⊕ *AlphaFin*: Benchmarking Financial Analysis with Retrieval-Augmented Stock-Chain Framework

Xiang Li[1]*, Zhenyu Li[1]*, Chen Shi[2]*,
Yong Xu[1], Qing Du[1]†, Mingkui Tan[1], Jun Huang[2]
[1]South China University of Technology, China
[2]Alibaba Group, China
{lixiangjacky, zhenyuli2148}@gmail.com, deling.sc@alibaba-inc.com

Abstract

The task of financial analysis primarily encompasses two key areas: stock trend prediction and the corresponding financial question answering. Currently, machine learning and deep learning algorithms (ML&DL) have been widely applied for stock trend predictions, leading to significant progress. However, these methods fail to provide reasons for predictions, lacking interpretability and reasoning processes. Also, they can not integrate textual information such as financial news or reports. Meanwhile, large language models (LLMs) have remarkable textual understanding and generation ability. But due to the scarcity of financial training datasets and limited integration with real-time knowledge, LLMs still suffer from hallucinations and are unable to keep up with the latest information. To tackle these challenges, we first release AlphaFin datasets, combining traditional research datasets, real-time financial data, and handwritten chain-of-thought (CoT) data. It has a positive impact on training LLMs for completing financial analysis. We then use AlphaFin datasets to benchmark a state-of-the-art method, called Stock-Chain, for effectively tackling the financial analysis task, which integrates retrieval-augmented generation (RAG) techniques. Extensive experiments are conducted to demonstrate the effectiveness of our framework on financial analysis.

Keywords: Large Language Models, Retrieval-Augmented Generation, Chain-of-Thoughts, Finance, Stock Trend Prediction, Financial Question Answering

1. Introduction

With the advancement of the financial industry, the importance of financial analysis has become increasingly prominent. The ability of financial analysis is primarily manifested in the areas of stock trend prediction and the corresponding financial Q&A. The advent of LLMs has attracted attention from the financial industry, as they possess exceptional generation capabilities (Dredze et al., 2016; Araci, 2019; Bao et al., 2021). Thus, there is a strong desire to leverage these LLMs to enhance the accuracy of financial analysis.[1]

Numerous recent studies have attempted to create efficient algorithms using ML&DL for stock trend prediction (Saad et al., 1998; Shah et al., 2022). At present, ML&DL has been widely used for stock trend prediction based on time series data, generating a positive impact on the industry. However, ML&DL algorithms have limited performance, only able to provide uncertain results and incapable of handling complex textual data. Meanwhile, they fail to provide investors with effective justifications and analyze the underlying causes, potentially dis-
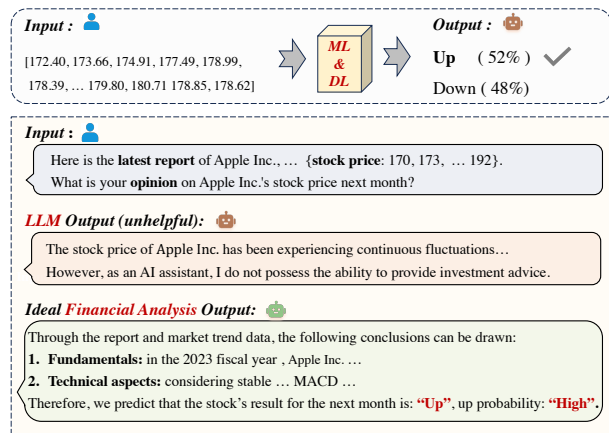

Figure 1: An example of the financial analysis task, including stock trend prediction and financial Q&A. Traditional ML&DL methods merely provide uncertain forecasts (Up/Down) without any justification, while original LLMs could offer analysis of the prediction but unhelpful.

rupting their investment confidence. As shown in Figure 1, for Apple Inc., ML&DL (such as LSTM) can predict an uncertain stock trend ("up") for the next month based on the former stock price data. Nevertheless, it could not provide a reliable result and analysis of the prediction. If they could offer effective analysis, it would greatly enhance investor's confidence in decision-making.

---

* Equal contribution. This work was conducted when Xiang Li and Zhenyu Li were interning at Alibaba.
† Corresponding author.
[1]Resources are publicly available at: https://github.com/AlphaFin-proj/AlphaFin

Fortunately, LLMs possess excellent capabilities in text processing and generation. To utilize the capacity of LLMs, FinGPT (Yang et al., 2023b) and BloombergGPT (Wu et al., 2023) have been specifically designed as FinLLMs. They can be applied to various financial tasks, catering to the needs of the industry. As shown in Figure 1, they have the potential to handle diverse text data, including news and reports (Zhang et al., 2023). This advancement empowers investors to make precise investment and trading decisions.

Nevertheless, building a FinLLMs is not a straightforward task. LLMs often exhibit phenomena such as hallucination and meaningless outputs (Mündler et al., 2023), even for the advanced ChatGPT (Ouyang et al., 2022). As shown in Figure 1, the content generated by general LLMs lacks helpfulness and fails to meet real-time requirements. This can be attributed to two reasons. Firstly, the quality of generated content relies on data availability. The absence of high-quality financial training datasets (Radford et al., 2019) impacts the quality of generation. Secondly, stock trend prediction relies on precise and real-time information, the absence of these information leads to LLMs hallucinations. Despite the widespread application of RAG (Lewis et al., 2021) in other fields, their adaptation in the financial domain remains insufficient.

