StableEval Arena: A Cost-Aware Agentic Benchmark for Stablecoin Price Stability Prediction

Anonymous Author(s)

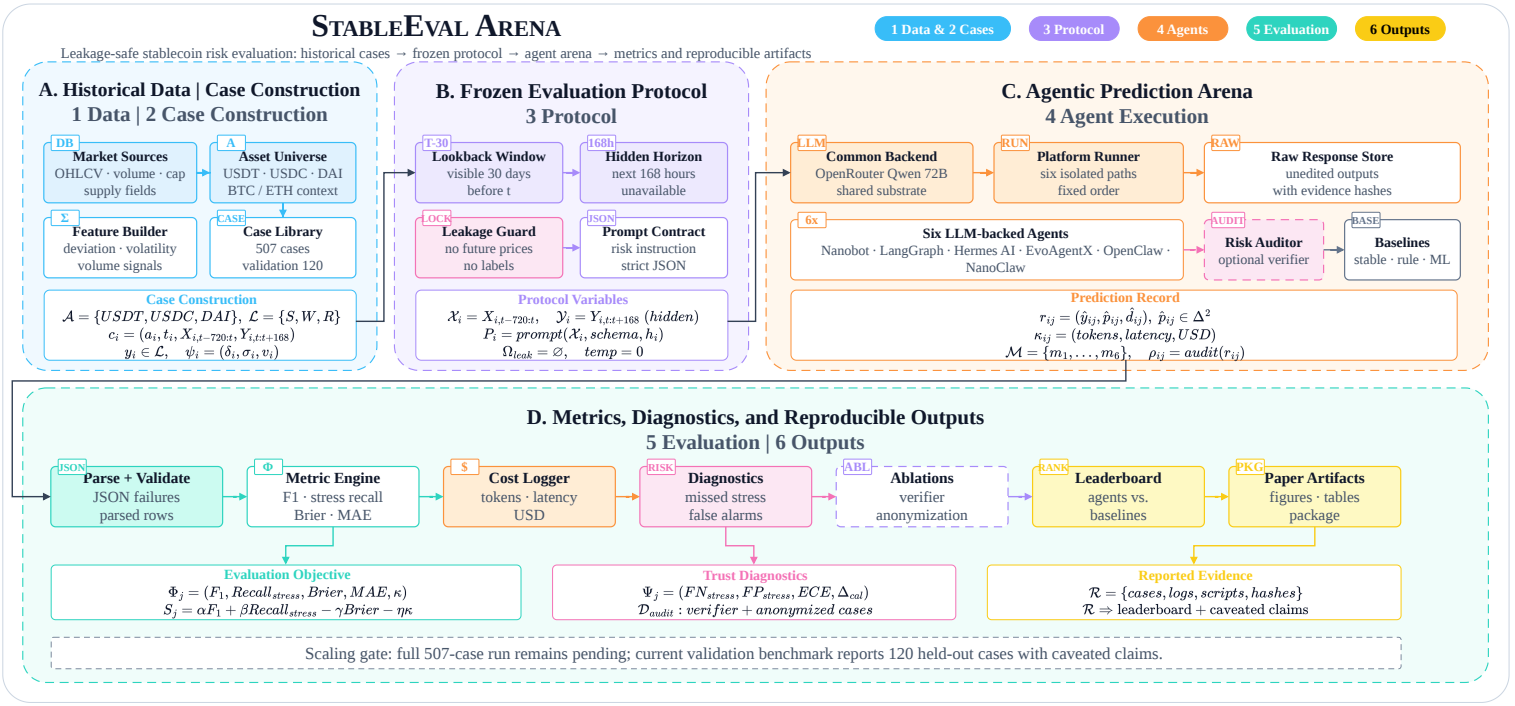


Figure 1: Overview of StableEval Arena.

