exposure.

```
System Prompt:
# memory IDs
short_memory_id_desc = "ID of short-term
 information."
mid_memory_id_desc = "ID of mid-term
information."
long_memory_id_desc = "ID of long-term
information."
reflection_memory_id_desc = "ID of
reflective-period information."
train_memory_id_extract_prompt = "Select
 and store the most investment-relevant
information from major sources (e.g.,
ARK, Two Sigma, Bridgewater Associates)
into the {memory_layer} memory."
test_memory_id_extract_prompt = "
Retrieve the most relevant information
from the {memory_layer} memory for the
current investment decision."

# trading summary
train_trade_reason_summary = "Based on a
 professional trader's advice, explain
why the trader would make such a
decision given the provided information
."
test_trade_reason_summary = "Based on
the text information and summarized
price trends, explain the reason for
your investment decision."
test_invest_action_choice = "Based on
the information, choose one of the
following actions: buy, sell, or hold."

# investment info
train_investment_info_prefix = (
    "The current date is {cur_date}. The
     observed market facts are as
    follows: "
    "For {symbol}, the price difference
    between the next and current trading
     day is {cur_record_t1}; "
    "the 7-day difference is {
    cur_record_t7}; "
    "the 30-day difference is {
    cur_record_t30}. "
    "Your decision return is {reward}.\n
    \n"
)
train_reward_explanation =  """Reward
reflects the quality of your past
decision:
        - **Positive**: Good decision;
        higher means better alignment
        with market.
        - **Negative **: - **Negative**:
         A weaker decision. The more
        negative the value, the worse
        the outcome - may caused by
        misreading available data.
        Use reward **only for reflection
        **, not for future predictions.
"""
test_investment_info_prefix = "The stock
 under analysis is {symbol}, and the
current date is {cur_date}."

# sentiment & momentum explanation
test_sentiment_explanation = """For
example, positive news about a company
typically boosts market confidence,
stimulates buying, and drives up the
stock price;
Negative news, on the other hand,
weakens confidence, triggers selling
pressure, and causes the stock price to
fall.
Industry or competitor dynamics may also
 indirectly affect the target company's
performance.
The sentiment score (positive, neutral,
negative) reflects the proportion of the
 text in each of the three sentiment
categories (summing to 1).
It can be combined with the "Importance"
 and "Timeliness" metrics to assess the
market impact and validity of the
information.
In addition, you need to combine the
output of the previous Direction
Decision Agent (i.e., the overall
strategic description of this
transaction) as the strategic basis for
quantitative decisions.
"""
test_momentum_explanation = """The
following summarizes recent price
movements, i.e., momentum.
Momentum reflects the idea that stocks
performing strongly in the short term
often continue rising,
while weak performers are more likely to
 keep declining.
"""

# training phase prompt
train_prompt = """Please complete the
following two tasks based on the
investment information below:
Important: Do NOT use any future price
differences (T+1, T+7, T+30) in your
reasoning. These are unavailable in real
-time trading. Any output referencing
them will be considered invalid.
1. Investment Amount and Risk Decision:
You already know the directional
decision (buy/sell/hold) made by the
Direction Decision Agent in the previous
 stage.
Based on this, determine the **specific
order quantity**(integer) and ensure
that the transaction volume does not
exceed the maximum limit {maxcvar}
recommended by the risk control module.
You must consider the following factors:
- Information in each memory layer (
short-term, medium-term, long-term, and
reflection period);
- Historical momentum and price
volatility;
- The sentiment, importance, and
timeliness of news or financial reports;
- Current account holdings and overall
risk exposure;
- Trading strategy determined in the
previous phase.
Please briefly explain your quantity
decision logic and indicate the memory
```

```
indexes supporting this decision.
2. Decision Reflection and Analysis:
The system will calculate a reward based
 on the order quantity and corresponding
 return.
- If the reward is negative, please
explain any market signals or risk
factors you may have misjudged;
- If the reward is positive, please
summarize the core rationale that led to
 your correct decision.
${investment_info}

Your output should strictly adhere to
the following JSON format and not
include any other content:
{
"order_size": integer (range 1 to {
maxcvar}),
"summary_reason": "Please enter your
quantity and risk decision logic here",
"short_memory_index": [list of integers
],
"middle_memory_index": [list of integers
],
"long_memory_index": [list of integers],
"reflection_memory_index": [list of
integers],
"reflection_analysis": "Please fill in
your reflection description here."
}
"""

# Testing phase prompt
test_prompt = """Based on the following
information, please determine the **
order quantity** for the current trade.
You know the directional decision (buy/
sell/hold). Please specify the specific
order quantity based on the risk
exposure and CVaR constraint (maximum
order quantity {maxcvar}).
You must consider:
- Memory information at each level (
short-term, medium-term, long-term,
reflection period);
- Momentum trend, sentiment, information
 importance, and timeliness;
- Current account holdings and overall
risk;
- Trading strategy for the previous
directional decision.
Please output a specific order quantity
(integer, not exceeding {maxcvar}) and
indicate the information index that
supports your judgment.

${investment_info}
${gr.complete_json_suffix_v2} }
"""
```

B Formulas of Classic Financial Metrics

To evaluate the risk-return characteristics of trading strategies, we summarize the formal definitions of commonly used financial evaluation metrics, including risk-adjusted return and downside risk measures, which are used throughout our experiments.

