Supporting Information for

Machine Behaviorism: Exploring the Behavioral Dynamics of Large Language Models in Decision Making

Yongxin Ni, Tianqi Xu, Chunxiao Li, Zhiqiang (Eric) Zheng, and Bin Gu

Corresponding author:
Zhiqiang (Eric) Zheng.
ericz@utdallas.edu;
Bin Gu.
E-mail: bgu@bu.edu;

This PDF file includes:

Figs. S1 to S8
Tables S1 to S3

A. Construction of LLM-based Financial Stock Trading Environment and Prompt Settings. To delve deeper into whether large language model (LLM)-based agents exhibit human-like biases, such as the disposition effect, we designed an experiment by creating an interactive market environment for these LLM-based agents to make stock trading decisions. This experimental setting provided real stock trading data, but to prevent agents from leveraging training data to predict future changes, all stock data were anonymized. We employed GPT-3.5 and GPT-4 as the foundational language models for our investment agents in the experiment and recorded their decision-making behaviors within the trading system.

In the construction of the stock trading environment, we did not implement a stock selection strategy. Instead, we only implemented a timing strategy for individual stocks and a simple position strategy. This is because in classical stock quantitative trading algorithms, the selection strategy, timing strategy, and position strategy are in a complex superimposed state, which is not conducive to our causal decoupling of the disposition effect in LLM decision-making. Conversely, when we only consider buying/selling individual stocks, we can reasonably attribute the disposition effect to agents' mentality of being content with small gains, as well as the personification of overemphasizing sunk costs.

A.1. Stock Data Preprocessing. At each decision-making stage, we provide historical K-line data of the stock (opening price, highest price, lowest price, and closing price), along with records of the LLM's previous decisions. To avoid interference from factor coupling, we did not inform the LLM about the news related to the stock. It is worth noticing that we have removed incomplete samples of stock data in each set of experiments to ensure reasonability. At the same time, the stocks are anonymized to prevent LLMs from obtaining subsequent changes through training data.

After collecting stock data from markets in two countries, we processed the stock data into the following Table S1, and its detailed description is shown in Table S2.

Table S1. The standard format of preprocessed financial stock data.

Date	code	open	high	low	close	change	decision	cash	hold_num	equity
2013/07/25	xxx2	43.41	44.43	43.35	44.24	None				
2013/07/26	xxx2	45.01	45.12	44.17	44.79	0.0124				
2013/07/29	xxx2	44.72	45.45	44.47	45.06	0.0060				
2013/07/30	xxx2	...	...	...	...	...				

Table S2. Detailed Description of Each Column.

Column Names	Description
date	The stock trading date.
code	The desensitized stock identifier. The purpose of desensitizing the stock identifiers is to prevent agents from using prior knowledge of a certain stock from its training corpus to predict stock trends.
open	Stock open price of the day.
high	Stock highest price of the day.
low	Stock lowest price of the day.
close	Stock close price of the day.
change	The relative change in stock close price between two trading days.
decision	The decisions that LLM-based agents made based on historical candlestick data and trading records. The decisions shall be within the range of {"buy" / "hold" / "sell"}.
cash	The cash held by agents after trading actions.
hold_num	The number of shares held by agents after trading actions.
equity	Equity owned by agents after trading actions equals the sum of cash and the market value of held shares.

During the data preprocessing stage, the columns for "decision", "cash", "hold_num", and "equity" are left blank. These four columns will be populated only after the initiation of the agents' stock investment decision-making process.

In our experimental setup, it's evident that the investment agent makes decisions for the subsequent round based on historical candlestick charts and its own decision records, which include changes in cash, hold_num, and equity. It's worth noting that in our experimental framework, the value of holding stocks is determined by the closing price of the day, while buying and selling of stocks occur at the opening of each trading day. Consequently, the decisions made on each trading day effectively impact the trading operations executed by the agents at the opening of the following trading day.

In the default configuration, prior to the first launch of the stock trading decision system for each stock, we present LLMs with the K-line history of each stock over the last 10 trading cycles. The purpose of this design is to advance the LLM's understanding of historical stock fluctuations, thereby enabling the agent to better adapt to the current simulated trading environment. This approach also aims to simulate the human decision-making process, acknowledging that individuals typically undergo a period of observation before engaging in decision-making.

A.2. The Construction of Timing Strategy. In the original data, the historical trends of each stock are first organized on a daily basis. In order to better simulate the trading behavior of traders at different frequencies, we merged the stock data according

[System Prompt]: *As an ordinary shareholder, you start with an initial equity of $1,000,000. Your task is to make trading decisions based on the provided historical dataset to maximize your equity.*

The dataset includes the following columns: ["date", "code", "open", "high", "low", "close", "change", "decision", "hold_num", "cash", "equity"]. The "date" column records the stock trading date. The "code" column represents the code of the current stock. The K-line of the stock consists of "open" (open price), "high" (highest price), "low" (lowest price) and "close" (close price). The "change" column records the change in stock's close price. The "decision" column articulates your actions, thoughtfully considering both the K-line data of the current period and the comprehensive historical data from past periods. The "hold_num" column records the number of shares you hold, and the "cash" column records your cash on hand in this period.The "equity" column is calculated by summing your stock value and cash on hand, which is also your maximizing target. Besides, the stock value is calculated by multiplying the "hold_num" column and the "close" column.

