Sam's Fans at the Crypto Trading Challenge Task: A Threshold-Based Decision Approach Based on FinMem Framework

You Wang*, Jingyi Wei* and Mingsong Ye*
*Equal Contribution
Stevens Institute of Technology, Hoboken, NJ, United States
{ywang408, jwei14, mye2}@stevens.edu

Abstract

The advancements of large language models (LLMs) demonstrate the value of pre-training on diverse datasets, enabling these models to excel across a wide range of tasks while adapting effectively to specialized applications. This study presents an approach to enhance LLMs' ability to process and trade based on cryptocurrency data across different time horizons. We fine-tuned two established language models, Llama-3.1-8b and Qwen2.5-7b, to effectively interpret and utilize temporal market data provided by the FinMem framework. Our methodology enables these models to analyze multi-period market data from FinMem, including price movements and momentum indicators, to execute effective cryptocurrency trading decisions. Results show that this fine-tuning approach improves the models' capacity to analyze market conditions and inform trading decisions based on multi-period market dynamics.

1 Introduction

Cryptocurrency trading markets are among the most complex and fast-paced environments in the financial world. These markets exhibit extreme volatility and are influenced by a broad range of data sources, including real-time price changes, breaking news, regulatory updates, social media sentiment, and macroeconomic indicators (FinNLP Workshop@COLING25, 2024; Fang et al., 2022). Successfully extracting actionable insights from this diverse and time-sensitive information requires sophisticated systems that can process multi-temporal data while addressing uncertainty and rapid market shifts.

Large Language Models (LLMs) (Bubeck et al., 2023; Li et al., 2023) have emerged as powerful tools for processing unstructured data, offering advanced capabilities in reasoning, natural language understanding, and decision-making. Models like Qwen2.5-7B (Hugging Face, 2024) and Llama-3.1-8B (Meta AI, 2024) have been proven effective in various financial applications, such as sentiment analysis, market text summarization, and asset price prediction (Li et al., 2023). Furthermore, the fine-tuning (Zaken et al., 2021; Hu et al., 2021) techniques can enhance LLMs' ability to handle data and tasks in specific domains.

In this study, we proposed a fine-tuning strategy to enhance LLMs' performance in automating the currency trading, combined with FinMem framework. We first curated the data from diverse areas and implemented LORA fine-tuning techniques to enhance LLMs' understanding of the complex cryptocurrency trading environment. And then supervised LLMs for the desired actions also through LoRA. We tested our approach with two standard LLMs, Llama-3.1-8B and Qwen2.5-7B, the results show the potential of LLMs' advance in the automated trading tasks. Our solution ranks as the top trading agent in the Cryptocurrency Trading Challenge competition (FinNLP Workshop@COLING25, 2024).

Our extensive experiments demonstrate that our approach is effective and partially meets the objectives underscored by this competition. First, LLMs shows different behaviors after being fine-tuned with knowledge base, suggesting a potential that LLMs understand cryptocurrency trading's unique complexity. Second, our final solution agents achieve a robust higher return in BTC trading than baseline agent. However, those final fine-tuned agents did not demonstrate significant improvement in ETH trading. We believe that this is caused by the naivety of the strategy we implemented to supervise LLMs' trading actions.

1.1 Competition Background

This study was initiated to address the Cryptocurrency Trading Challenge at FinNLP @ COLING25, where a trading agent is required to integrate within FinMem Framework (Yu et al., 2024). FinMem is

Proceedings of the Joint Workshop of the 9th FinNLP, the 6th FNP, and the 1st LLMFinLegal, pages 407–413
January 19–20, 2025.

a versatile platform designed for financial decision-making, leveraging LLMs as core components to integrate multi-source information and facilitate sequential decision-making. Specifically, from FinMem, the required agent will receive a comprehensive coverage related to the asset of interest and then react with 'buy, hold, sell' decisions.