B.1 Definitions of Evaluation Metrics

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p}, \tag{6}$$

where R_p is the average return of the portfolio, R_f is the risk-free rate, and σ_p is the standard deviation of portfolio returns. A higher sharpe ratio indicates more efficient risk-adjusted performance.

$$\text{MDD} = \max_{t=1}^{N} \left(\frac{P^t - P^t_{trough}}{P^t} \right) \tag{7}$$

where t denotes the index of the trading day, P^t is the account value (the market value of the current stock position) at day t, and P^t_{trough} is the lowest future account value observed after day t. A smaller MDD reflects stronger downside protection and greater robustness of the strategy.

B.2 Conditional Value at Risk (CVaR)

Let the profit and loss over a trading horizon be denoted by PnL. The Value at Risk (VaR) at a confidence level α represents the maximum potential loss not exceeded with probability α, formally defined as:

$$\text{VaR}_\alpha(PnL) = \inf\{l \in \mathbb{R} : P(PnL \leq l) \geq \alpha\}.$$

The Conditional Value at Risk (CVaR) measures the expected loss that occurs beyond the VaR threshold, providing a more comprehensive view of downside risk:

$$\text{CVaR}_\alpha(PnL) = \mathbb{E}[PnL \mid PnL \leq \text{VaR}_\alpha(PnL)].$$

Importantly, during trading, CVaR is computed online using a 20-trading-day rolling window of past realized returns and updated daily, ensuring that no forward-looking or test-period information is used in position sizing. A smaller CVaR indicates stronger downside protection and more effective risk control.

B.3 Calmar Ratio

The Calmar Ratio evaluates the trade-off between return and maximum drawdown. It is defined as:

$$\text{Calmar Ratio} = \frac{R_{\text{annual}}}{|\text{MDD}|}$$

where R_{annual} denotes the annualized return, and MDD represents the maximum drawdown during the same period. A higher Calmar Ratio indicates a better risk-adjusted performance.

C Experimental Setup

C.1 Position Awareness and Decision Structures

As summarized in Tab. 3, most existing LLM-based trading agents formulate decision-making as a discrete action selection problem, typically restricted to *buy*, *sell*, or *hold*, with fixed or implicit trade sizes. FinGPT adopts a single-agent architecture with predetermined position sizes, while FinMem leverages memory mechanisms to guide reasoning without explicitly modeling position magnitude or exposure. Although FinCon and FinAgent employ multi-stage or debate-based reasoning pipelines, their final outputs remain direction-only decisions without explicit position sizing. However, trading volume is a fundamental component of position awareness. Without the ability to adjust exposure magnitude, agents operating under direction-only actions lack fine-grained control over risk. For instance, when holding a large long position, an agent may detect increased downside risk but can only respond through unit-based actions, resulting in delayed or insufficient risk mitigation. In contrast, FinPos explicitly models position evolution through a two-stage decision structure that decouples directional reasoning from quantitative position sizing. By incorporating CVaR-based position control and reflection guided by multi-timescale rewards, FinPos enables risk-aware adjustments to both trading direction and exposure magnitude, leading to more coherent and realistic position management.

C.2 LLM-based Baselines Trading Agents

We compare FinPos against a representative set of state-of-the-art LLM-based trading agents that differ in architectural design, information processing pipelines, and decision structures, as summarized in Tab 3. We strictly follow the inference settings reported in their original papers.

C.3 Deep Reinforcement Learning Baselines

In addition to LLM-based agents, we benchmark FinPos against classical deep reinforcement learning (DRL) agents, including A2C, PPO, and DQN. All DRL baselines are implemented using the FinRL framework (Liu et al., 2020). These agents operate solely on numerical features derived from market prices and technical indicators, without access to textual or semantic information. The key hyperparameters for each RL agent are listed in Table 4.

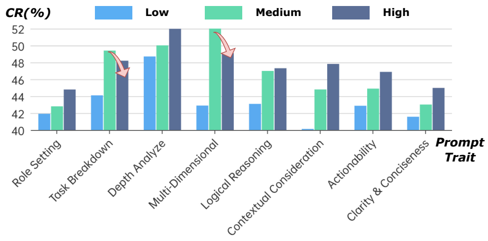

Figure 5: Prompt ablation across eight characteristics and three emphasis levels.

D More Experiments

D.1 Sensitivity to LLM Sampling Hyperparameters

FinPOS does not train or fine-tune large language models; therefore, its performance is independent of random initialization seeds typically used in neural network training. However, to address reviewers' concerns regarding stochasticity introduced during LLM inference, we conduct a sensitivity analysis by varying key decoding hyperparameters, including temperature and top_p.

Specifically, we evaluate FinPOS under multiple sampling configurations on TSLA while keeping all other components unchanged. As shown in Tab. 5, the overall performance remains stable across different settings, with only minor variations in cumulative return (CR), Sharpe ratio (SR), and maximum drawdown (MDD). These results indicate that FinPOS is not overly sensitive to reasonable changes in LLM sampling strategies, and its trading behavior is robust under inference-time stochasticity.

D.2 More Stock Trading Result Graphs

Similarly, for AAPL, the supplementary plots (Fig. 6) provide complementary evidence from profitability and risk control. FinPos again occupies the upper-right region in the CR% vs. risk-control space, indicating that its returns are achieved without compromising drawdown management. The time-series and exposure-risk overlays highlight that, during high-volatility periods, non-position-aware methods experience sharp spikes in exposure and subsequent losses, whereas FinPos effectively anticipates and mitigates these risks. Overall, these results reinforce that position