



# WebCryptoAgent: Agentic Crypto Trading with Web Informatics

Ali Kurban<sup>1\*</sup> Wei Luo<sup>2\*</sup> Liangyu Zuo<sup>1\*</sup> Zeyu Zhang<sup>3†</sup>

Renda Han<sup>4</sup> Zhaolu Kang<sup>3</sup> Hao Tang<sup>3‡</sup>

<sup>1</sup>AI Geeks <sup>2</sup>XJTU <sup>3</sup>Peking University <sup>4</sup>QTNU

\*Equal contribution. †Project lead. ‡Corresponding author: bjdx tanghao@gmail.com.

## Abstract

Cryptocurrency trading increasingly depends on timely integration of heterogeneous web information and market microstructure signals to support short-horizon decision making under extreme volatility. However, existing trading systems struggle to jointly reason over noisy multi-source web evidence while maintaining robustness to rapid price shocks at sub-second timescales. The first challenge lies in synthesizing unstructured web content, social sentiment, and structured OHLCV signals into coherent and interpretable trading decisions without amplifying spurious correlations, while the second challenge concerns risk control, as slow deliberative reasoning pipelines are ill-suited for handling abrupt market shocks that require immediate defensive responses. To address these challenges, we propose WEBCRYPTOAGENT, an agentic trading framework that decomposes web-informed decision making into modality-specific agents and consolidates their outputs into a unified evidence document for confidence-calibrated reasoning. We further introduce a decoupled control architecture that separates strategic hourly reasoning from a real-time second-level risk model, enabling fast shock detection and protective intervention independent of the trading loop. Extensive experiments on real-world cryptocurrency markets demonstrate that WEBCRYPTOAGENT improves trading stability, reduces spurious activity, and enhances tail-risk handling compared to existing baselines. Code will be available at <https://github.com/AIGeeksGroup/WebCryptoAgent>.

## 1 Introduction

In recent years, the rapid development of large language models (LLMs) has catalyzed a new paradigm of *agentic trading systems* (Shi et al., 2025; Zhang et al., 2025b; Lin et al., 2025; Ge et al., 2025; Zhang et al., 2025a), where autonomous

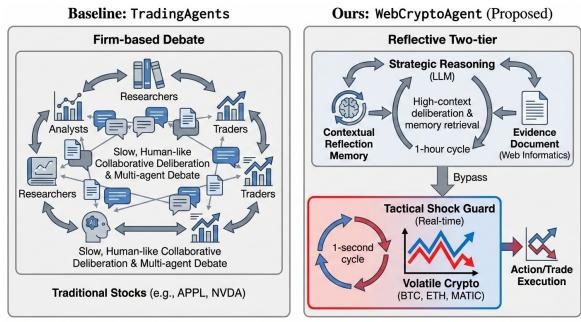


Figure 1: Structural comparison between the horizontal firm-based debate model (TradingAgents) and our proposed vertical reflective two-tier architecture (WebCryptoAgent).

agents leverage textual and numerical information to make financial decisions. With the global expansion of the cryptocurrency market, characterized by extreme volatility and round-the-clock trading, the demand for intelligent trading assistants has intensified. These agents are designed not only to process heterogeneous data sources—such as news, social media sentiment, and historical market data—but also to reason and act in dynamic environments. Early efforts in this direction include domain-adapted financial assistants such as PIXIU (FinMA) (Xie et al., 2023), FinGPT (Yang et al., 2023b), and Instruct-FinGPT (Zhang et al., 2023a), which fine-tune general-purpose LLMs on financial corpora to enhance domain sensitivity. Meanwhile, large-scale pretrained models such as BloombergGPT (Wu et al., 2023), XuanYuan 2.0 (Zhang et al., 2023b), and Fin-T5 (Lu et al., 2023) have demonstrated that hybrid domain-general corpora can achieve competitive reasoning capabilities while maintaining financial expertise. Collectively, these advances reveal the potential of language-based agents in financial contexts; however, most existing systems emphasize domain adaptation over agentic autonomy, leaving open challenges in continuous reasoning, context-

tual awareness, and decision self-correction.

Beyond static financial modeling, recent work has explored LLM-based agents that directly interact with live trading environments. GPT-3.5/4 and open-source alternatives such as Qwen (Bai et al., 2023) and Baichuan (Yang et al., 2023a) have been tested on sentiment-driven trading tasks (Lopez-Lira and Tang, 2023), showing promising profit margins even under naïve strategies. FinGPT-based pipelines (Kirtac and Germano, 2024) and reasoning-augmented frameworks like Wall-StreetLLM (Fatouros et al., 2024) extend this idea by incorporating news summarization and contextual interpretation. FinMem (Yu et al., 2023) and TradingGPT (Li et al., 2023) introduce memory-enhanced and multi-agent debate mechanisms that reduce hallucination and improve backtesting performance, while hybrid RL-reflection designs such as SEP (Koa et al., 2024) and PPO-augmented approaches (Ding et al., 2023) aim to optimize long-term trading returns. The latest evolution, TradingAgents (Xiao et al., 2025), simulates an entire virtual trading firm where specialized LLM agents (analysts, researchers, traders, and risk managers) collaborate to achieve superior Sharpe ratios and drawdown control.