This competition specifically highlights three objectives for the ultimate evaluation of LLM agents' performance:

1. knowing the unique complexity of cryptocurrency market
2. extracting effective information from data of various sources
3. delivering robust trading returns regarding multi-turn actions

1.2 Related Works

This study is related to research of two disciplines: automated trading systems, as discussed by (Huang et al., 2019) and large language model agents, as explored by (Xi et al., 2023)

The automated trading system traditionally relies on technical analysis (Lev and Thiagarajan, 1993), focusing on identifying short-term trading dynamics through statistical models. However, with the recent integration of machine learning (ML) techniques for contextual data analysis, fundamental analysis (Lo et al., 2000)—which assesses the long-term intrinsic value of an asset—has also been incorporated into the automated trading system.

Automated trading can be mathematically modeled using stochastic programming (Shapiro et al., 2021), typically addressed through approximate dynamic programming and reinforcement learning techniques (Sutton and Barto, 2018). Yang et al. (2020) conducted experiments with deep reinforcement learning to develop an ensemble trading strategy. Their findings indicated superior performance over three individual algorithms and two baseline models in terms of risk-adjusted returns, as quantified by the Sharpe ratio.

Machine learning (ML) has become extensively utilized in the field of financial technology, FinTech, for purposes of analysis and forecasting. For instance, natural language processing enables the extraction of semantics and dependencies from textual data. Additionally, advanced non-linear machine learning models are employed to identify behavioral patterns.

Recent advancements in LLMs have been readily incorporated into FinTech innovations. For instance, Bloomberg has developed a finance-specific LLM, BloobergGPT (Wu et al., 2023), which surpasses existing LLMs in financial tasks while maintaining robust performance across standard LLM benchmarks.

One method to enhance the performance of LLM agents is through prompt engineering. Although LLMs are renowned for their remarkable zero-shot learning capabilities (Kojima et al., 2022) and in-context learning (Brown, 2020), Prompt engineering enables the decomposition of a task into multiple parts, making the LLM appear more intelligent by facilitating a more manageable, step-by-step approach to problem-solving. For example, the chain-of-thought prompting (Wei et al., 2022) technique is commonly utilized to aid LLMs in reasoning through complex tasks, such as solving multi-step mathematical problems or processing intricate natural language queries.

The other method to enhance the performance of LLMs in specific domains involves fine-tuning based on established models such as ChatGPT (OpenAI) and Llama (Meta AI, 2024). Parameter-efficient fine-tuning techniques, such as the Low-Rank Adapter (Hu et al., 2021), are widely used due to the computational intensity of LLM training. In LoRA, a trainable auxiliary matrix is introduced to the pre-trained transformer model (Vaswani, 2017). This matrix is reparametrized using low-rank decomposition, significantly reducing the number of parameters required.

2 Methodology

In this section, we propose a fine-tuning strategy to enhance LLMs for cryptocurrency trading tasks. Our approach includes two steps which are shown in Figure 1: the first step is to enhance the LLMs' understanding of cryptocurrency trading environments through a knowledge dataset consisting of domain-specific questions and answers; the second step is to supervise the LLMs' trading actions.

2.1 Base Knowledge Integration

In the first phase, we focused on addressing the LLMs's limited understanding of cryptocurrency markets by curating comprehensive datasets consisting of domain-specific questions and answers. The dataset are question-answer pairs, which covers foundational principles, market dynamics, and

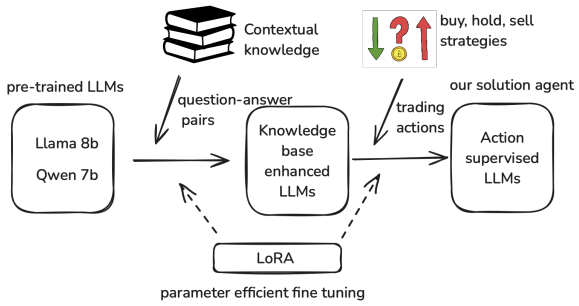


Figure 1: The two-stage training diagram

key concepts in blockchain and cryptocurrency. By learning from this targeted material, the models acquired a robust base of specialized knowledge. This foundational training ensures that the LLMs develop the contextual understanding necessary for practical application in cryptocurrency trading.