You have to select your decision between ['buy', 'sell', 'keep']. If you choose 'buy', you will use 20% of your cash to buy the stock. If you choose 'sell', on the contrary, you will sell 20% of the shares you hold and receive the corresponding cash. If you choose 'keep', you conduct neither buying nor selling, not making any new trades.

Please note that you cannot perform any operations during the first 10 periods. These 10 periods are used to show you the current trend of the stock.

Please be aware that purchasing stocks incurs a 0.025% commission fee, and when selling stocks, a 0.025% commission fee and an additional 0.1% tax apply. Additionally, stocks are only purchasable in multiples of 100 shares.

Ensure that the response is presented exclusively in JSON format, adhering strictly to the following structure: {'decision': 'buy'/'sell'/'keep'}. Any deviation from this format is not acceptable.

Fig. S1. The prompt of timing strategy construction for Chinese stocks. The text highlighted in yellow entail special trading rules and transaction fees in the Chinese A-shares market.

to weekly and monthly frequencies. For the **Observation** stage, the default trading period we use is monthly. As for the **Hypothesis Formulation** and **Intervention** phases, we take weekly windows as the default trading period to create more investment samples since we limit the time frame between 2022 and 2023. The results on a monthly basis are qualitatively similar and available upon request.

The participants (LLM-based agents) were endowed with an initial capital of 1,000,000 (in RMB for Chinese stocks / in USD for US stocks), and they were required to make buy, sell, or hold decisions for a given stock during each period. Before the experiment began, we provided the agents with a stock history of 10 cycles to help them become more familiar with real-world stock changes. This means that during the initial 10 cycles, LLM agents were not allowed to engage in any trading activities. Before each round of decision-making, the actual stock prices from the historical periods were released to compare with their historical investment decisions. To guide LLMs in making decisions, we provided the following basic prompt, as shown in Figure S1 for Chinese stocks and Figure S2 for US Stocks.

[System Prompt]: *As an ordinary shareholder, you start with an initial equity of $1,000,000. Your task is to make trading decisions based on the provided historical dataset to maximize your equity.*

The dataset includes the following columns: ["date", "code", "open", "high", "low", "close", "change", "decision", "hold_num", "cash", "equity"]. The "date" column records the stock trading date. The "code" column represents the code of the current stock. The K-line of the stock consists of "open" (open price), "high" (highest price), "low" (lowest price) and "close" (close price). The "change" column records the change in stock's close price. The "decision" column articulates your actions, thoughtfully considering both the K-line data of the current period and the comprehensive historical data from past periods. The "hold_num" column records the number of shares you hold, and the "cash" column records your cash on hand in this period.The "equity" column is calculated by summing your stock value and cash on hand, which is also your maximizing target. Besides, the stock value is calculated by multiplying the "hold_num" column and the "close" column.

You have to select your decision between ['buy', 'sell', 'keep']. If you choose 'buy', you will use 20% of your cash to buy the stock. If you choose 'sell', on the contrary, you will sell 20% of the shares you hold and receive the corresponding cash. If you choose 'keep', you conduct neither buying nor selling, not making any new trades.

Please note that you cannot perform any operations during the first 10 periods. These 10 periods are used to show you the current trend of the stock.

Please be aware that when you sell stocks, you need to pay a handling fee of (0.0021% of the transaction amount + US$0.000119 per share).

Ensure that the response is presented exclusively in JSON format, adhering strictly to the following structure: {'decision': 'buy'/'sell'/'keep'}. Any deviation from this format is not acceptable.

Fig. S2. The prompt of timing strategy construction for US stocks. The yellow highlights represent transaction fees in the US stock market.

To ensure the simulation of real-world trading rules, the prompts in Figures S1 and S2 include detailed instructions and constraints specific to each stock market. For the Chinese stock market, the prompt emphasizes the transaction fees and special rules such as the 0.025% commission fee and 0.1% tax on transactions, as well as the requirement for stocks to be purchasable in multiples of 100 shares. These rules are highlighted in yellow to draw attention to their importance in accurately simulating the trading environment.

For the US stock market, the prompt includes a commission fee of 0.0021% of the transaction amount plus a fixed fee per share, also highlighted in yellow. These fees are included to ensure that the LLM-based agents consider the impact of transaction costs on their trading decisions, thereby promoting a more realistic trading simulation.