As illustrated in Figure 1, while TradingAgents relies on a horizontal organizational structure with multiple specialized roles engaging in deliberative debate, our proposed WEBCRYPTOAGENT introduces a vertical, two-tier architecture specifically designed for the high-velocity requirements of cryptocurrency markets. This separation of strategic reasoning and tactical execution allows for complex decision-making without compromising the reaction speed necessary for crypto assets. Nevertheless, despite these innovations, two key challenges remain prevalent across agentic trading systems: (i) limited self-correction capability, as current agents rarely utilize retrieved historical reasoning traces for reflective improvement; and (ii) insufficient or underdeveloped risk management mechanisms, leading to unstable performance in volatile crypto markets.

Motivated by these observations, we aim to address the aforementioned limitations by introducing a novel agentic architecture that integrates *contextual reflection* and *structured risk management* into a unified pipeline. Our motivation stems from two core needs: first, enabling trading agents to autonomously reflect on past reasoning trajectories, refine decision policies, and adapt to evolving mar-

ket conditions; and second, embedding robust risk assessment and control procedures into the decision loop to ensure both profitability and stability in high-risk environments such as cryptocurrency trading. By combining reflective reasoning with dynamic risk calibration, our approach aspires to move beyond single-step prediction toward sustained, self-corrective intelligence.

To realize these goals, we propose **WebCryptoAgent**, an end-to-end web-enabled crypto trading agent designed to perform autonomous trading, self-reflection, and adaptive risk management. Specifically, we design a *contextual reflection module* that leverages retrieved decision histories and environmental cues to iteratively refine policy reasoning. In parallel, we introduce a *hierarchical risk management framework* that evaluates portfolio exposure, volatility dynamics, and model uncertainty to adjust position sizes and safeguard returns. Furthermore, we conduct comprehensive experiments across multiple benchmark datasets and real-world simulation environments, demonstrating that WebCryptoAgent consistently outperforms existing baselines in profitability, stability, and drawdown control.

In summary, our main contributions can be outlined as follows:

- **WebCryptoAgent Framework:** We introduce an agentic trading pipeline that integrates reasoning, self-reflection, and execution for cryptocurrency markets. The proposed contextual reflection module enables dynamic policy refinement based on historical feedback.
- **Hierarchical Risk Management:** We design a multi-level risk assessment mechanism (as shown in the “Tactical Shock Guard” of Figure 1) that quantifies uncertainty, manages portfolio exposure, and prevents excessive drawdowns in high-volatility environments.
- **Comprehensive Evaluation:** Through extensive experiments on synthetic and real-world crypto datasets, we show that WebCryptoAgent achieves superior performance in cumulative return, Sharpe ratio, and risk-adjusted metrics compared to state-of-the-art agentic traders.

## 2 Related Work

**Agentic Financial Assistants** Domain-adapted language models for finance are generally obtained

either through fine-tuning general-purpose LLM agents or pretraining from scratch on financial corpora. Fine-tuning enhances a model’s domain sensitivity while retaining its general reasoning ability. Examples include PIXIU (FinMA) (Xie et al., 2023), which fine-tunes LLaMA on 136K finance-related instructions; FinGPT (Yang et al., 2023b), which applies LoRA to models such as LLaMA and ChatGLM with roughly 50K finance-specific samples; and Instruct-FinGPT (Zhang et al., 2023a), which incorporates 10K sentiment-oriented instruction datasets. These specialized variants significantly outperform untuned models like BLOOM or OPT (Zhang et al., 2022) on classification benchmarks, sometimes even surpassing BloombergGPT (Wu et al., 2023), though they typically fall short of GPT-4 on open-ended reasoning tasks. Another line of work trains finance-specific LLM agents entirely from scratch. BloombergGPT (Wu et al., 2023), Xu-anYuan 2.0 (Zhang et al., 2023b), and Fin-T5 (Lu et al., 2023) exemplify this trend, using mixtures of general text and finance-domain corpora. BloombergGPT, in particular, demonstrates superior performance on market sentiment classification while remaining competitive on general NLP tasks. Collectively, these studies highlight the value of high-quality domain corpora in adapting LLM agents to financial contexts.

**Agentic Traders** LLM agents have also been positioned as autonomous trading agents capable of ingesting heterogeneous market signals and issuing trading actions. News-driven agents rely on textual market updates, financial reports, and sentiment analysis. Both closed-source models (e.g., GPT-3.5/4) and open-source LLMs (e.g., Qwen (Bai et al., 2023), Baichuan (Yang et al., 2023a)) have been tested on stock-news sentiment prediction (Lopez-Lira and Tang, 2023), with even simple sentiment-based strategies producing nontrivial returns. Further improvements arise from fine-tuned variants such as FinGPT or OPT-based financial sentiment models (Kirtac and Germano, 2024), as well as reasoning-augmented pipelines that summarize and interpret evolving news streams (Fatouros et al., 2024). Beyond direct sentiment mapping, reasoning-enhanced frameworks such as FinMem (Yu et al., 2023) integrate layered memory to contextualize decisions, while TradingGPT (Li et al., 2023) employs multi-agent debates with distinct agent profiles. Such

designs reduce hallucinations and yield superior backtest metrics. Reinforcement learning methods further refine trading performance by optimizing outputs against simulated returns; SEP (Koa et al., 2024) exemplifies this reflection-RL hybrid, while PPO-based approaches (Ding et al., 2023) integrate LLM-generated embeddings into conventional RL pipelines. Recent work such as TradingAgents (Xiao et al., 2025) extends this direction by simulating a realistic trading firm environment with multiple specialized LLM agents (analysts, researchers, traders, and risk managers), achieving superior cumulative returns, Sharpe ratios, and drawdown control compared to traditional baselines.