To tackle these challenges, in this paper, we formalize the task of financial analysis and release AlphaFin for fine-tuning FinLLMs, which contain traditional research datasets, real-time financial data, and handwritten CoT data. Moreover, we propose a Stock-Chain framework integrated with RAG. Stock-Chain not only provides investors with stock trend prediction but also integrates real-time market data and macroeconomic news through RAG, enabling accurate stock analysis during interactions for investors.

Experimental results demonstrate that Stock-Chain is able to achieve the task of stock trend prediction with state-of-the-art accuracy and over 30% annualized rate of return (ARR). Meanwhile, Stock-Chain can provide comprehensive analysis in the financial Q&A, enhancing investors' confidence in decision-making and providing a solid foundation for their investment choices. We conduct extensive supplementary experiments like ablation study, GPT4&human preference evaluation, and case study.

In summary, the contribution lies in four folds:

- We formally define the task of financial analysis, which aims to accomplish stock trend prediction and the corresponding financial Q&A.

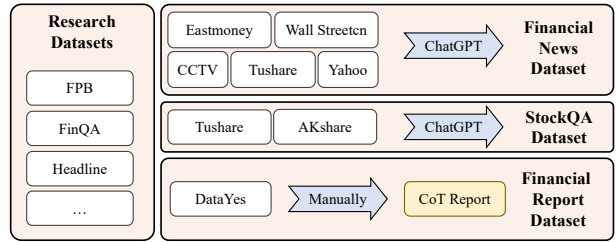- We propose AlphaFin datasets, which contains traditional research datasets, real-time finan-

Figure 2: The data source and preprocessing of the proposed AlphaFin datasets.

cial data, and handwritten CoT data, enhancing LLMs's ability in financial analysis.

- We fine-tune a StockGPT based on AlphaFin datasets and integrate it to a Stock-Chain framework, which is further integrated with a real-time financial database through RAG. By integrating with RAG, we address the issue of the hallucination of LLMs's output and LLMs's inability to generate real-time content.

- We conduct extensive experiments on the AlphaFin datasets, to reveal that Stock-Chain outperforms all the baseline methods, and shows effectiveness for financial analysis.

2. AlphaFin Datasets

We release AlphaFin datasets, as shown in Figure 2, which include four parts, research datasets, StockQA, financial news, and financial reports. AlphaFin is sourced from a dozen of data sources. From Table 1, it is evident that the traditional research dataset exhibits relatively shorter length of label, which hampers the training of FinLLMs. Thus, AlphaFin addresses the issue of low quality and length in traditional research datasets. In this section, we provide the details of its sources and construction process.

2.1. Data Sources

- **Research datasets**: This part includes traditional financial datasets from the academic, including FPB (Malo et al., 2014), FinQA (Maia

Dataset	Size	Input	Label	Type
Research	42,373	712.8	5.6	en
StockQA	21,000	1313.6	40.8	zh
Fin. News	79,000	497.8	64.2	zh
Fin. Reports	120,000	2203.0	17.2	zh
Fin. Reports CoT	200	2184.8	407.8	zh

Table 1: The details of the AlphaFin datasets. "Input" and "Label" denote their text length.

et al., 2018), convFinQA (Chen et al., 2022b) and Headline (Sinha and Khandait, 2020), etc. which enhance the information extraction and summarization ability for LLMs.

- **StockQA dataset**: This part encompasses stock price and other financial data from Tushare (Tushare, 2021) and AKshare (AKshare, 2021). It utilizes sequential data format, such as the real-world stock price trend (e.g. $\{..., 170, 173, 171, 175, 173, 170, ...\}$).

- **Financial News dataset**: To provide real-world financial knowledge for LLMs, we incorporate online news sources, such as the financial sections of CCTV, and Wall Street CN.

- **Financial reports dataset**: We build financial report datasets via DataYes (DataYes, 2021), including professional analysis and knowledge of companies conducted by financial institutions.

2.2. Data Preprocessing

As shown in Figure 2, we explore the details of AlphaFin preprocessing:

- **Research dataset**: Traditional research datasets are primarily in English and of substantial quantity. To enhance the LLMs's ability in Chinese and ensure quality fine-tuning, we only sample a portion from the source.

- **StockQA dataset**: Given the source data is presented in sequential format, we utilize ChatGPT with the following prompt, to generate financial questions upon it. `Based on the ..., give me a good financial question. Input: <sequential data>, Output: <Question>.` This can facilitate training and enhance the diversity of questions. Subsequently, we use ChatGPT to generate responses and obtain Q&A pairs for training LLMs.

- **Financial News dataset**: We leverage ChatGPT to extract a summary for each news, and construct the financial news dataset. This process improves LLMs's ability to generate summaries for financial news.

- **Financial reports dataset**: We manually align the financial reports for the companies and their stock price on the day of report publication, and use the following template to generate the final data. `According to ... give a clear answer up or down. Input: <reports & stock price>, Output: <Up/down>.` Furthermore, we manually create 200 financial reports CoT data with professional financial

knowledge and longer labels, to provide the LLMs with progressive analytical ability. The output format is:

```
According to ...  conclusions can
be drawn: 1.  Fundamentals:  ...
2.  Technical aspects:  ...
Therefore, we predict the ...  is
<up/down>, probability:  <Prob>
```

3. Stock-Chain Framework

We treat the financial analysis task as two counterparts, stock trend prediction and the corresponding financial Q&A. Thus, our proposed Stock-Chain framework is divided into two stages as shown in Figure 3. In this section, we first formalize the task, and then introduce the details of both two stages.