Abstract

We introduce StableEval Arena, a demonstration framework that deploys autonomous AI agents to predict stablecoin price stability and deviations from the one-dollar peg, while jointly evaluating predictive precision and computational cost. Using historical replay and simulated paper-trading with high-fidelity on-chain and exchange data, our framework measures directional accuracy without real-capital risk. We propose a dual-metric protocol scoring both prediction quality and operational cost—token consumption, API latency, and inference spend. Reported findings are illustrative across selected time windows and are expected to vary with different market regimes, stablecoin compositions, and evaluation horizons. Future work may extend this infrastructure to live capital deployment, multi-agent cross-platform orchestration, dynamic inference budgeting, sub-minute mempool and oracle-feed integration, and granular evaluation of detection latency and confidence calibration. StableEval Arena offers a reproducible, low-cost testbed for trustworthy agentic AI in decentralized finance. In the spirit of open science, we have released the dataset on Hugging Face and the source code on GitHub as open-access resources to facilitate replication.

CCS Concepts

• **Computing methodologies** → **Multi-agent systems**; • **Applied computing** → *Secure online transactions.*

Keywords

Agentic AI Evaluation, Stablecoin Price Stability, Cost-aware Benchmarking, AI Agent Infrastructure, Decentralized Finance, Trustworthy AI

1 Introduction

Stablecoins have become core infrastructure for digital finance. As fiat-referenced programmable assets, they support trading, lending, payments, collateral, liquidity provision, and settlement across centralized exchanges, DeFi, and Web3 payment systems. Their utility depends on a fragile promise: a dollar-pegged stablecoin should remain near one U.S. dollar. When this promise weakens, downstream applications may face pricing errors, liquidity stress, payment failure, or settlement uncertainty. Recent stablecoin research therefore calls for transparent metrics, stakeholder-oriented evaluation, and reproducible benchmarking infrastructure [18].

Stablecoin stability prediction differs from ordinary cryptocurrency forecasting. Rather than predicting the direction of volatile assets such as BTC or ETH, the task is to determine whether a stablecoin remains stable, enters a warning state, or experiences

stress relative to the one-dollar peg. This requires time-bounded evidence on peg deviation, volatility, trading volume, market context, and stablecoin-specific mechanisms. While prior forecasting work highlights multimodal time-series evidence [7], stablecoin evaluation must also distinguish normal peg noise from rare, high-consequence stress events.

Agentic AI systems are increasingly used for structured reasoning, tool use, workflow execution, and decision support, but trustworthy evaluation requires more than final-answer accuracy. Benchmarks should also measure protocol compliance, output validity, runtime cost, reliability, and failure modes. Agent2 RL-Bench emphasizes unified agent-facing interfaces, fixed-budget execution, runtime recording, and post-hoc diagnosis [2]. TSFM-Bench motivates standardized forecasting protocols [14], Prediction Arena evaluates autonomous prediction agents with cost and behavior metrics [17], and TabArena demonstrates reproducible comparison through curated datasets, leaderboards, and public code [5]. However, existing benchmarks do not directly evaluate whether agentic AI systems can serve as cost-aware and trustworthy stablecoin price-stability agents.

We introduce StableEval Arena, a historical-replay benchmark for cost-aware agentic stablecoin evaluation. Each case gives an agent a frozen 30-day lookback window for a target stablecoin and asks it to predict the next-7-day stability state and dollar-peg deviation. We compare multiple agentic AI platforms with simple baselines under a unified protocol, measuring prediction quality, confidence, calibration, reliability, latency, token usage, and inference cost.

This paper addresses three research questions:

- **RQ1: Evaluation.** How do agentic AI systems compare with simple baselines on stablecoin stable/watch/stress prediction and dollar-peg deviation forecasting?
- **RQ2: Trustworthiness.** How reliable and cost-aware are agents under stress recall, missed-depeg risk, false alarms, calibration, latency, token usage, and inference cost?
- **RQ3: Agentic behavior.** What do the results reveal about the strengths and limitations of current agentic AI systems on Web3 financial-risk tasks?

Our results show that agents can produce valid structured outputs and improve aggregate warning classification under calibrated protocols, yet remain weak at rare severe stress detection. This gap between operational reliability and financial-risk trustworthiness is the central finding of StableEval Arena. We position the benchmark as a reproducible evaluation standard at the intersection of agentic AI, decentralized finance, and trustworthy cost-aware benchmarking.

In the spirit of open science, we have released the dataset on Hugging Face[1] and the source code on GitHub[2] as open-access resources to facilitate replication.