**Agentic Alpha Miners** Instead of executing trades, LLM agents can also contribute by generating *alpha factors*, i.e., novel predictive signals for trading. QuantAgent (Wang et al., 2024) demonstrates a nested loop design in which a writer agent proposes scripts for factor generation, a judge agent provides feedback, and outer-loop evaluation against market data closes the feedback cycle. AlphaGPT (Wang et al., 2023) extends this to a human-in-the-loop paradigm where experts collaborate with agents to iteratively refine alpha strategies. These systems underscore the potential of LLM-driven alpha discovery, highlighting their ability to automate exploratory research and accelerate quantitative investment strategy design.

## 3 Method

### 3.1 Overview

Our approach integrates large language model (LLM) reasoning with systematic trading execution through three interdependent components: (1) an agentic reasoning workflow for multi-modal market understanding, (2) a contextual reflection mechanism inspired by Reflexion (Liu et al., 2025), and (3) a regime-aware risk management layer ensuring capital efficiency and adaptive exposure.

### 3.2 Agent Workflow

The proposed trading agent operates as a reasoning-execution pipeline that transforms heterogeneous market inputs into structured trading decisions. At each decision epoch  $t$ , the agent constructs a market snapshot  $\mathcal{D}_t = \{O_t, I_t, N_t, R_t\}$ , where  $O_t$  denotes multi-scale OHLCV data (15-minute and 1-hour bars), and  $I_t$  represents the indi-

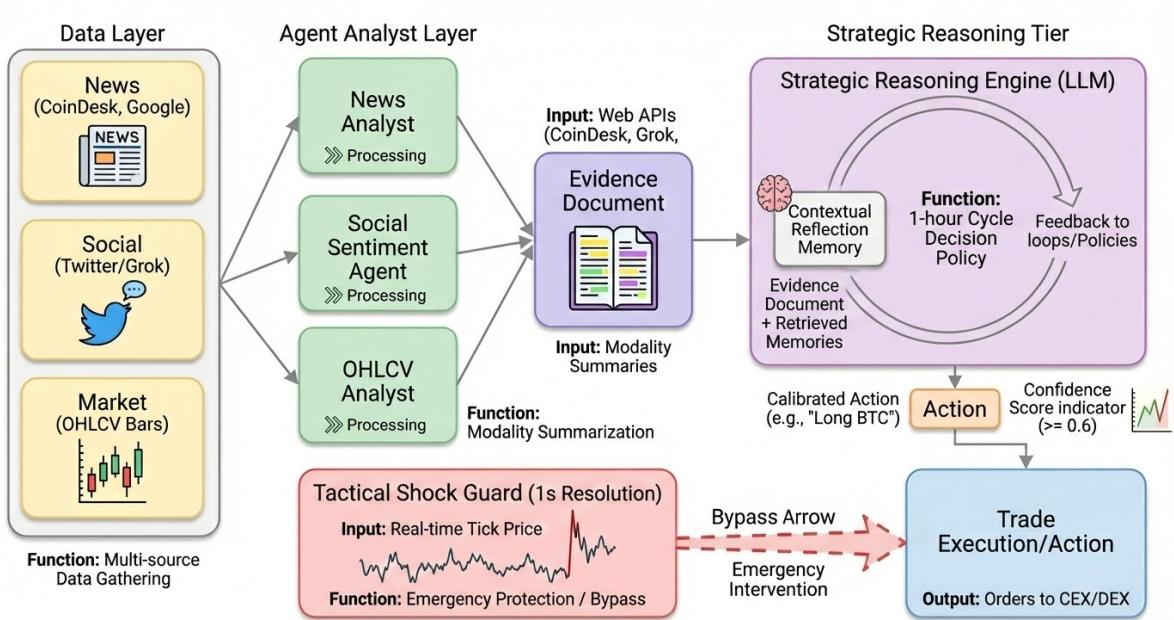


Figure 2: Overview of the WebCryptoAgent architecture. The framework employs a two-tier decision-making process: (1) a Strategic Tier where specialized agents aggregate multi-modal data (News, Social, Market) into an Evidence Document for LLM-based reasoning with contextual memory reflection; and (2) a Tactical Tier (Shock Guard) that monitors high-frequency tick data to trigger low-latency emergency bypasses. Final actions are dispatched to the Execution Layer for CEX/DEX deployment.

cator set

$$\mathcal{I} := \left\{ \begin{array}{l} \text{EMA}_{21}, \text{EMA}_{50}, \text{EMA}_{200}, \text{RSI}_{14}, \\ \text{MACD}, \text{ATR}_{14}, \text{BB}, \text{VWAP}, \text{PDH}, \text{PDL} \end{array} \right\}.$$

encodes the current regime snapshot describing macro sentiment, volatility state, and liquidity depth.