In the construction of the stock trading environment, we did not consider a stock selection strategy. Instead, we only implemented a timing strategy for individual stocks and a simple position strategy. This is because in classical stock quantitative trading algorithms, the selection strategy, timing strategy, and position strategy are in a complex superimposed state, which is not conducive to our causal decoupling of the disposition effect in LLM decision-making. Conversely, when we only consider buying/selling individual stocks, we can reasonably attribute LLM's disposition effect to its mentality of being content with small gains, as well as the personification of overemphasizing sunk costs.

By incorporating these prompts, we aim to provide a structured and controlled environment for the LLM-based agents to make trading decisions, reflecting the real-world complexities and costs associated with trading in different stock markets. The experimental design required agents to make decisions to buy, sell, or hold based on historical and current period data in each decision-making cycle. Through this simulation experiment, we hope to understand whether data-driven models would display similar behavior patterns in the absence of human emotional and cognitive biases.

A.3. The Construction of Equity-Calculation System. To thoroughly simulate and analyze the dynamics of agents' equity in this context, we developed an equity calculation system. This system meticulously simulates how LLM-based agents make decisions based on the provided stock trading data, and how these decisions impact the value of their investment portfolios.

Initially, at the 10th cycle, the system is set up with indicators such as the number of stocks held, stock value, actual position size, cash on hand, and total assets. This initialization process simulates the observation period before actual trading starts, ensuring agents have sufficient data background to make more informed decisions.

Starting from the 11th cycle, the equity calculation system iterates through each trading cycle, adjusting the agents' portfolios based on the decisions made in the previous cycle to buy, sell, or hold, as well as the market conditions of the current cycle. This includes logic to handle price changes, ensuring that if significant market changes occur, the number of stocks held by agents can be accordingly adjusted to maintain the continuity and adaptability of the investment strategy.

In terms of decision-making logic, the equity calculation system precisely calculates the cash and share portions based on the agents' decisions to buy or sell, including calculating slippage and trading costs to ensure that the financial consequences of trades are accurately reflected in the agents' portfolios. This approach not only demonstrates the agents' ability to analyze and understand market data but also showcases their strategic planning capabilities for long-term asset appreciation.

Slippage: Slippage refers to the deviation between the actual execution price and the expected price during the execution of a trade, caused by market fluctuations or insufficient liquidity. Typically, traders aim to execute trades at a specific price, but due to instant changes in market conditions, the actual execution price may be higher or lower than the expected price, resulting in slippage. Slippage often leads to increased trading costs, particularly important to control for high-frequency trading or large-volume trades. **Here we set slippage at 0.01/share.**

Stock Ex-Dividend Adjustment: When a stock undergoes ex-dividend adjustment, there is a significant price change between the day before and the current day, which may cause some decision-making interference for LLM agents. To mitigate this interference, we implemented a stock adjustment algorithm in our code to correct the adjusted quantity of stocks held by the agent.

Trading Costs: The trading costs for stocks differ between the market in China and the US stock market. In the Chinese market, buying stocks incurs a trading commission of approximately 0.025%, while selling stocks incurs a trading commission of approximately 0.025% plus a 0.1% stamp duty. On the other hand, in the US stock market, buying stocks has no cost, while selling stocks incurs a trading activity fee of $0.000119 per share to FINRA, and a Securities Exchange Commission (SEC) fee of 0.00051% of the transaction value.

Trading Details: In the US stock market, there is no concept of daily price limits (up or down); however, in the Chinese market, if the opening price differs from the previous day's closing price by more than 10%, trading for that stock is halted for the day, and no further buying or selling can occur. In the US stock market, the quantity of stocks purchased is unrestricted, whereas in the Chinese market, only a whole hundred shares can be purchased. The US stock market allows for day trading, while the trading restriction in the Chinese market is N+1, meaning each stock can only be traded once per day. These limitations are reflected in our code implementation.

By updating the portfolio's stock value, total assets, and actual position size at the end of each trading cycle, the equity calculation system continuously tracks the financial condition of the agents. This dynamic tracking method allows us to deeply analyze whether LLM-based agents exhibit behavior biases similar to human investors, such as the disposition effect, and how they make decisions based on data-driven models without the influence of external intervention from environments and humans.

The detailed simulation of asset dynamics not only provides us with valuable insights into the application of AI in financial market decision-making but also offers a new perspective for simulating and analyzing human economic behaviors, further deepening our understanding of how complex behavioral patterns can be simulated by AI agents.

B. Implementation Details of Intervention Strategies. Here are the specific implementation details of each intervention strategy. Some interventions are directly implemented by modifying the prompt, while others (including those based on contextual factors and agent intelligence) involve more complex technical details. For the former, we present the corresponding prompt diagram, and for the latter, we provide a detailed description of the technical implementation along with the prompt.

```
input_data['follower number'] = ''

for k in range(len(input_data)):
    input_data.at[k, 'follower number'] = random.randint(10000, 100000)
```

Fig. S3. Illustration of the spotlight effect on stock trading decisions. The figure shows how introducing a "follower number" for each stock influences LLM-based agents, simulating social scrutiny and optimizing decision-making behavior.