2 Background and Related Work

2.1 Stablecoin: Price Stability and Monitoring

Stablecoins are designed to provide fiat-referenced digital value, yet their stability is not automatic. Prior work shows that peg stability depends on market design, arbitrage access, liquidity, redemption mechanisms, reserve or collateral quality, and user confidence instead of price history alone [1, 9, 12, 16]. This makes stablecoin monitoring a domain-specific risk-evaluation problem, where a useful system should not only estimate whether a token remains close to its one-dollar reference value, but also distinguish ordinary market noise from operationally meaningful stress. Recent work further argues that stablecoins should be evaluated as programmable financial infrastructure whose performance depends on the needs of different stakeholders and should be assessed through transparent, reproducible benchmarks. [18]. StableEval Arena follows this direction by operating price stability as a historical-replay prediction task over frozen evidence windows, while leaving richer reserve, redemption, and regulatory data integration to future work.

2.2 Time-Series Forecasting and Agentic Financial Prediction

Though Stablecoin price-stability prediction is related to time-series forecasting, StableEval does not evaluate ordinary forecasting models. Time-series benchmark papers such as TSFM-Bench emphasize unified experimental protocols, standardized dataset splitting, loading, normalization, and evaluation settings to enable fair comparison across foundation models [14]. Cryptocurrency forecasting work also suggests that market prediction benefits from multimodal evidence rather than price-only modeling [7]. StableEval adopts the protocol discipline of forecasting benchmarks: it integrates fixed lookback windows, fixed prediction horizons, and standardized metrics; however, it shifts the unit of evaluation from a trained model to an agentic decision system. Each agent receives a frozen 30-day evidence packet and produces a structured next-7-day prediction of stable/watch/stress status, peg-deviation magnitude, direction, and confidence. Thus, the benchmark is forecasting-inspired, but its core question is whether agentic systems can transform bounded financial evidence into reliable and cost-aware risk judgments.

2.3 Evaluating Agentic AI Systems

Recent agent benchmarks have increasingly move beyond static question answering to evaluating systems that act under constraints. AgentBench evaluates large language models as agents across interactive environments, highlighting reasoning, decision-making, and instruction following as core obstacles for SOTA agents [15]. Similarly, Agent2 RL-Bench argues that agent evaluation should capture system behavior under fixed budgets, including runtime recording, failure diagnosis, and post-hoc summarization rather than final scores alone [2]. StableEval follows this agent-evaluation perspective in a financial historical-replay setting. Specifically, the evaluated agentic AI systems are not only judged by their textual rationales; moreover, they are evaluated by structured outputs, platform execution mode, JSON validity, failure rate, latency, token usage, and predictive performance against hidden future outcomes. Adding on to that, this design also allows us to separate native/local agent-framework runs from adapter-based executions, which is important for fair interpretation of heterogeneous agent platforms.

2.4 Cost-Aware and Trustworthy Benchmarking

Accuracy alone is insufficient for financial agent benchmarks. Similar prior work such as Prediction Arena evaluates autonomous AI

[1] *URL hidden for anonymous review.*
[2] *URL hidden for anonymous review.*

Table 1: Comparison between Agentic AI Systems: OpenClaw-Adjacent Alternatives

This table summarizes State-Of-The-Art representative Agentic AI systems, covering personal agents, OpenClaw-family alternatives, self-improving agents, self-evolving workflows, and graph-based orchestration frameworks.