3.1. Problem Definition

For the first stage, given a set of companies $C = \{c_i\}_{i=1}^{N}$ and the corresponding knowledge documents $D = \{d_j\}_{j=1}^{M}$, we can predict stock trends:

$$Pred_i = \phi(c_i, d_j), \quad Pred_i \in \{up, down\} \quad (1)$$

where ϕ represents a stock predicting system, and d_j is retrieved as the related documents for company c_i. The goal is to choose a subset of companies C_{chosen} that are predicted to rise.

$$C_{chosen} = \{c_i | c_i \in C \land Pred_i = up\} \quad (2)$$

For the second stage, we treat a multi-turn dialogue session as a sequence of several query-response pairs between two interlocutors. We denote Q_t and R_t as the user query and agent response at the current time step t, and $H_t = [Q_0, R_0, ..., Q_{t-1}, R_{t-1}]$ as the dialogue history. Then we formalize the financial Q&A task as obtaining the response based on the current query, dialogue history, and corresponding documents:

$$R_t = \pi(d_k, H_t, Q_t) \quad (3)$$

where π represents the conversation system, and d_k is the retrieved document related to Q_t.

3.2. Stage-1: Stock Trend Prediction

As shown in the left part of Figure 3, our first stage is stock trend prediction. Given a company c_i and the corresponding document d_j, this stage maintains a stock predicting system ϕ by combining LLMs and AlphaFin datasets, to give the stock trend prediction $Pred_i$ for c_i.

3.2.1. Knowledge Processing

As shown in Figure 3 ①, given a company c_i, we first retrieve the related documents d_j for it. Then, we design a prompt template $Prompt_1$ as follows:
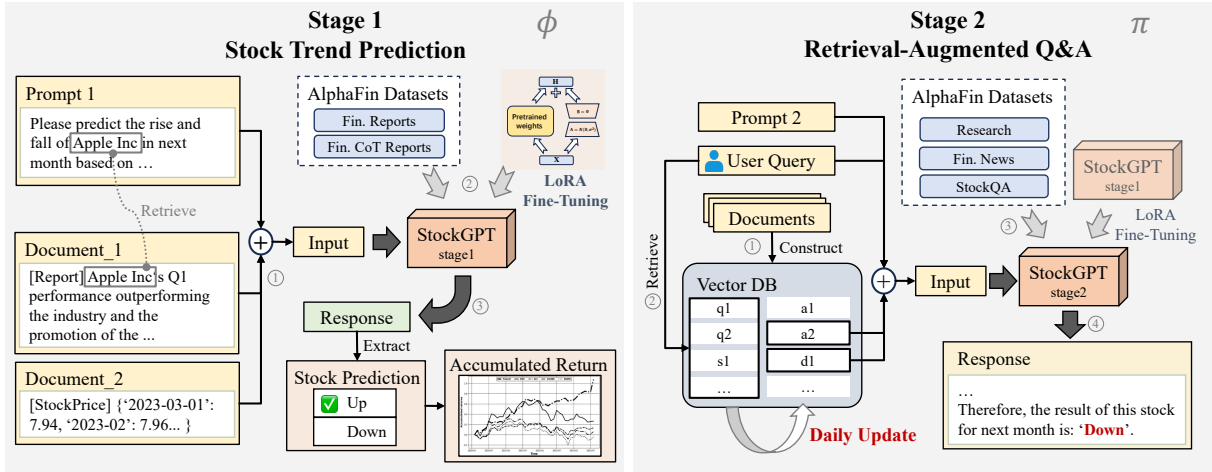
Figure 3: An illustration of the Stock-Chain framework of the two stages in financial analysis.

```
Please predict the rise and fall
of the stock next month based on
the research reports and data pro-
vided below.  Please provide a clear
answer, either "up" or "down".  <re-
port><market data>
```

where `<report>` and `<market data>` compose d_j. Ultimately, we concatenate the prompt with the documents to get the input I_i to the LLMs.

$$I_i = concat(Prompt_1, d_j) \qquad (4)$$

3.2.2. StockGPT Fine-Tuning

As shown in Figure 3 ②, we design the fine-tuning process of an LLM, called StockGPT, which includes two steps. Firstly, we leverage all the financial report datasets of AlphaFin for training. In the second step, we utilize the manually created report CoT dataset to guide the model to think step-by-step. All fine-tuning processes for StockGPT utilize the Low-Rank Adaptation (LoRA) (Hu et al., 2021) method.

Through the fine-tuning of the two steps, we obtain $StockGPT_{stage1}$, which is able to predict the trend of c_i based on d_j more accurately, as well as providing detailed analysis and explanations.

3.2.3. Prediction and Post-process

As shown in Figure 3 ③, given the Input I_i, we leverage StockGPT to predict the rise and fall of the stock, which can be viewed as a binary classification task. By feeding $StockGPT_{stage1}$ the input I_i, we obtain the response text Res_i about c_i.