Before decision generation, the agent retrieves contextually similar historical episodes from the experience memory  $\mathcal{B}$  through a top- $K$  similarity search:

$$\mathcal{E}_t = \text{TopK}(\mathcal{B}, \mathcal{D}_t, K),$$

where similarity is defined by a weighted combination of cosine distance in embedding space and exact regime matching. This retrieved context provides exemplars of how analogous market states evolved in the past.

The reasoning model  $f_{\text{LLM}}(\cdot)$ , implemented using a large-language-model backbone (e.g., GPT-5 or Gemini-2.0-Flash-Thinking), processes both the current context and retrieved experiences to generate a structured decision tuple:

$$\mathcal{A}_t = f_{\text{LLM}}(\mathcal{D}_t, \mathcal{E}_t, R_t) = \{b_t, c_t, m_t, \rho_t\},$$

where  $b_t \in \{\text{LONG}, \text{FLAT}\}$  is the directional bias,  $c_t \in [0, 1]$  is the confidence score,  $m_t$  is the

expected move in basis points, and  $\rho_t$  is the generated rationale explaining the recognized pattern.

To avoid unstable oscillations in trade direction, we employ a regime-dependent hysteresis function:

$$b_t = \begin{cases} \text{LONG}, & c_t p_{\text{long}} \geq \theta_{\text{adopt}}(R_t), \\ & \text{trigger fired}, \\ \text{FLAT}, & c_t p_{\text{long}} < \theta_{\text{hold}}(R_t), \\ b_{t-1}, & \text{otherwise}. \end{cases}$$

Thresholds  $\theta_{\text{adopt}}$  and  $\theta_{\text{hold}}$  are adaptively calibrated by regime type, with  $\theta_{\text{adopt}} > \theta_{\text{hold}}$  to enforce persistence. A bias refresh occurs every eight hours, ensuring adaptation to new regimes while maintaining temporal stability.

The overall strategic decision process is summarized in Algorithm 1.

### 3.3 Contextual Reflection

Our self-improvement process is inspired by the *Reflexion* framework (Shinn et al., 2023) and extended through *Contextual Experience Replay (CER)* (Liu et al., 2025). This component allows the agent to iteratively evaluate its own decisions, identify sources of error, and incorporate refined insights back into its reasoning context.

---

**Algorithm 1:** Strategic Agent Decision Workflow

---

**Input:** Market data streams at time  $t$ , replay buffer  $\mathcal{B}$

**Output:** Trading action  $a_t$

Construct market snapshot  
 $\mathcal{D}_t = \{O_t, I_t, N_t, R_t\};$

Retrieve contextual experiences  
 $\mathcal{E}_t \leftarrow \text{TopK}(\mathcal{B}, \mathcal{D}_t, K);$

Generate decision tuple  $\mathcal{A}_t = \{b_t, c_t, m_t, \rho_t\} \leftarrow f_{\text{LLM}}(\mathcal{D}_t, \mathcal{E}_t, R_t);$

Update directional bias via regime-dependent hysteresis (Eq. (H));

**if**  $c_t \geq \theta_{\text{exec}}(R_t)$  **then**

- Execute trade with size determined by risk controller;

**else**

- Abstain from trading;

**return**  $a_t$

---

After each trade cycle, the agent observes realized outcomes at multiple horizons (4h, 8h, 24h, 7d) and forms a post-trade tuple:

$$\tau_t = (\mathcal{D}_t, \mathcal{A}_t, r_{h,t}),$$

where  $r_{h,t}$  is the realized net return (in basis points) after transaction costs. A reflection query is then composed for the LLM, containing the trade rationale  $\rho_t$ , the corresponding outcomes, and the regime context at entry. The LLM outputs a structured reflection:

$$\mathcal{F}_t := \begin{cases} \text{outcome\_label, attribution,} \\ \text{lesson, pattern\_validity} \end{cases}.$$

where the outcome label  $\in \{\text{WIN, LOSS, BREAK\_EVEN}\}$  and the attribution field explains which input signals (technical, news, regime) most contributed to performance.

Each reflection is distilled into a compressed experience embedding:

$$e_t := \text{Distill}(\tau_t) = \begin{cases} \text{context}_{\text{embed}}, R_t, \text{pattern,} \\ \text{cost, } \{r_h\}, \text{ lesson} \end{cases}.$$

which is stored in the replay buffer  $\mathcal{B}$  with exponential decay  $w(e_t, t') = \exp(-\frac{t'-t}{\lambda})$ , where  $\lambda$  is the half-life parameter (e.g., 30 days). During future inference cycles, the agent retrieves top- $K$  semantically similar experiences from  $\mathcal{B}$  and conditions

---

**Algorithm 2:** Contextual Reflection and Experience Replay (CER)

---

**Input:** Executed trade  $\mathcal{A}_t$ , realized returns  $\{r_{h,t}\}$

**Output:** Updated replay buffer  $\mathcal{B}$

Form post-trade tuple  $\tau_t = (\mathcal{D}_t, \mathcal{A}_t, r_{h,t});$

Query LLM for structured reflection  
 $\mathcal{F}_t \leftarrow \text{Reflect}(\tau_t);$

Distill compressed experience embedding  
 $e_t \leftarrow \text{Distill}(\tau_t, \mathcal{F}_t);$

Assign decay weight  $w(e_t) = \exp(-\frac{t'-t}{\lambda});$

Insert  $(e_t, w)$  into replay buffer  $\mathcal{B};$

**return**  $\mathcal{B}$

---

the next reasoning step on these reflections, effectively reusing its prior knowledge as contextual exemplars.