2.2 Threshold-Based Decision-Making

In the second phase, we use the FinMem framework to generate inputs for the dataset by organizing financial data into short, mid, and long-term memory layers, offering insights into price changes and momentum indicators. FinMem also captures key insights like price changes and momentum indicators across different time frames, ensuring critical information is readily accessible. Using FinMem-generated data for both training and testing ensures consistency, enabling the LLMs to process multi-source information effectively and enhancing their ability to develop reliable trading strategies in the dynamic cryptocurrency market.

To create labels for model training, we use a threshold-based decision-making method to generate actionable signals: "buy," "sell," or "hold." These labels are based on predicted returns. A "buy" label is assigned when predicted returns exceed 1%, indicating a strong opportunity to invest. A "sell" label is triggered if predicted returns fall below -1%, signaling a likely loss. Predicted returns between -1% and 1% result in a "hold" label, minimizing unnecessary trades in marginal conditions. This approach ensures the dataset provides clear, practical targets, aligning model predictions with real-world trading strategies.

3 Experiment and Analysis

3.1 Experiment Setup

In this section, we present the experiments of fine-tuning LLMs using our proposed approach regarding the cryptocurrency trading task. The experiments were conducted on a virtual machine with a single Nvidia H100 GPU, which had a limited GPU memory of 30-40GB. Given the computational constraints of working with large models, we implemented several optimization techniques to ensure efficient training while maintaining model performance.

Our approach primarily relies on Parameter-Efficient Fine-Tuning (PEFT) (Houlsby et al., 2019), a framework that enables model adaptation by modifying only a small subset of parameters. Among PEFT's various techniques, including prompt tuning and adapter methods, we selected Low-Rank Adaptation (LoRA) (Hu et al., 2021) for its proven effectiveness in preserving model performance while minimizing additional parameters through low-rank decomposition.

To optimize memory usage and training efficiency, we implemented several technical enhancements. We employed mixed precision training with bfloat16 utilizing Flash Attention 2 (Dao, 2023), and further reduced memory consumption by using 4-bit int quantization in loading models, improving upon the default 8-bit int quantization in Hugging Face. The LoRA configuration includes a LoRA-α value of 8, rank of 5, and dropout of 0.1, targeting key projection matrices (query, key, value, and input layers).

We conducted experiments using two models: Llama-3.1-8B and Qwen2.5-7B, training and testing them on datasets described in Sections 2.1 and 2.2. The implementation leverages PEFT and Quantization libraries from Huggingface. To evaluate model performance, we employed multiple metrics including semantic similarity, cumulative returns, and Sharpe ratio, with a buy-and-hold strategy serving as the baseline.

Our experimental design included two key investigations. First, we examined the importance of base knowledge integration by comparing models with and without this foundation, visualizing the differences through cumulative returns from backtesting. Second, we evaluated the impact of threshold-based decision training on models with integrated base knowledge. Performance metrics, including cumulative returns and Sharpe ratios, were calculated through backtesting and compared against both the buy-and-hold baseline and models without threshold-based decision training. This comprehensive evaluation framework allowed us to assess the individual contributions of base knowl-

edge and threshold-based decision training to overall model performance.