The format of Res_i could be referred to as Sec 2.2. The `<Prob>` is a category in the range of: $\{very\ large,\ large,\ medium\ to\ upper,\ average\}$, which could be a piece of supplementary information for the investor's decision-making.

$$Res_i = StockGPT_{stage1}(I_i) \qquad (5)$$

Then, we manually extract the prediction result $Pred_i$ from Res_i. Finally, we choose all stocks predicted as "up" as C_{chosen}.

$$Pred_i = \begin{cases} up, & if\ "up" \in Res_i \\ down, & else \end{cases} \qquad (6)$$

$$C_{chosen} = \{c_i | Pred_i = "up"\} \qquad (7)$$

Additionally, we implement this investment strategy rolling by months. Every month, for all the c_i in C_{chosen}, we hold them throughout the month. The proportion of each stock in the portfolio is calculated via a capitalization-weighted approach.

$$AR_m = AR_{m-1} + \sum_{c_i \in C_{chosen}} w * R_{c_i} \qquad (8)$$

where AR_m is the accumulated return of month m, and R_{c_i} is the return of stock c_i. w denotes the proportion of stock c_i in the portfolio. v_i is the market value of company c_i.

$$w = \frac{v_i}{\sum_{n=1}^{N} v_n} \qquad (9)$$

3.3. Stage-2: Financial Q&A

Besides stock trend prediction, the proposed Stock-Chain also has the ability for financial Q&A, which could be more constructive for investors.

Given a dialogue history H_t, user query Q_t, and the retrieved document d_j related to Q_t, conversation system π can give a response R_t. We adopt RAG to enhance the Q&A ability of LLMs, which typically includes three parts: vector DB construction, knowledge retrieval, and response generation.

3.3.1. Vector DB Construction

As shown in Figure 3 stage 2 ①, vector DB is an important part of RAG, which is used for efficient storage and retrieval of knowledge documents.

Knowledge Extraction   To improve the accuracy and efficiency of document retrieval, we extract the key knowledge from the document. We adopt two extraction strategies: coarse-grained document-level summarizing with Chat-GPT, and fine-grained entity-level dialogue generation through RefGPT (Yang et al., 2023a). For document d_k, the extraction process for the two strategies is as follows:

$$s_k = ChatGPT(d_k) \qquad (10)$$

$$(q_{k0}, a_{k0}), (q_{k1}, a_{k1}), ... = RefGPT(d_k) \qquad (11)$$

where s_k denotes the summary of d_k, and $(q_{k\_}, a_{k\_})$ is query-answer pair of the generated dialogues. For example, for d_k about k line, $q_{k\_}$ could be "What is the meaning of k line?".

Knowledge Embedding   We take the extraction strategy of ChatGPT as an example. Given summary s_k, we obtain the embedding vector e_{s_k} via a sentence embedding model. This vector would be stored in the database for subsequent retrieval.

$$e_{s_k} = SentEmbed(s_k) \qquad (12)$$

where $SentEmbed$ is a sentence embedding model, such as BGE (Xiao et al., 2023) and SGPT (Muennighoff, 2022). We adopt BGE as the embedding model in our framework.

Continuous Updating   Finally, we construct a vector DB including reports, market data, and financial news. The knowledge documents in the DB could be continuously updated via online data backflow, to keep the knowledge in real time.

3.3.2. Knowledge Retrieval

To retrieve knowledge in the vector DB, user query Q would also be fed into the same sentence embedding model to obtain the embedding vector e_Q.

$$e_Q = SentEmbed(Q) \qquad (13)$$

We choose the document with the highest cosine similarity to the query as the external knowledge aided StockGPT in generating responses.

$$d^* = \arg\max_{d_k} \frac{e_Q^\top \cdot e_{s_k}}{|e_Q||e_{s_k}|} \qquad (14)$$

where s_k and d_k could be replaced by $q_{k\_}$ and $a_{k\_}$ for the extraction strategy of RefGPT.

3.3.3. LLMs Fine-Tuning

We inherit $StockGPT_{stage1}$ as the base LLMs in this part, then continue training $StockGPT_{stage1}$ on the research dataset, financial news, and StockQA datasets of AlphaFin to obtain $StockGPT_{stage2}$.

3.3.4. Response Generation

Given a dialogue history H_t, user query Q_t, and retrieved document d^* related to Q_t, the goal is to give the response R_t in a conversation of turn t. We provide a prompt template $Prompt_2$ as follows:

```
  You are an intelligent assistant,
please answer my question.  To help
you ...  local knowledge base is pro-
vided as follows:  <knowledge>
Now, answer the question...:  <his-
tory><query>
```

Then, we concatenate the prompt template, retrieved knowledge, conversation history, and user query to get the input I_t for LLMs. Feeding I_t into StockGPT, we can get the response R_t.

$$I_t = concat(Prompt_2, d^*, H_t, Q_t) \qquad (15)$$

$$R_t = StockGPT_{stage2}(I_t) \qquad (16)$$

4.   Experiments

In this section, we conduct experiments to validate Stock-Chain's ability to accomplish the task of financial analysis. Due to the structure of our framework, experiments can be divided into two parts. The experiments in the first part mainly examine the model's annualized rate of return and accuracy. In the second part, we demonstrate the performance of our Stock-Chain via preference evaluation with human&GPT-4, ablation study, and case study.