This closed reflection–replay loop enables continual self-improvement without retraining. Over time, the agent develops regime-specific priors on success likelihoods and adaptively modifies its decision thresholds based on accumulated experience. Empirically, this feedback mechanism increases consistency, reduces regime-specific overconfidence, and leads to smoother cumulative performance trajectories.

The contextual reflection and experience replay mechanism is formalized in Algorithm 2.

### 3.4 Risk Management

The risk management subsystem converts qualitative reasoning outputs into executable, quantitatively constrained trades. Position sizing is based on Average True Range (ATR)–derived volatility measures, where the stop-distance multiplier adapts to the current regime. In stable RISK-ON phases, positions are larger and stops tighter; during high-volatility or RISK-OFF periods, exposure is reduced and stops widened. Position sizes are further modulated using a fractional Kelly criterion, linking LLM confidence to statistical edge estimation while capping leverage through a conservative scaling factor. To ensure capital preservation, a hierarchy of risk controls is applied:

- Circuit breakers halt trading after predefined loss or drawdown thresholds.
- Portfolio exposure limits restrict concentration by asset and by total equity share.

---

**Algorithm 3:** Overall WebCryptoAgent Pipeline

---

**Input:** Streaming market data, web signals, replay buffer  $\mathcal{B}$

**Output:** Executed trades and updated memory

**while** *market is open* **do**

- Collect multi-source inputs (News, Social, OHLCV);
- // Strategic Tier (hourly cadence)
- if** *decision epoch reached* **then**
  - Generate trading action  $a_t$  via Strategic Agent (Algorithm 1);
- // Tactical Tier (second-level monitoring)
- Monitor high-frequency price stream for shock conditions;
- if** *shock detected* **then**
  - Override strategic action and trigger emergency protection;
- // Execution
- Submit final action to execution layer (CEX/DEX);
- // Post-trade reflection
- if** *trade cycle completed* **then**
  - Update replay buffer  $\mathcal{B}$  via Contextual Reflection (Algorithm 2);

**return** Executed trades and updated replay buffer  $\mathcal{B}$

---

- Time-based stops close positions automatically when liquidity deteriorates or when maximum holding durations are reached.

Before order submission, an explicit cost gate compares the model’s expected edge against cumulative frictional costs (liquidity-provider fee, impact, gas, spread, and MEV). Trades are executed only if the expected return exceeds the estimated cost margin.

The overall end-to-end operation of WebCryptoAgent is summarized in Algorithm 3.

## 4 Experiment

This section reports the empirical performance of four LLM-based trading agents on BTCUSDT, evaluated with and without memory. All results are produced under identical market data, execution rules, and decision schedules.

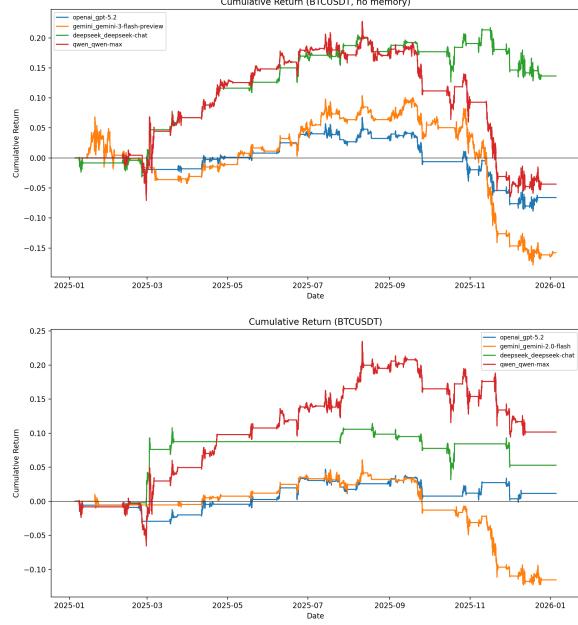


Figure 3: Cumulative return on BTCUSDT from 2025-01-05 to 2026-01-05. **Top:** no-memory configuration. **Bottom:** memory-enabled configuration. Each line corresponds to one LLM trading agent.

### 4.1 Experimental Setting

The experiment is conducted on BTCUSDT using 15-minute OHLCV data from 2025-01-05 to 2026-01-05, totaling 35,040 bars. Each model generates trading decisions at 122 fixed timestamps. Position sizing, transaction logic, and initial equity (\$10,000) are held constant across all runs.

Two configurations are evaluated:

- **Memory-enabled:** the model receives past decision–outcome information.
- **No-memory:** the model acts solely on the current market snapshot.

### 4.2 Cumulative Return

Figure 3 shows cumulative return curves for all models under both configurations.

The figure shows visible differences in return trajectories, drawdowns, and final equity between models and between memory settings.