3.2 Evaluation Results and Analysis

3.2.1 Base Knowledge Impact

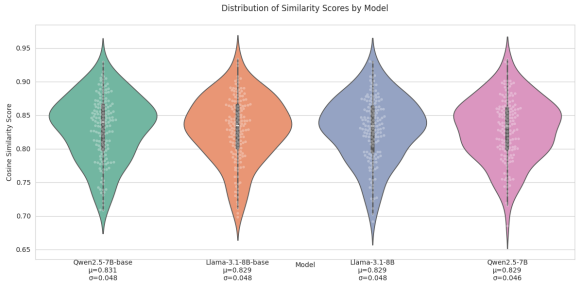


Figure 2: Similarity Distribution of Models with/without Base Knowledge

To illustrate the necessity of incorporating base knowledge, we categorized the LLMs into two groups: (1) models pre-trained using a specialized question-and-answer dataset to integrate base knowledge ("with base"), and (2) the original LLM models without this additional pre-training ("without base"). We evaluated the impact of training on the base knowledge dataset by comparing the semantic similarity between answers generated by the models and the corresponding answers in a test dataset. For this analysis, OpenAI's text-embedding-ada-002 model was employed to generate embeddings for both sets of texts, followed by calculating their cosine similarity. The distribution of similarity scores for both groups was then analyzed and visualized, as depicted in Figure 2.

As shown in the violin plot, both the original models, Llama-3.1-8B and Qwen2.5-7B, achieved an average semantic similarity of 85%. However, the models trained with the base knowledge dataset demonstrated no significant improvement in either the mean similarity or the variance. To explore the practical implications of base knowledge integration, we further investigated whether models with base knowledge could achieve better performance in trading cryptocurrencies.

We evaluated the cumulative returns of LLMs with and without base knowledge to compare their trading performance, as shown in Figure 3.

In BTC trading, integrating base knowledge does not improve performance. Both Llama-3.1-8B and Qwen2.5-7B with base knowledge show only minor differences in returns and Sharpe ratios compared to their original versions and the baseline. This suggests limited value for base knowledge in this context. For ETH trading, the results are mixed. Llama-3.1-8B with base knowledge achieves higher returns and a better Sharpe ratio than its untrained counterpart but underperforms the baseline. Conversely, Qwen2.5-7B with base knowledge performs worse, showing negative returns and a poor Sharpe ratio, while its untrained version stabilizes near zero returns, outperforming the trained model but still falling short of the baseline. These findings highlight that while base knowledge alters model behavior, it fails to enhance performance, likely due to a mismatch with real-time market dynamics.

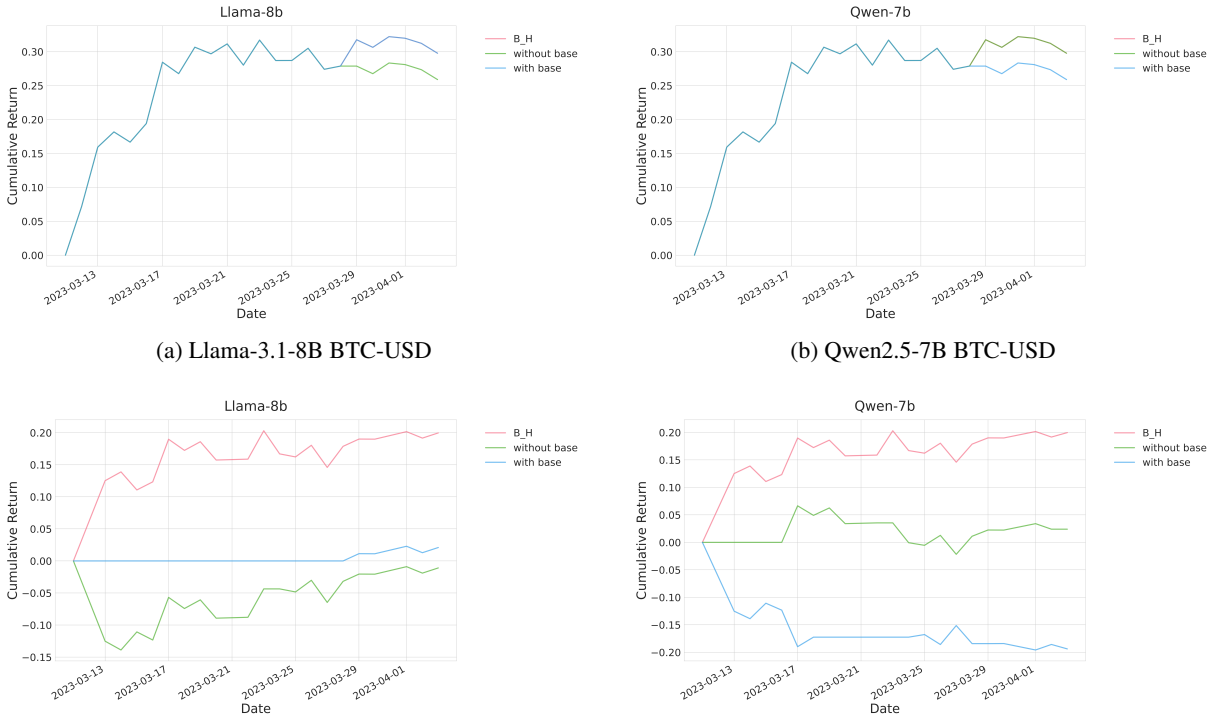
As a result, in the next subsection, we introduce our second dataset to address these challenges. The inputs consist of processed information, including short-, mid-, and long-term market memory, momentum indicators, and price changes. Labels are derived using a threshold-based decision-making process. This approach aims to align static knowledge with dynamic market data, bridging the gap between pre-trained knowledge and real-time trading conditions.