B.1. Social Factors: Spotlight Effect. To effectively implement social factors in the intervention stage, we simulate the "spotlight effect", referring to the phenomenon where individuals alter their behavior when they believe their actions are observable by others. In the context of stock trading, this can be simulated by introducing a "follower number" to each stock, implying that the trading decisions are being observed (and possibly being followed) by a large audience. As illustrated in Fig. S3, we demonstrate how to leverage the spotlight effect to optimize agents' decision-making behavior in stock trading.

- Initialize Follower Number: Each stock is assigned a random number of followers to simulate the spotlight effect, creating a sense of social scrutiny for the LLM-based agents. This number acts as a social factor influencing the decision-making process.
- Historical Data Processing: Use historical data to make buy, sell, or hold decisions. Incorporate the follower number into the decision-making process to simulate the impact of social factors. The agent reviews past trading data to make informed decisions while being aware of the number of followers, simulating the pressure of public scrutiny.
- Decision Making: Adjust the decision-making process to factor in the social scrutiny represented by the follower number. Agents might behave more cautiously, mimicking the spotlight effect where they anticipate public evaluation of their trades. The agent's buy/sell/hold decisions are adjusted based on the number of followers, reflecting cautious behavior typically seen in humans under observation.

This approach leverages the social dimension to modify agent behavior, potentially reducing biases such as the disposition effect by simulating the pressure of social observation.

```
# Add "reference Point"

message_our = [
    {"role": "system", "content": system_prompt}
]
# Process each row of data, e.g., print or perform other operations
for row in counter.itertuples():
    user_prompt = f"{row}"
    if buy_price == -1:
        user_prompt += "\n" + f'Please note that you have not bought any shares yet.'
    else:
        user_prompt += "\n" + f'Please note that the average purchase price of your current holdings is {buy_price}'
    user_prompt_dic = {"role": "user", "content": user_prompt}
    message_our.append(user_prompt_dic)
```

Fig. S4. The prompt of changing agents' reference points. By recalculating the average purchase price after each transaction, the system reinforces the significance of the reference point, aiding agents in making more informed investment decisions.

B.2. Psychological Factors: Reference Point. The reference point is the benchmark used by individuals to evaluate gains and losses. In behavioral economics, Kahneman and Tversky's Prospect Theory states that people tend to evaluate outcomes relative to a reference point rather than in absolute terms. The Anchoring Effect refers to the tendency of individuals to heavily rely on the first piece of information (the anchor) when making decisions, which significantly influences subsequent judgments and decisions.

We illustrate how we change agents' inward reference points in Fig. S4. In our code, the concept of the reference point is implemented by updating the 'buy-price'. Every time the agent decides to buy stocks, a new weighted average purchase price is calculated, which serves as the new reference point. In our implementation, by continuously updating and reminding the agent of its original purchase price, we reinforce the significance of this reference point. The specific steps in the implementation are as follows:

- 1. Updating Purchase Price: After each stock purchase, the agent's purchase price ('buy-price') is updated to ensure this price always reflects the latest holding costs.

- 2. Generating Prompt Messages: During each decision point, a prompt containing the current purchase price of holdings is generated to ensure the agent can refer to this anchor when making decisions.

By implementing this approach, we introduce the concept of a reference point in the system, leveraging the Anchoring Effect from psychology to improve the agent's decision-making behavior, reduce the impact of the Disposition Effect, and enhance overall rationality and effectiveness in investment decisions.

B.3. Contextual Factors: Domain knowledge. This strategy illustrates how contextual factors,e.g., Domain knowledge, can be leveraged to fine-tune agent behavior, resulting in more rational and informed financial decisions. In our experimental setup, LLM agents act as financial decision-makers, engaging in stock trading. Although these agents have a deep understanding of the stock market from their internal knowledge bases, they may lack the necessary investment background and up-to-date external informaiton on a stock. This can lead to behavioral biases similar to those exhibited by average investors.

To address this, we introduce an innovative strategy that integrates external investment background knowledge, and specific stock expert analysis into the agents' decision-making processes. This approach is informed by historical research, which suggests that incorporating information from diverse sources can create a more comprehensive understanding of the stock market, thereby reducing behavioral biases, as is shown in Figure S5. Specifically, we emphasizes two key forms of knowledge integration:

- General Knowledge (**highlighted in green**): it provides fundamental background information on stock market operations, including the factors influencing stock prices, essential columns for analyzing stock performance, and the importance of transaction costs in financial planning.

- Specific Knowledge (**highlighted in yellow**): it focuses on using historical trading data to summarize a stock's recent characteristics and offers concise recommendations based on the identified trends. This targeted advice helps guide decision-making with practical, up-to-date information.

By combining these knowledge types, agents can make well-informed trading decisions, effectively reducing the risk of behavioral biases and improving overall financial performance.