4.1.   AlphaFin-Test Datasets

We select a subset of data from the data sources, which is excluded from the training dataset, to serve as our test dataset. Given that all the research datasets are in English, our main focus is on sampling from other datasets. Such as financial reports and StockQA datasets. For stage 1, we choose the test datasets from the financial report dataset. An example demonstration is as follows: `According-ing to ..., please judge the trend of the company and give a clear answer up or down.  Input: <reports & stock price>, Output:  <Up/down>.` As for stage 2, the test dataset is sampled from the StockQA and research datasets. The AlphaFin-test datasets allow us to evaluate the model's ability in the capital market.

4.2.   Baselines

To fully validate the effectiveness of our Stock-Chain on the test dataset, we select four categories of models:

- **Major Indices**: We select indices in the Chinese capital market, including the SCI, CSI 300, SSE50, and CNX.
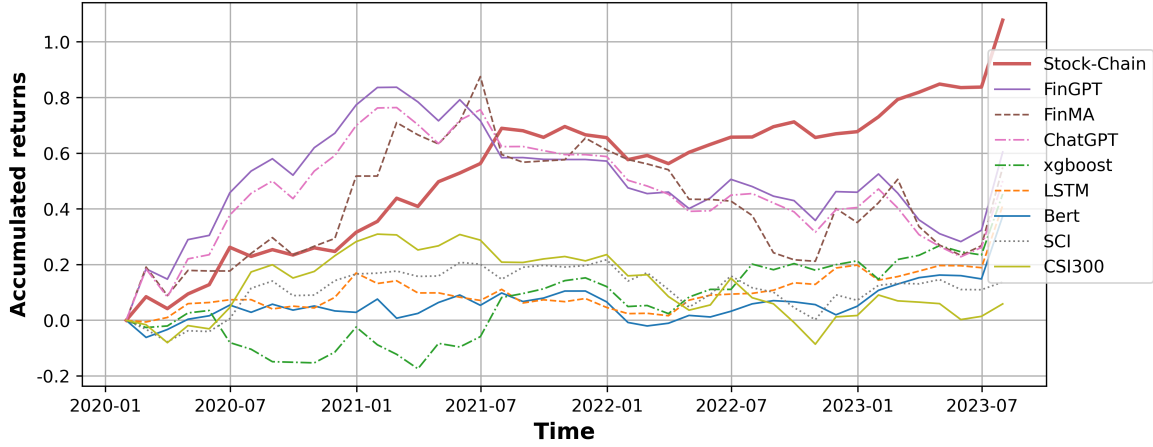
Figure 4: Accumulated returns (AR) of each baseline under the test set of the financial report dataset from January 2020 to July 2023. The figure shows the curves of some baselines.

Model	ARR ↑	AERR ↑	ANVOL ↓	SR ↑	MD ↓	CR ↑	MDD ↓	ACC ↑
SSE50	-1.0%	-2.7%	19.3%	-0.054	45.9%	-0.023	29	-
CSI 300	1.7%	0	18.2%	0.092	39.5%	0.043	30	-
SCI	3.9%	2.2%	14.8%	0.266	21.5%	0.183	19	-
CNX	7.6%	5.9%	26.5%	0.287	41.3%	0.185	20	-
Randomforest	9.8%	8.1%	19.5%	0.501	16%	0.608	22	55.5%
RNN	8.1%	6.4%	10.9%	0.742	15.7%	0.515	12	54.1%
BERT	10.7%	9.0%	16.1%	0.664	13.5%	0.852	14	51.4%
GRU	11.2%	9.5%	13.7%	0.814	14.6%	0.765	21	54.7%
LSTM	11.8%	10.1%	15.4%	0.767	15.3%	0.768	19	55.2%
Logistic	12.5%	10.8%	27.1%	0.463	32.5%	0.385	18	54.8%
XGBoost	13.1%	11.4%	20.5%	0.633	20.9%	0.619	17	**55.9%**
Decision Tree	13.4%	11.7%	19.6%	0.683	**11.9%**	1.126	20	55.1%
ChatGLM2	8.1%	6.4%	24.9%	0.324	62.6%	0.126	26	49.5%
ChatGPT(3.5Turbo)	14.3%	12.6%	27.7%	0.516	53.6%	0.267	23	51.4%
FinMa	15.7%	14.0%	37.1%	0.422	66.3%	0.236	25	49.1%
FinGPT	17.5%	15.8%	28.9%	0.605	55.5%	0.312	24	50.5%
Stock-Chain	**30.8%**	**29.1%**	**19.6%**	**1.573**	13.3%	**2.314**	**10**	55.7%

Table 2: The main experimental results on AlphaFin-Test. ARR (Annualized rate of return) and ACC (Accuracy rate) are core indicators, while the middle indicators (like AERR, ANVOL, etc.) could assist investors in evaluating the model's effectiveness. Since the rate of return usually fluctuates wildly, to ensure the stability of the performance, we run each model 10 times and obtain the average result.

- **ML&DL Algorithms**: We employ ML algorithms such as Logistic and XGBoost, and DL models like LSTM and GRU, which are widely employed for time-series prediction.

- **General LLMs**: We focus on the general-purpose LLMs like ChatGLM2 and ChatGPT. These LLMs have been chosen due to their ability and wide range of applications in NLP.

- **FinLLMs**: In the financial domain, we focus on open-source FinLLM, such as FinGPT and FinMA, which have been trained for financial tasks like financial analysis and forecasting.