3.2.2 Final Model Evaluation

We finalized the Llama-3.1-8B and Qwen2.5-7B models by integrating base knowledge and training them using a threshold-based decision strategy. During backtesting, we compared cumulative returns and Sharpe ratios across three scenarios: the Buy and Hold (B_H) baseline, models with base knowledge but no threshold training, and the finalized models. The cumulative returns across scenarios are shown in Figure 4.

In BTC trading, the Qwen2.5-7B final model outperformed both the baseline and the model with base knowledge. And for Llama-3.1-8B in BTC trading, the baseline slightly outperformed the final model in cumulative returns. This highlights how integrating base knowledge and training on threshold-based decisions led to better cumulative returns and Sharpe ratios, enabling more effective decision-making and adaptability to BTC market conditions.

In ETH trading, the baseline buy-and-hold strategy consistently outperformed all models. While the Qwen2.5-7B final model slightly outperformed its base knowledge-only counterpart, neither Llama-3.1-8B nor Qwen2.5-7B achieved positive cumulative returns or Sharpe ratios. A potential explanation, based on checking the Fin-



(a) Llama-3.1-8B BTC-USD

(b) Qwen2.5-7B BTC-USD

(c) Llama-3.1-8B ETH-USD

(d) Qwen2.5-7B ETH-USD

Figure 3: Base Knowledge Impact based on Cumulative Returns Comparison

	Llama-3.1-8B - BTC			Qwen2.5-7B - BTC			Llama-3.1-8B - ETH			Qwen2.5-7B - ETH		
	B_H	BK	no BK	B_H	BK	no BK	B_H	BK	no BK	B_H	BK	no BK
CR ↑	0.298	0.259	0.298	0.298	0.259	0.298	0.200	0.021	-0.011	0.200	-0.194	0.024
SR ↑	4.633	4.071	4.633	4.633	4.071	4.633	3.336	2.829	-0.180	3.336	0.657	-3.532

'CR': cumulative return, 'SR': Sharpe Ratio. 'B_H': 'buy and hold'. 'BK': model with base knowledge, 'no BK': model without base knowledge. ' ↑' indicates the higher the better.