### 4.3 BTCUSDT Results

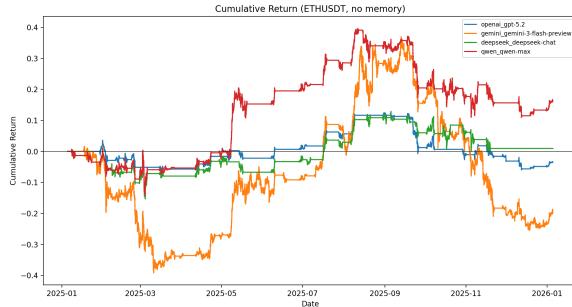
Table 1 reports summary statistics for all runs, including total return, drawdown, Sharpe ratio, and final equity.

Model	Memory	Trades	Win Rate	Total Ret.	Max DD	Sharpe	Equity End
GPT-5.2	On	23	0.61	0.0115	0.0464	0.21	10115
GPT-5.2	Off	27	0.56	-0.0659	0.1461	-0.67	9341
Gemini Flash	On	26	0.42	-0.1155	0.1732	-1.27	8845
Gemini Flash	Off	50	0.46	-0.1579	0.2553	-0.89	8421
DeepSeek Chat	On	10	0.50	0.0529	0.0742	0.76	10529
DeepSeek Chat	Off	29	0.66	0.1365	0.0728	1.19	11365
Qwen-Max	On	36	0.64	0.1016	0.1139	0.80	11016
Qwen-Max	Off	42	0.62	-0.0436	0.2378	-0.17	9564

Table 1: Performance metrics for BTCUSDT trading experiments with and without memory.



(a) Cumulative return for ETHUSDT with memory enabled.



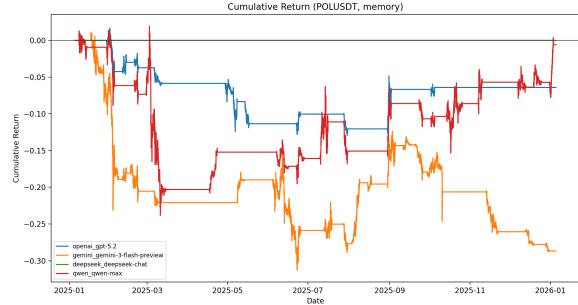
(b) Cumulative return for ETHUSDT without memory.

Figure 4: Equity curves for ETHUSDT trading with and without contextual memory.

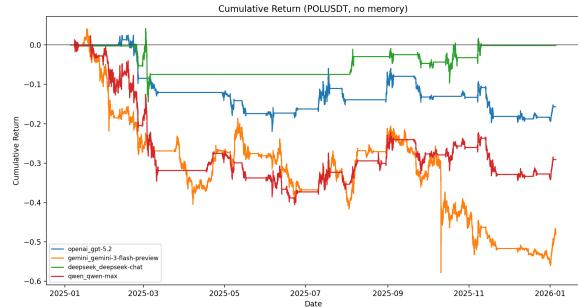
#### 4.4 ETHUSDT Results (Memory vs No-Memory)

We repeat the same evaluation protocol on ETHUSDT over 2025-01-05 to 2026-01-05 using 15-minute bars (35,040 bars) and 122 decision points. Table 2 summarizes performance for each model under memory-enabled and no-memory configurations.

Overall, the results differ across model backbones and between memory settings. GPT-5.2 shifts from a negative return without memory to a positive return with memory. DeepSeek-Chat changes from a small positive return without memory to a small negative return with memory. Qwen-Max shows the opposite pattern, achieving its strongest performance in the no-memory configuration, while memory reduces its return.



(a) Cumulative return for POLUSDT with memory enabled.



(b) Cumulative return for POLUSDT without memory.

Figure 5: Equity curves for POLUSDT trading with and without contextual memory.

#### 4.5 POLUSDT Results (Memory vs No-Memory)

We evaluate LLM-based trading agents on POLUSDT over the period 2025-01-05 to 2026-01-05 using 15-minute OHLCV data (35,040 bars) and 122 fixed decision points. All models operate under identical execution rules and initial equity (\$10,000).

Table 3 reports performance metrics for memory-enabled runs (top) and no-memory runs (bottom).

#### 4.6 ETHUSDT Equity Curves

Figure 4a and Figure 4b show cumulative returns for ETHUSDT with and without memory, evaluated over the same time period and decision points as the BTCUSDT experiments.

Provider	Model	Memory	Trades	Win Rate	Total Ret.	CAGR	Max DD	Sharpe	Avg Ret/Trade	Median Ret/Trade	Equity End	Fallbacks
openai	gpt-5.2	On	26	0.5769	0.0419	0.0420	0.0868	0.4334	0.00589	0.00272	10418.95	0
openai	gpt-5.2	Off	32	0.5313	-0.0355	-0.0356	0.1671	-0.2246	0.00089	0.00272	9645.15	0
gemini	gemini-3-flash-preview	On	50	0.3400	-0.2788	-0.2795	0.3425	-0.9348	-0.03063	-0.02946	7211.94	4
gemini	gemini-3-flash-preview	Off	50	0.4000	-0.1992	-0.1997	0.4557	-0.2579	-0.02705	-0.03442	8007.58	3
deepseek	deepseek-chat	On	10	0.4000	-0.0155	-0.0155	0.1057	-0.0973	-0.01216	-0.00754	9845.37	0
deepseek	deepseek-chat	Off	30	0.4667	0.00949	0.00951	0.1608	0.1468	0.00258	-0.00021	10094.87	0
qwen	qwen-max	On	34	0.5588	-0.0148	-0.0149	0.1678	0.0143	0.00066	0.00564	9851.64	0
qwen	qwen-max	Off	47	0.6383	0.1604	0.1609	0.2055	0.7325	0.02284	0.01820	11604.35	0

Table 2: ETHUSDT performance metrics for memory-enabled vs no-memory trading runs. All models are evaluated over the same period (2025-01-05 to 2026-01-05), using 15-minute bars (35,040) and 122 decision points.