4.3. Settings

For stage 1, the experiments aim to predict the trend of stock price for the next month and observe the model's returns in the real market.

- ML&DL: Due to their limitations, they can only analyze time series data. Thus, their inputs are restricted to the stock prices.

- LLMs: In contrast, LLMs possess generative abilities, allowing them to incorporate both the report data and the stock price series data.

For stage 2, we examine the model generation capability and use GPT4&human as the evaluator.

All LLMs' generation strategies are greedy search to achieve optimal and stable performance and integrated RAG in our experiments.

Among them, the hyperparameters are as follows: batch size 16, LoRA rank 8, cosine lr scheduler, learning rate 5e-5, bf16, and 1 NVIDIA A800-80GB for all training processes. Specifically, in stage 1, we trained 4 epochs for the first step and 20 epochs for the second step. In stage 2, we used $StockGPT_{stage1}$ as the base model, and incremental fine-tuned it for 2 epochs on the AlphaFin dataset.

4.4. Metrics

For stage 1, we utilize two categories of indicators. The first category is core indicators, including ARR and ACC, which gauge profitability. The second category is supplementary indicators that assist in analyzing different models, such as maximum drawdown (MD), Calmar Ratio (CR), and Sharpe ratio (SR), which measure risk assessment. With these indicators, we have a thorough evaluation of the model's capabilities. For stage 2, we use the ROUGE (Chin-Yew, 2004) as an evaluation metric, which is used to measure the similarity between the generated output and reference information. Also, we use GPT4&human as the referee for scoring. By considering these indicators, we can better evaluate the performance of the models.

Model	ARR ↑	SR ↑	Out_len ↑	N/A ↓
ChatGLM2	8.1%	0.324	228.1	52.3%
w/ raw_data	15.8%	0.636	17.2	-
w/ CoT_data	10.1%	0.469	**476.1**	32.4%
Stock-Chain	**30.8%**	**1.573**	254.8	25.9%

Table 3: The ablation results for stage 1 under different training data. Out_len: average length of LLMs outputs. N/A: invalid answer ratio.

Model	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑
ChatGLM2	0.2794	0.1944	0.2642
w/ Fin. News	0.3477	0.2821	0.3445
w/ Fin. reports	0.2611	0.1603	0.2396
Stock-Chain	**0.4352**	**0.3056**	**0.4031**

Table 4: The ablation results for stage 2 under different training data.

4.5. Comparison Results

As shown in Figure 4, the curve represents the AR of each method. It is noteworthy that Stock-

Chain achieves the highest AR and maintains an upward trend, starting from 2023. It indicates the effectiveness of Stock-Chain in investment.

By referring to Table 2, Stock-Chain achieves the highest 30.8% of ARR demonstrating its effectiveness. Based on Table 2, we can derive the following observations:

Firstly, ML&DL possesses certain analytical abilities in stock trend prediction, they achieve impressive ARR. Secondly, after integrating report data with market data, LLMs generally surpass ML&DL, leading to an enhancement in stock trend prediction abilities. ChatGPT achieves an ARR of 14.3%. While LLMs are trained on vast amounts of textual data, they lack optimization for financial domains. Thus, by fine-tuning for financial domains, FinLLMs can improve stock trend prediction ability. The FinGPT model achieves an ARR of 17.5%.

Finally, after fine-tuning Stock-Chain based on financial report cot data, we achieve an ARR of 30.8% and an ACC of 55.63%. AlphaFin datasets play a crucial role in the training of LLMs. By utilizing comprehensive financial data for fine-tuning, we improve prediction accuracy and return, thus validating the performance of Stock-Chain.
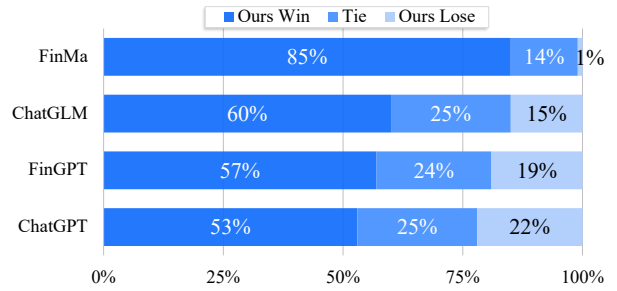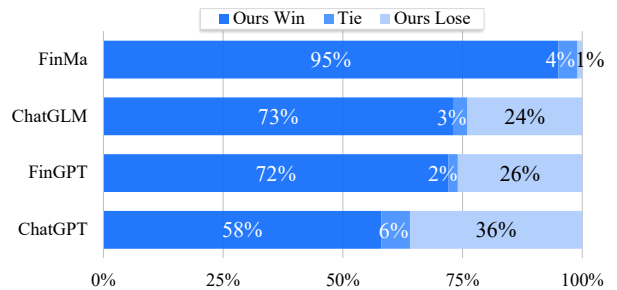

Figure 5: Preference evaluations via human.

Figure 6: Preference evaluations via GPT-4.