Strategy Implementation. For contextual-based intervention, we introduce a domain-knowledge-enhancing strategy to fine-tune agent behavior. This involves instilling general investment knowledge and customized investment recommendations for specific stocks into the agents' decision processes. The strategy is divided into two parts:

1. Offline RAG for General Static Knowledge:

- Objective: Enrich agents with foundational investment knowledge.

- Method: Utilize Retrieval-Augmented Generation (RAG) technology to gather relevant background information and real-world investment experiences from the internet. This includes general trading knowledge, stock background knowledge, trading skills, and practical experience (**with green color**). This static information serves as a global knowledge base for the agents (**with yellow color**).

- Implementation: The collected data is processed and formatted into prompts that are used during the decision-making process to ensure agents have sufficient informational support.

2. Online Analysis for Specific Dynamic Knowledge:

- Objective: Provide tailored, real-time investment recommendations.

- Method: Deploy a specialized LLM assistant that analyzes the current stock's historical trends and provides specific investment recommendations. This includes technical analysis and recommendations on specific stocks.

- Implementation: The assistant processes the latest trading data, summarizing the stock's recent characteristics and generating one-sentence recommendations. These are dynamically presented to the agents during their decision-making process to guide their actions based on the most recent market conditions.

By combining foundational investment knowledge with real-time expert analysis, the contextual-based intervention strategy significantly enhances the decision-making quality of LLM agents. This dual approach ensures that agents are well-informed and can adapt to current market conditions, effectively mitigating behavioral biases such as the disposition effect.

[System Prompt]: *As an ordinary shareholder, you start with an initial equity of $1,000,000. Your task is to make trading decisions based on the provided historical dataset to maximize your equity.*

…..

[General Knowledge]: *Here is some general external information for the task. The following information serves as background knowledge relevant to this task. It has been obtained through extensive searches, manual selection, and summarization using search tools. When making decisions, it is important to carefully consider and study this information. It can serve as a solid foundation for your basic knowledge. The knowledge provided includes:*

The stock market operates on the principle of stocks representing partial ownership in companies, with prices driven by supply, demand, and various factors like company performance and economic indicators. Analyzing stock performance involves key dataset columns like 'open,' 'high,' 'low,' 'close,' and 'change,' aiding in trend identification and future predictions. Effective trading decisions consider investment goals, risk tolerance, portfolio composition, and financial planning. Transaction costs, including explicit fees and implicit costs, impact returns and should be managed through efficient order types and trade planning.

…..

[Specific Knowledge]: *Based on the historical trading data, please summarize the stock's recent characteristics and offer potential recommendations in one concise sentence (no more than 50 words).*

E.g., The stock has been in a downward trend with decreasing highs and lows, and a negative price change. Consider selling or avoiding this stock.

Fig. S5. The provided prompt for agents to absorb collected domain knowledge. The prompt includes both general and specific knowledge to support decision-making processes. General knowledge covers the foundational aspects of stock market operations, including factors like supply and demand, company performance, and economic indicators. Specific knowledge focuses on analyzing historical trading data to identify stock trends and make informed recommendations.

B.4. Prompting Strategies: ICL and CoT. In recent advancements in the field of large language models (LLMs), prompting strategies have emerged as an effective tool to reshape the behavior and decision-making capabilities of these models. Among the various prompting techniques, In-Context Learning (ICL) and Chain of Thought (CoT) have gained significant attention due to their ability to promote analytical thinking and improve the quality of decisions made by LLMs. These strategies are designed to emulate human cognitive processes, thereby reducing biases and fostering more rational outcomes. This document illustrates the specific implementations of ICL and CoT prompting strategies and their impact on decision-making quality, using a stock trading scenario as an example.

In-Context Learning (ICL)

As is shown in Figure S6, ICL is designed to provide LLM with contextual examples within the prompt to help it make more accurate predictions. The examples act as a guide, showcasing the expected input-output behavior, thus assisting the model in understanding the task better. The ICL strategy simulates how humans learn from examples and context, promoting analytical thinking and reducing reliance on intuition.

In the provided scenario, ICL involves giving the LLM examples of stock trading decisions made in previous periods. These examples (**highlighted in yellow**), demonstrate the ground-truth results of decisions made in the first 10 periods for each stock. This serves a dual purpose: it shows the current trend of each stock and guides the model in learning how to make correct decisions based on this data. Here's how it is implemented:

- 1. Initial Setup: The system prompt sets the context for the task.
- 2. Contextual Examples: The prompt includes several examples of historical data and the decisions made during those periods. Using the provided historical data and examples, the model make new decisions:

Chain of Thought (CoT) As illustrated in Figure S7, the CoT prompting strategy involves guiding the LLM through a detailed, step-by-step reasoning process to arrive at a decision. This approach emulates deeper analytical reasoning (System 2 Thinking), where each step in the thought process is explicitly laid out. The method helps the LLM break down complex problems into multiple decision steps, considering various factors that may influence each step before reaching a decision.