Provider	Model	Memory	Trades	Win Rate	Total Ret.	CAGR	Max DD	Sharpe	Avg Ret/Trade	Median Ret/Trade	Equity End
openai	gpt-5.2	On	13	0.3846	-0.0641	-0.0643	0.1437	-0.6121	-0.03511	-0.02342	9358.58
gemini	gemini-3-flash-preview	On	34	0.2941	-0.2868	-0.2875	0.3201	-1.3348	-0.05613	-0.06332	7131.94
deepseek	deepseek-chat	On	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.00000	0.00000	10000.00
qwen	qwen-max	On	30	0.5333	-0.0060	-0.0060	0.2525	0.0949	-0.01545	0.00916	9940.49
openai	gpt-5.2	Off	21	0.4762	-0.1571	-0.1575	0.2389	-0.7855	-0.01778	-0.00765	8429.40
gemini	gemini-3-flash-preview	Off	52	0.2692	-0.4810	-0.4819	0.5951	-0.8395	-0.06527	-0.06772	5190.21
deepseek	deepseek-chat	Off	10	0.6000	-0.0016	-0.0016	0.1790	0.0693	0.00829	0.00951	9983.74
qwen	qwen-max	Off	39	0.3846	-0.2912	-0.2918	0.4191	-0.8093	-0.01801	-0.01538	7088.23

Table 3: POLUSDT performance metrics for LLM-based trading agents with memory enabled (top) and without memory (bottom). All runs use the same time period and decision points.

## 4.7 POLUSDT Equity Curves

Figure 5a and Figure 5b show cumulative returns for POLUSDT with and without memory under the same evaluation protocol.

## 4.8 Summary

Across all models, both the cumulative return curves and summary metrics show that enabling memory leads to different performance outcomes compared to no-memory execution. The magnitude and direction of these differences vary across model backbones.

## 5 Social Impact

WebCryptoAgent illustrates how reflective, memory-augmented agentic systems can contribute to real-world financial infrastructures operating under extreme volatility. By decoupling strategic reasoning from low-latency risk control, the framework addresses the mismatch between deliberative decision making and the rapid dynamics of digital markets, enabling more stable and interpretable behavior. This design reduces excessive trading activity and mitigates abrupt losses, which is particularly relevant for retail participants and smaller institutions. Beyond individual performance, the contextual reflection and experience replay mechanism promotes adaptive yet conservative decision making without continuous retraining, allowing the agent to internalize regime-dependent priors and selectively abstain under uncertainty. Such behavior supports smoother trading dynamics and helps limit the amplification of noise-driven market fluctuations.

At a broader level, WebCryptoAgent provides a practical blueprint for deploying large language models in high-stakes financial workflows where robustness and accountability are critical, and the two-tier reflective architecture may inform decision-support systems beyond cryptocurrency trading, including market monitoring and real-time economic analysis.

## 6 Potential Risks

The use of LLM-driven trading agents involves several practical considerations. Model behavior may vary under distribution shifts or rare market conditions, and reliance on external data sources can introduce noise or latency. In addition, reflection-based memory updates and automated execution require conservative configuration and ongoing monitoring. These considerations motivate cautious deployment and appropriate risk controls in real-world settings.

## 7 Conclusion

We presented WebCryptoAgent, a reflective agentic trading framework that integrates web-informed reasoning, contextual experience replay, and regime-aware risk control for short-horizon cryptocurrency trading. By decoupling strategic LLM-based reasoning from low-latency tactical protection, the proposed two-tier architecture enables robust decision making under extreme market volatility. Extensive experiments demonstrate that WebCryptoAgent improves trading stability, reduces spurious activity, and achieves stronger risk-adjusted performance compared to existing base-

lines. Beyond cryptocurrency markets, this work highlights the potential of reflective, memory-augmented agents for high-frequency decision-making tasks in dynamic and uncertain environments.

## Limitation and Future Work

While WebCryptoAgent demonstrates encouraging performance, several limitations remain. The framework currently relies on proprietary large language models for strategic reasoning, which may affect reproducibility across deployments. In addition, although the contextual reflection mechanism supports online adaptation without retraining, the replay buffer is updated using simple heuristics, and its long-term behavior warrants further study. Future work may explore alternative model choices, more principled reflection updates, and broader evaluation settings. We also expect that the two-tier reflective architecture could be applicable beyond cryptocurrency trading, though such extensions are left for future investigation.