4.6. Ablation Study

We conduct two ablation experiments. Firstly, we observe the stock trend prediction ability of our Stock-Chain by analyzing the effects of fine-tuning at different data. Based on Table 3, compared to the ChatGLM2, the ability of the LLMs to predict stock price has shown improvement after fine-tuning with
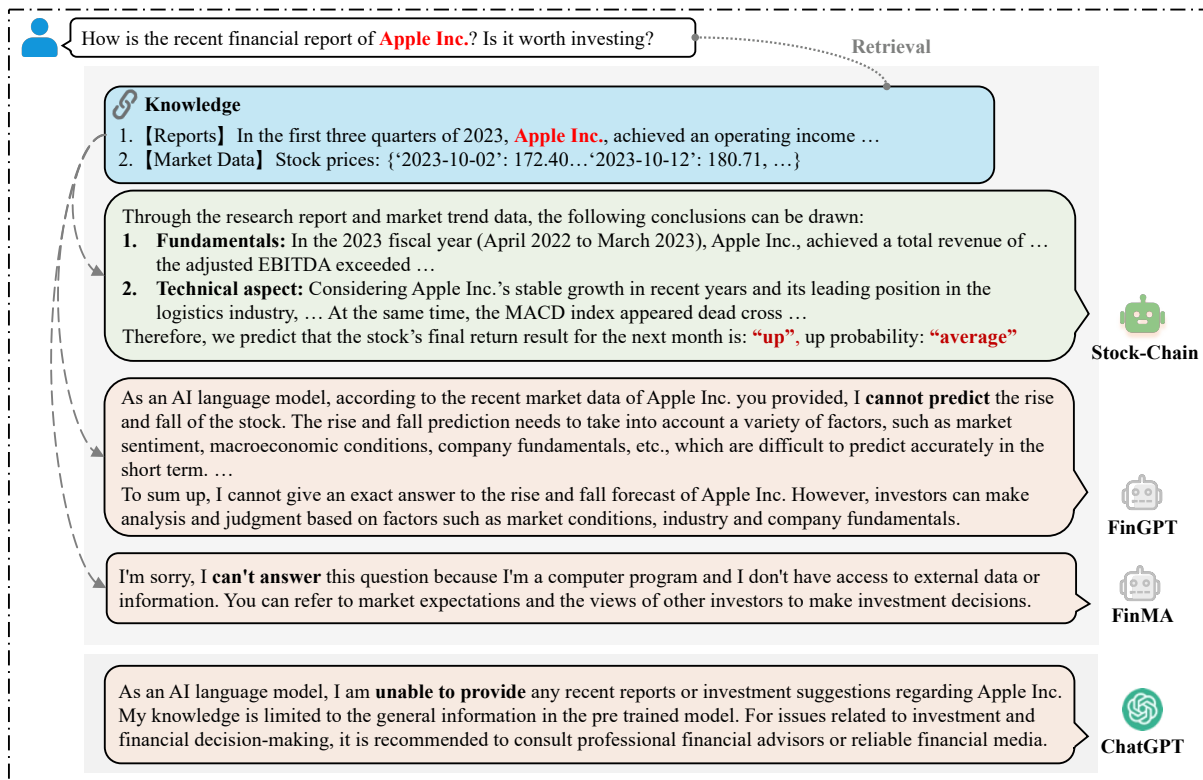
Figure 7: Case of outputs of Stock-Chain, FinGPT, FinMA, and ChatGPT

raw and CoT data, achieving returns of 15.8% and 10.1% respectively.

Additionally, the proportion of invalid answers also improved. It is worth mentioning that after fine-tuning with the raw data, the LLMs's output only includes rise and fall, thus resolving the issue of invalid answers. After fine-tuning with two sets of data, our Stock-Chain achieves optimal performance with 30.8% ARR, and the proportion of invalid answers also decreased, reaching 25.9%.

As for the second ablation experiment, we investigate whether the quality of the output improved after fine-tuning the LLMs at different data.

Based on Table 4, we observe that the scores of Stock-Chain in terms of rouge1 and rouge2 reach 0.3477 and 0.2821 respectively after fine-tuning with News data. Furthermore, it is noteworthy that Stock-Chain achieves optimal performance after fine-tuning with both news and reports.

4.7. Preference Evaluation

We employ GPT-4 and humans as the judge to rate the output performance of each LLM on the test datasets. All LLMs have integrated RAG in this experiment. In the human section, Stock-Chain outperforms other LLMs in terms of content effectiveness. Based on Figure 5, compared to Chat-GLM2, Stock-Chain achieves a win rate of over 60%, and when compared to FinLLMs, such as Fin-GPT, the win rate reaches 62%. Based on Figure 6, when GPT4 is the judge, similar conclusions were drawn. Stock-Chain exhibits higher rates compared to human ratings, with a win rate of 58% against ChatGPT and 73% against ChatGLM2. Overall, Stock-Chain's output is effective.

4.8. Case Study

We present partial outputs of Stock-Chain for qualitative analysis. As shown in Figure 7, when users inquire about recent reports and investment advice related to Appe Inc., Stock-Chain acquires real-time relevant information and market data from the knowledge base and provides them as input to the LLMs. The output of the LLMs has been enhanced, and the news remains up-to-date, enabling investors to analyze and receive recommendations. However, as for ChatGPT and FinGPT, we observe that it has great defects in the quality and real-time response. Thus, by integrating RAG, Stock-Chain addresses the issues of hallucinations and insufficient real-time outputs in LLM, enhancing the LLMs's practicality and ability.