The implementation of CoT (**highlighted in yellow in the prompt**) specifies the required structured output format. It emphasizes that the output must be in JSON format, with each step clearly defined, ultimately leading to a decision of "buy," "sell," or "keep." This structured output format not only ensures transparency and logic in the decision-making process but also enhances the model's reasoning capabilities in complex financial scenarios. Here's how it is implemented:

- 1. Initial Setup: Similar to ICL, the system prompt sets up the context.
- 2. Reasoning Steps: The model is instructed to provide a detailed reasoning process for each decision, breaking down the thought process into steps. The output format includes both the reasoning steps and the final decision.

Both ICL and CoT prompting strategies aim to reduce biases and improve decision-making by mimicking human cognitive processes. ICL provides contextual examples to guide the model, while CoT involves a step-by-step reasoning process to ensure transparent and logical decisions. By leveraging these strategies, LLMs can produce more rational and well-reasoned outputs, mitigating the disposition effect bias.

[System Prompt]: *As an ordinary shareholder, you start with an initial equity of $1,000,000. Your task is to make trading decisions based on the provided historical dataset to maximize your equity.*
……

[In Context Learning (ICL):] Note, to aid in your decision-making process, we provide the standard ground-truth results of decisions made in the first 10 periods for each stock as examples. This serves a dual purpose: it demonstrates the current trend of each stock and guides you in learning how to make correct decisions based on this data:

2020-01-10,XXXXXXX,17.01,17.34,16.52,16.69,-0.0285,None
2020-01-17,XXXXXXX,16.75,17.27,16.2,16.39,-0.018,sell
2020-01-23,XXXXXXX,16.43,16.61,15.39,15.54,-0.0519,sell
2020-02-07,XXXXXXX,13.99,14.89,13.99,14.62,-0.0592,buy
2020-02-14,XXXXXXX,14.51,15.14,14.3,15.03,0.028,buy
2020-02-21,XXXXXXX,15.04,15.72,14.93,15.58,0.0366,sell
2020-02-28,XXXXXXX,15.46,15.46,14.46,14.5,-0.0693,buy
2020-03-06,XXXXXXX,14.55,15.64,14.46,15.03,0.0366,sell
2020-03-13,XXXXXXX,14.71,14.88,13.9,14.52,-0.0339,sell
2020-03-20,XXXXXXX,14.45,14.46,11.91,12.52,-0.1377,buy
……

Fig. S6. The provided prompt of In-Context Learning (ICL).

[System Prompt]: *As an ordinary shareholder, you start with an initial equity of $1,000,000. Your task is to make trading decisions based on the provided historical dataset to maximize your equity.*

……

[Chain of thought :] *You should first provide a 'CoT' (Chain of Thought) reasoning process, detailing each step of your thought process leading to your decision. Please ensure that the answer is output solely in JSON format, without any additional text.*

The output should strictly follow the format: {'CoT': ['Step1: XXXX', 'Step2: XXXX', ...], 'decision': 'buy'/'sell'/'keep'}. Any deviation from this format is not acceptable.

Fig. S7. The provided prompt of Chain of Thought (CoT) Reasoning.

B.5. Agent Intelligence. In our experiment, we aimed to enhance agent intelligence by equipping them with various LLMs known for having distinct intelligence levels. We utilized GPT-3.5 as the baseline and increased the intelligence using Supervised Fine-Tuning (SFT) to expand their knowledge base, to effectively reduce the disposition effect.

- **GPT-3.5 Baseline:** The basic version of GPT-3.5 was employed as the control group in our experiments. This provided a reference for evaluating the impact of enhanced intelligence on agent behavior.

- **Enhanced GPT-3.5 using SFT:** We further improved GPT-3.5 through Supervised Fine-Tuning (SFT). This process involved fine-tuning the model with practical human trading experiences, aligning the agent's responses more closely with real-world trading behaviors.

The SFT process consists of the following steps:

1. Introduction to Fine-Tuning. Fine-tuning leverages pre-trained models by training them further on task-specific data. This process allows for higher quality results, the ability to train on more examples, token savings, and lower latency requests. Fine-tuning is an improvement over few-shot learning, which uses a few examples in a prompt to guide the model's behavior. Fine-tuning allows for training on large volume of examples, leading to better performance on a wide range of tasks.