## References

- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. [Qwen technical report](#). Preprint, arXiv:2309.16609.
- Yujie Ding, Shuai Jia, Tianyi Ma, Bingcheng Mao, Xizhe Zhou, Liuli Li, and Dongming Han. 2023. [Integrating stock features and global information via large language models for enhanced stock return prediction](#). Preprint, arXiv:2310.05627.
- Georgios Fatouros, Konstantinos Metaxas, John Soldatos, and Dimosthenis Kyriazis. 2024. [Can large language models beat wall street? unveiling the potential of ai in stock selection](#). Neural Computing and Applications.
- Jinchao Ge, Tengfei Cheng, Biao Wu, Zeyu Zhang, Shiya Huang, Judith Bishop, Gillian Shepherd, Meng Fang, Ling Chen, and Yang Zhao. 2025. [Vasevqa: Multimodal agent and benchmark for ancient greek pottery](#). arXiv preprint arXiv:2509.17191.
- Kemal Kirtac and Guido Germano. 2024. [Sentiment trading with large language models](#). Finance Research Letters, 62:105227.
- Kelvin J. L. Koa, Yunshan Ma, Ritchie Ng, and Tat-Seng Chua. 2024. [Learning to generate explainable stock predictions using self-reflective large language models](#). In Proceedings of the ACM Web Conference 2024 (WWW '24). ACM.
- Yang Li, Yangyang Yu, Haohang Li, Zhi Chen, and Khalidoun Khashanah. 2023. [Tradinggpt: Multi-agent system with layered memory and distinct characters for enhanced financial trading performance](#). Preprint, arXiv:2309.03736.
- Yufei Lin, Chengwei Ye, Huanzhen Zhang, Kangsheng Wang, Linuo Xu, Shuyan Liu, and Zeyu Zhang. 2025. [Ccl: collaborative curriculum learning for sparse-reward multi-agent reinforcement learning via co-evolutionary task evolution](#). In International Conference on Intelligent Computing, pages 51–62. Springer.
- Yitao Liu, Chenglei Si, Karthik R Narasimhan, and Shunyu Yao. 2025. [Contextual experience replay for self-improvement of language agents](#). In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14179–14198, Vienna, Austria. Association for Computational Linguistics.
- Alejandro Lopez-Lira and Yuehua Tang. 2023. [Can chatgpt forecast stock price movements? return predictability and large language models](#). Preprint, arXiv:2304.07619.
- Dakuan Lu, Hengkui Wu, Jiaqing Liang, Yipei Xu, Qianyu He, Yipeng Geng, Mengkun Han, Yingxi Xin, and Yanghua Xiao. 2023. [Bbt-fin: Comprehensive construction of chinese financial domain pre-trained language model, corpus and benchmark](#). Preprint, arXiv:2302.09432.
- Jingwei Shi, Zeyu Zhang, Biao Wu, Yanjie Liang, Meng Fang, Ling Chen, and Yang Zhao. 2025. [Presentagent: Multimodal agent for presentation video generation](#). In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 760–773.
- Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflexion: Language agents with verbal reinforcement learning](#). arXiv preprint arXiv:2303.11366.
- Saizhuo Wang, Hang Yuan, Lionel M. Ni, and Jian Guo. 2024. [Quantagent: Seeking holy grail in trading by self-improving large language model](#). Preprint, arXiv:2402.03755.
- Saizhuo Wang, Hang Yuan, Leon Zhou, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2023. [Alpha-gpt: Human-ai interactive alpha mining for quantitative investment](#). Preprint, arXiv:2308.00016.
- Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabrowski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kamdar, David Rosenberg, and Gideon Mann. 2023. [Bloomberggpt: A large language model for finance](#). Preprint, arXiv:2303.17564.
- Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. 2025. [Tradingagents: Multi-agents llm financial trading framework](#). Preprint, arXiv:2412.20138.

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](#). *Preprint*, arXiv:2306.05443.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, and 36 others. 2023a. [Baichuan 2: Open large-scale language models](#). *Preprint*, arXiv:2309.10305.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023b. [Fingpt: Open-source financial large language models](#). *Preprint*, arXiv:2306.06031.

Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Denghui Zhang, Rong Liu, Jordan W. Suzhou, and Khaldoun Khashanah. 2023. [Finmem: A performance-enhanced llm trading agent with layered memory and character design](#). *Preprint*, arXiv:2311.13743.

Boyu Zhang, Hongyang Yang, and Xiao-Yang Liu. 2023a. [Instruct-fingpt: Financial sentiment analysis by instruction tuning of general-purpose large language models](#). *Preprint*, arXiv:2306.12659.

Nonghai Zhang, Zeyu Zhang, Jiazi Wang, Yang Zhao, and Hao Tang. 2025a. [Vasevqa-3d: Benchmarking 3d vlms on ancient greek pottery](#). *arXiv preprint arXiv:2510.04479*.

Ruicheng Zhang, Yu Sun, Zeyu Zhang, Jinai Li, Xiaofan Liu, Hoi Fan Au, Huawei Guo, and Puxin Yan. 2025b. [Marl-mambacountour: Unleashing multi-agent deep reinforcement learning for active contour optimization in medical image segmentation](#). In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 7815–7824.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuhui Chen, Christopher De-wan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mi-haylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. [Opt: Open pre-trained transformer language models](#). *Preprint*, arXiv:2205.01068.

Xuanyu Zhang, Qing Yang, and Dongliang Xu. 2023b. [Xuanyuan 2.0: A large chinese financial chat model with hundreds of billions parameters](#). *Preprint*, arXiv:2305.12002.