. Sheng, A. N. Angelopoulos, T. Li, D. Li, H. Zhang, B. Zhu, M. Jordan, J. E. Gonzalez, and I. Stoica, "Chatbot arena: An open platform for evaluating llms by human preference," 2024.

[9] DeepSeek-AI, "Deepseek-v3 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2412.19437

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[11] C. Ying, X. Zhou, H. Su, D. Yan, N. Chen, and J. Zhu, "Towards safe reinforcement learning via constraining conditional value-at-risk," in *International Joint Conference on Artificial Intelligence*, 2022. [Online]. Available: https://arxiv.org/abs/2206.04436