2. Preparing and Uploading Training Data. The first step in fine-tuning is to prepare the training data. The agents were trained using historical stock trading data, encapsulated in the K-line format (including opening price, highest price, lowest price, and closing price) and the agents' previous decision records. This data was anonymized to prevent agents from

```json
{
  "messages": [
    {
      "role": "system",
      "content": "As an ordinary shareholder, your task is to make trading decisions based on the provided historical dataset to maximize your equity. The dataset includes the following columns: [\"date\", \"code\", \"open\", \"high\", \"low\", \"close\"]. The \"date\" column records the stock trading date. The \"code\" column represents the code of the current stock. The K-line of the stock consists of \"open\" (open price), \"high\" (highest price), \"low\" (lowest price) and \"close\" (close price). The \"change\" column records the change in stock's close price.\n\nYou have to select your decision between ['buy', 'sell', 'keep']. If you choose 'buy', you will use 20% of your cash to buy the stock. If you choose 'sell', on the contrary, you will sell 20% of the shares you hold and receive the corresponding cash. If you choose 'keep', you conduct neither buying nor selling, not making any new trades.\n\nYour task is to provide a decision to estimate the action for the next trading period, thoughtfully considering both the K-line data of the current period and the comprehensive historical data from past periods."
    },
    {
      "role": "user",
      "content": {
        "data": [
          {"date": "2015-01-05", "code": "sh600176", "open": 15.45, "high": 16.35, "low": 15.15, "close": 15.79},
          {"date": "2015-01-06", "code": "sh600176", "open": 15.70, "high": 15.70, "low": 15.20, "close": 15.55},
          {"date": "2015-01-07", "code": "sh600176", "open": 15.31, "high": 15.98, "low": 15.12, "close": 15.17},
          {"date": "2015-01-08", "code": "sh600176", "open": 15.10, "high": 15.24, "low": 14.56, "close": 14.60},
          {"date": "2015-01-09", "code": "sh600176", "open": 14.50, "high": 14.90, "low": 14.30, "close": 14.37},
          {"date": "2015-01-12", "code": "sh600176", "open": 14.43, "high": 14.78, "low": 14.38, "close": 14.49},
          {"date": "2015-01-13", "code": "sh600176", "open": 14.50, "high": 14.76, "low": 14.32, "close": 14.50},
          {"date": "2015-01-14", "code": "sh600176", "open": 14.55, "high": 15.35, "low": 14.55, "close": 15.33},
          {"date": "2015-01-15", "code": "sh600176", "open": 15.33, "high": 15.34, "low": 14.88, "close": 15.31},
          {"date": "2015-01-16", "code": "sh600176", "open": 15.29, "high": 15.31, "low": 15.00, "close": 15.16}
        ]
      }
    },
    {
      "role": "assistant",
      "content": "decision: buy"
    }
  ]
}
```


Fig. S8. The data preparation of Supervised Fine-Tuning (SFT) in our experiment. The training data consists of historical stock trading data organized in JSON format, including daily trading information and previous decision records.

recognizing the specific stocks from their training data. The dataset was organized into JSON format, where each entry represented a day's trading data for a specific stock, as is shown in Figure S8.

3. Training the Fine-Tuned Model. To train the model, we used OpenAI's API for fine-tuning. The process involved the following steps:

- a. Uploading the Training Data: First, we uploaded the prepared JSON training data to OpenAI's servers.
- b. Initiating the Fine-Tuning Process: We called OpenAI's fine-tuning endpoint with the appropriate parameters, specifying the base model (e.g., GPT-3.5) and the uploaded training data.

4. Evaluating the Fine-Tuned Model. After the fine-tuning process, we evaluated the model by testing it on a separate testing dataset, which contained similar structured data but was not used during training. The model's performance was assessed based on its ability to make rational trading decisions and reduce the disposition effect. Metrics for evaluation included:

- Reduction in Disposition Effect: Measured by the proportion of realized gains and losses.
- Investment Returns: Overall performance in terms of returns generated from trading decisions.

5. Iterative Refinement. Based on the evaluation results, the fine-tuning process was iteratively refined:

- Revisiting Training Data: Adding more examples or refining existing ones to better capture the desired behavior.
- Adjusting Training Parameters: Modifying learning rates, batch sizes, or other hyperparameters to optimize performance.

This iterative approach ensured continuous improvement of the model's decision-making capabilities.

The fine-tuning process significantly improved the intelligence and rationality of the GPT-based agents. By training on specific historical trading data and aligning responses with human trading behaviors, the agents exhibited reduced biases and enhanced decision-making quality. This method showcases the potential of SFT in tailoring LLMs for specialized applications, particularly in high-stakes financial environments.

C. The vast research field of machine behaviorism. In many decision-making scenarios where agents are delegated tasks on behalf of humans, Machine Behaviorism can be leveraged to improve these agents' decisions, e.g., by addressing their decision biases. For instance, LLMbased agents have been used in finance to assist or replace human investors in stock trading, to maximize returns while controlling risks. Utilizing Machine Behaviorism informed by Behavioral Finance can assist in discerning if these agents mirror the behaviors of human traders, who frequently fall prey to behavioral biases like loss aversion in trading. Building upon this, Machine Behaviorism can subsequently guide the design of strategies to alleviate such biases. In organizational management, LLM-based agents can be employed to make organizational decisions such as hiring. Machine Behaviorism can help identify and avoid the commonly observed manager overconfidence in corporate decision-making. In education, agents have been deployed to design personalized teaching strategies, where subjective judgments from LLMs may be involved. Machine Behaviorism can help identify potential irrational judgments and minimize the subsequent negative effects. We have provided an extensive list of related domains in Table S3 for further reference.