Table 1: Base Knowledge Impact based on Performance Metrics

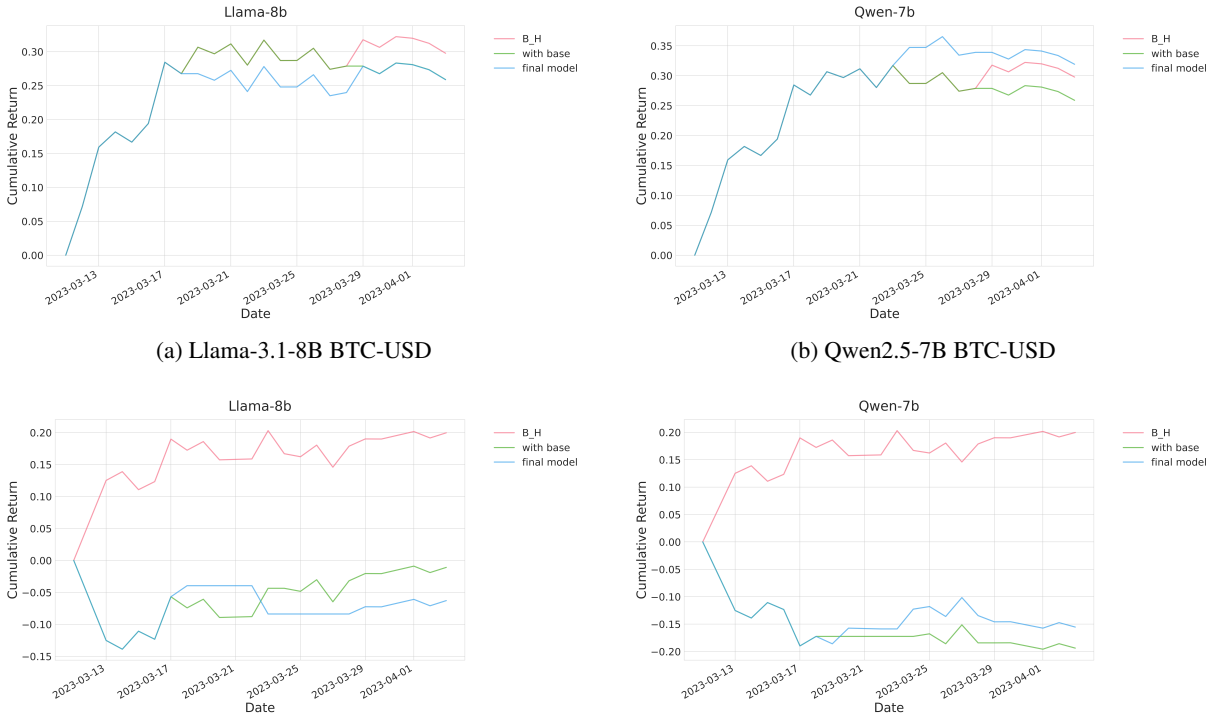
Mem processed data and ETH price trends, is the lag between news inputs and ETH price movements, which may hinder the models' ability to effectively align static knowledge with the dynamic and rapidly evolving market conditions.

Performance metrics in Table 2 further support these findings. Overall, the Qwen2.5-7B final model excelled in BTC trading, demonstrating the value of combining base knowledge with threshold-based training. However, ETH trading results revealed that while these methods help align static and dynamic information, they require refinement to handle the specific challenges of ETH markets.

4 Conclusion

In this study, we fine-tuned the Llama-3.1-8B and Qwen2.5-7B models, combined with the FinMem framework to address the challenges of automated cryptocurrency trading with Bitcoin and Ethereum data. Our approach integrated domain-specific knowledge and implemented a threshold-based decision-making framework to handle the volatility and complexity of cryptocurrency markets. Despite these efforts, the models did not outperform the baseline "Buy and Hold" strategy in the ETH market, highlighting areas for improvement in our methodology.