5. Related Work

5.1. Financial Datasets

General financial datasets include a wealth of information derived from various sources within the financial industry, such as Internet data and proprietary data. Currently, the main financial dataset sets include a variety of tasks. FPB (Malo et al., 2014) and FiQA-SA (Maia et al., 2018) are mainly used for emotion analysis. The Headline (Sinha and Khandait, 2021) dataset is primarily utilized for news headline identification. For question answering, we primarily utilize the FinQA (Chen et al., 2022a) and ConvFinQA (Chen et al., 2022b) datasets. Regrettably, the finance domain still suffers from a scarcity of text datasets, impeding the development of FinLLMs. To bridge this gap, we propose AlphaFin, providing support for the financial industry in training its own FinLLMs.

5.2. Algorithms in Financial Domain

Traditional ML&DL algorithms, such as LSTM (Yu et al., 2019), Logistic (Sperandei, 2014), and BERT (Devlin et al., 2018), have been applied in stock trend prediction. However, ML&DL focuses on the final result, without analyzing the underlying factors driving market trend. As for FinLLMs, although BloombergGPT (Wu et al., 2023), FinMA (Xie et al., 2023), and FinGPT (Yang et al., 2023b) play important roles in the community, they are mainly based on English-language datasets. In contrast, Stock-Chain relies on Chinese-language and is specifically designed for stock trend prediction.

As for RAG, it has attracted growing attention in the community (Li et al., 2022). Compared to the traditional method, RAG has remarkable performance in various NLP tasks (Cai et al., 2021; Weston et al., 2018). However, without RAG, FinLLMs often produce hallucinations and meaningless outputs. Thus, we integrate LLMs with RAG to address the above issues.

6. Conclusion

In this work, we formalize the task of financial analysis, and propose AlphaFin datasets to enhance LLMs's ability, and fine-tune the StockGPT upon it. Then we propose the Stock-Chain framework that integrates a real-time financial database through RAG, to address the issue of hallucination of LLMs's output and LLMs's inability to generate real-time content. We conduct extensive experiments on the proposed AlphaFin datasets, as well as some supplementary experiments like ablation study, GPT4&human preference evaluation and case study, to reveal that Stock-Chain outperforms all the baseline methods, and shows effectiveness for the task of financial analysis.

7. Ethical considerations and limitations

We assert that there are no ethical dilemmas surrounding the submission of this article and have no known competing financial interests or personal relationships that could have had an impact on the research work presented.

Despite the positive contributions of this study, we recognize that there is still great room for development in our work. In our future work, we will further contribute to the open-source FinLLMs, improve its generalization, enhance its ability in other financial tasks, and create a more powerful open-source FinLLMs.

8. Bibliographical References

AKshare. 2021. Akshare - a financial online documentation.

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Siqi Bao, Huang He, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, Zhihua Wu, Zhen Guo, Hua Lu, Xinxian Huang, et al. 2021. Plato-xl: Exploring the large-scale pre-training of dialogue generation. *arXiv preprint arXiv:2109.09519*.

Deng Cai, Yan Wang, Huayang Li, Wai Lam, and Lemao Liu. 2021. Neural machine translation with monolingual translation memory. *arXiv preprint arXiv:2105.11269*.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2022a. Finqa: A dataset of numerical reasoning over financial data.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022b. Convfinqa: Exploring the chain of numerical reasoning in conversational finance question answering. *arXiv preprint arXiv:2210.03849*.

Lin Chin-Yew. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out, 2004*.

DataYes. 2021. Datayes - financial data and analytics platform.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mark Dredze, Prabhanjan Kambadur, Gary Kazantsev, Gideon Mann, and Miles Osborne. 2016. How twitter is changing the nature of financial news discovery. In *proceedings of the second international workshop on data science for macromodeling*, pages 1–5.

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*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022. A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110*.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: financial opinion mining and question answering. In *Companion proceedings of the the web conference 2018*, pages 1941–1942.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796.

Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *arXiv preprint arXiv:2305.15852*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Emad W Saad, Danil V Prokhorov, and Donald C Wunsch. 1998. Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on neural networks*, 9(6):1456–1470.

Jaimin Shah, Darsh Vaidya, and Manan Shah. 2022. A comprehensive review on multiple hybrid deep learning approaches for stock prediction. *Intelligent Systems with Applications*, page 200111.

Ankur Sinha and Tanmay Khandait. 2020. Impact of news on the commodity market: Dataset and results.

Ankur Sinha and Tanmay Khandait. 2021. Impact of news on the commodity market: Dataset and results. In *Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 2*, pages 589–601. Springer.

Sandro Sperandei. 2014. Understanding logistic regression analysis. *Biochemia medica*, 24(1):12–18.

Tushare. 2021. Tushare - a financial data interface.

Jason Weston, Emily Dinan, and Alexander H Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. *arXiv preprint arXiv:1808.04776*.

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*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. Pixiu: A large language model, instruction data and evaluation benchmark for finance.

Dongjie Yang, Ruifeng Yuan, Yuantao Fan, , Yifei Yang, Zili Wang, and Shusen Wang. 2023a. Refgpt: Reference-to-dialogue by gpt and for gpt. https://github.com/ziliwangnlp/RefGPT.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023b. Fingpt: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*.

Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270.

Boyu Zhang, Hongyang Yang, and Xiao-Yang Liu. 2023. Instruct-fingpt: Financial sentiment analysis by instruction tuning of general-purpose large language models. *arXiv preprint arXiv:2306.12659*.