Table S3. Research Fields for Machine Behaviorism Studies

Fields	Description	Relation with Machine Behaviorism	Examples
Behavioral Finance	Behavioral Finance explores how psychological factors influence financial decision-making. It investigates deviations from rational behavior and traditional economic models, taking into account cognitive biases, emotional influences, and social dynamics.	Machine Behaviorism can align with this field by examining how LLMs' decision-making behavior in financial scenarios mirrors or diverges from human biases and heuristics. By studying the disposition effect, as mentioned in the introduction, Machine Behaviorism can contribute to understanding whether LLMs exhibit a tendency to hold on to losing investments longer than they should and vice versa, similar to human investors. This connection allows for insights into the application of behavioral finance principles to optimize LLMs' decision-making in financial contexts.	Disposition effect, Overconfidence, Herd Behavior, Loss Aversion, Anchoring Bias, Mental Accounting, Prospect Theory, Framing Effect, Availability Bias
Behavioral Management	Behavioral management focuses on understanding and influencing human behavior within organizational and managerial contexts. It considers how cognitive, social, and emotional factors impact decision-making, team dynamics, and performance.	Machine Behaviorism can draw parallels to this field by studying LLMs' behaviors in complex decision-making scenarios within organizations. By examining how LLMs respond to various managerial strategies, team dynamics, and organizational structures, Machine Behaviorism can provide insights into optimizing LLMs' decision-making processes in real-world management contexts. This connection allows for the application of behavioral management principles to enhance LLMs' performance and their integration into organizational settings.	Leadership Styles, Communication Patterns, Decision-Making Biases, Organizational Culture, Team Dynamics, Employee Engagement, Conflict Resolution, Change Management, Performance Appraisal, Work-Life Balance, Organizational Behavior, Talent Management
Education and Learning Sciences	Education and learning sciences involve studying how people learn and designing effective educational practices. It explores teaching methodologies, educational technologies, cognitive processes, and social factors that influence learning outcomes, aiming to improve educational experiences and outcomes.	Machine Behaviorism can investigate LLMs' behavior and decision-making in educational settings. This involves understanding how LLMs support personalized learning, adaptive tutoring, and educational content recommendation. By aligning with education and learning sciences, researchers can optimize LLMs' decision-making to enhance educational outcomes.	Educational Psychology, Curriculum Development, Instructional Design, Learning Theories, Educational Technology, Assessment and Evaluation, Classroom Management, Educational Policy, Online Learning, Teacher Professional Development
Healthcare Management	Healthcare management involves overseeing and organizing the operations of healthcare facilities and organizations. It focuses on optimizing healthcare delivery, improving patient outcomes, and ensuring efficient resource allocation.	Machine Behaviorism can explore how LLMs make decisions in healthcare management, such as treatment recommendations, resource allocation, and patient prioritization. By studying LLMs' behavior in healthcare contexts, researchers can optimize their decision-making processes to improve patient outcomes and resource utilization.	Strategic Planning, Healthcare Policy, Quality Improvement, Healthcare Information Systems, Healthcare Leadership, Patient Safety, Healthcare Finance, Healthcare Operations, Healthcare Analytics, Healthcare Marketing
Marketing and Consumer Behavior	Marketing and consumer behavior involve studying how individuals make choices and decisions as consumers. It examines the psychological, social, and cultural factors that influence consumer behavior, helping businesses understand and predict consumer preferences to develop effective marketing strategies.	Machine Behaviorism can investigate LLMs' behavior in marketing and consumer decision-making contexts. This includes understanding how LLMs process and respond to marketing stimuli, predict consumer preferences and make personalized recommendations. These insights can be leveraged to enhance marketing strategies and improve customer satisfaction.	Consumer Psychology, Brand Management, Consumer Decision-Making, Market Research, Advertising and Promotion, Consumer Segmentation, Digital Marketing, Social Media Marketing, Consumer Satisfaction, Product Innovation
Human Resources and Talent Management	Human resources and talent management focus on the recruitment, development, and retention of employees within organizations. It includes activities such as talent acquisition, training and development, performance management, and employee relations to maximize workforce productivity and create a positive work environment.	Machine Behaviorism can investigate LLMs' behavior and decision-making in human resources and talent management. This involves understanding how LLMs assess job applicants, make hiring decisions, and provide career development recommendations. By aligning with human resources and talent management practices, researchers can optimize LLMs' decision-making to support effective talent acquisition and management.	Recruitment and Selection, Performance Management, Talent Acquisition, Training and Development, Employee Engagement, Compensation and Benefits, Leadership Development, Succession Planning, Diversity and Inclusion, HR Analytics
...	...	...	...