Several factors could explain these results. First, the relatively small size of the 8B and 7B models may limit their inference capabilities, suggesting that larger models with more parameters could better capture complex market patterns. Second, the threshold-based decision framework may require further tuning to adapt to specific market characteristics, such as Ethereum's unique trading behavior. Lastly, the static knowledge dataset itself may lack sufficient granularity or timeliness to align well with real-time market fluctuations.



(a) Llama-3.1-8B BTC-USD

(b) Qwen2.5-7B BTC-USD

(c) Llama-3.1-8B ETH-USD

(d) Qwen2.5-7B ETH-USD

Figure 4: Cumulative Returns Comparison for Finalized Models

	Llama-3.1-8B - BTC			Qwen2.5-7B - BTC			Llama-3.1-8B - ETH			Qwen2.5-7B - ETH		
	B_H	BK	final	B_H	BK	final	B_H	BK	final	B_H	BK	final
CR ↑	0.298	0.259	0.298	0.298	0.259	0.319	0.200	-0.011	-0.063	0.200	-0.194	-0.155
SR ↑	4.633	4.071	4.069	4.633	4.071	5.165	3.336	-0.180	-1.117	3.336	-3.532	-2.655

'CR': cumulative return, 'SR': Sharpe Ratio. 'B_H': 'buy and hold'. 'BK': model with base knowledge, 'final': finalized model with base knowledge trained on threshold-based decisions. ' ↑' indicates the higher the better.

Table 2: Performance Metrics of Finalized Models

Future work should focus on addressing these limitations by exploring larger models, implementing more sophisticated decision strategies, and combining static knowledge with real-time inputs in a more seamless and adaptive way. These refinements could help bridge the gap between static knowledge and dynamic market conditions, enhancing the models' overall performance.

Acknowledgments

We would like to express our gratitude to the organizers of the FinNLP @ COLING25 Cryptocurrency Trading Challenge for providing this platform and the developers of the FinMem framework and pre-trained LLMs for their foundational contributions. We also acknowledge Hanlon Lab of Stevens Institute of Technolgy for computational support and our colleagues for their valuable feedback throughout this project.

References

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. arxiv. *arXiv preprint arXiv:2303.12712*.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Fan Fang, Carmine Ventre, Michail Basios, Leslie Kanthan, David Martinez-Rego, Fan Wu, and Lingbo Li. 2022. Cryptocurrency trading: a comprehensive survey. *Financial Innovation*, 8(1):13.

FinNLP Workshop@COLING25. 2024. Agent-based single cryptocurrency trading challenge. https://coling2025cryptotrading.thefin.ai/. Accessed: 2024-11-23.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Boming Huang, Yuxiang Huan, Li Da Xu, Lirong Zheng, and Zhuo Zou. 2019. Automated trading systems statistical and machine learning methods and hardware implementation: a survey. *Enterprise Information Systems*, 13(1):132–144.

Hugging Face. 2024. Qwen2.5-7b model. https://huggingface.co/Qwen/Qwen2.5-7B. Accessed: 2024-12-02.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Baruch Lev and S Ramu Thiagarajan. 1993. Fundamental information analysis. *Journal of Accounting research*, 31(2):190–215.

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023. Large language models in finance: A survey. In *Proceedings of the fourth ACM international conference on AI in finance*, pages 374–382.

Andrew W Lo, Harry Mamaysky, and Jiang Wang. 2000. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The journal of finance*, 55(4):1705–1765.

Meta AI. 2024. Llama 3.1-8b model. https://huggingface.co/meta-llama/Llama-3.1-8B. Accessed: 2024-12-02.

OpenAI. Chatgpt. Large Language Model.

Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. 2021. *Lectures on stochastic programming: modeling and theory*. SIAM.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Ashish Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

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

Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Denghui Zhang, Rong Liu, Jordan W Suchow, and Khaldoun Khashanah. 2024. Finmem: A performance-enhanced llm trading agent with layered memory and character design. In *Proceedings of the AAAI Symposium Series*, volume 3, pages 595–597.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*.