Dimension	OpenClaw	Nanobot	NanoClaw	Hermes Agent	EvoAgentX	LangGraph
System Type	General-purpose personal AI assistant	Ultra-lightweight personal AI agent	Containerized OpenClaw alternative	Self-improving personal AI agent	Framework for evolving multi-agent workflows	Graph-based agent workflow framework
Access / License	Open-source; MIT [OC-GH 2026]	Open-source; MIT [NB-GH 2026]	Open-source; MIT [NC-GH 2026]	Open-source; MIT [HA-GH 2026]	Open-source; MIT; paper available [EAX-GH 2026]; [Wang et al. 2025]	OS LangChain ecosystem [LG-GH 2026]; [LG-Docs 2026]
Interface / Deployment	Multi-channel assistant; desktop/-mobile; gateway	CLI, gateway, chat apps, Docker/Linux	Messaging apps via skills; container-first runtime	CLI plus messaging gateway; multi-backend terminals	Python framework, notebooks, MCP/-tool workflows	Python framework; graph workflows; deployment support
Memory and Persistence	Sessions, skills, cron, usage commands	Long-term memory, Dream memory, heartbeat scheduling	Group memory and scheduled jobs	Persistent memory, user modeling, self-improving skills	Short-/long-term memory modules	Short-/long-term memory; durable execution
Tools / Orchestration	Skills, model failover, multi-agent routing	Small agent loop; tools, skills, MCP	Skills, web access, Anthropic Agents SDK	Tools, skills, cron, isolated subagents, MCP	Workflow plus TextGrad, AFlow, MIPRO	Graph control, tools, HITL, persistence
Safety / Control	Optional Docker sandbox for non-main sessions	Hardened shell/tool behavior; Docker deployment	Container isolation and credential separation	Command approval, DM pairing, container isolation	HITL checkpoints and tool control	Human-in-the-loop control and durable state

Note. Rows are selected from common agentic-AI review dimensions, including agent framework, action/interface surface, memory or persistence, tool orchestration, and safety or human control. These categories reflect how recent surveys describe LLM-based agents as systems combining planning, memory, tool use, feedback, and evaluation/control.
Selection. OpenClaw and Nanobot are retained as planned experimental baselines; NanoClaw is added as a security-oriented OpenClaw-family alternative. Hermes Agent represents a self-improving personal-agent comparator, EvoAgentX represents self-evolving multi-agent workflows, and LangGraph serves as a mature reference for durable, controllable, long-running agent orchestration.
Source tags. OC-GH: OpenClaw GitHub; NB-GH: Nanobot GitHub; NC-GH: NanoClaw GitHub; HA-GH: Hermes Agent GitHub; EAX-GH: EvoAgentX GitHub; Wang et al. 2025: *EvoAgentX: An Automated Framework for Evolving Agentic Workflows*; LG-GH: LangGraph GitHub; LG-Docs: LangGraph documentation.

models in prediction markets and analyzes by not only returns and win rates, but also by token usage, cycle time, settlement behavior, exit patterns, and cross-platform differences [17]. StableEval adapts this cost-aware evaluation logic to stablecoin risk monitoring. A trustworthy stablecoin agent should be judged not only by aggregate accuracy or macro-F1, but also by stress recall, missed-depeg rate, false-alarm rate, calibration, computational cost, and operational reliability. This distinction is important because a system may produce syntactically valid outputs while still failing at rare, high-consequence stress detection.

Our benchmark design is also motivated by recent work on benchmark infrastructure. TabArena argues that benchmark papers should curate datasets and models carefully, release reproducible code, report leaderboard-style comparisons, and document limitations rather than treating benchmarks as one-off static artifacts [5]. Dataset documentation practices such as Datasheets for Datasets further motivate transparent reporting of data provenance, intended use, limitations, and potential leakage risks [8]. StableEval therefore positions itself as a reproducible, cost-aware, and domain-grounded benchmark for evaluating agentic AI systems in decentralized finance. Our aim is not to let current agents solve stablecoin stress prediction problems, but to provide a standard for comparing agent capabilities, costs, and trustworthiness limitations.

3 StableEval Arena

3.1 Overview

StableEval Arena is a historical-replay benchmark for testing whether agentic AI systems can act as cost-aware agents for stablecoin price-stability and dollar-peg deviation prediction. It supports trustworthy evaluation through three requirements: comparative testing of agents and non-agent baselines under the same input, output, and scoring protocol; financially meaningful short-horizon stablecoin predictions; and operational auditability through logged outputs, costs, latency, failure status, and reproducibility metadata. The benchmark therefore combines three components: frozen historical evidence, structured agent outputs, and joint evaluation of prediction quality, operational cost, and reliability.

3.2 Task Formulation

For StableEval Arena, each benchmark case is defined by a target stablecoin (USDT, USDC, DAI), an observation time t, a frozen 30-day lookback window with hourly data (720 hours), and a hidden 7-day (168 hours) prediction horizon. The target agent (Nanobot,

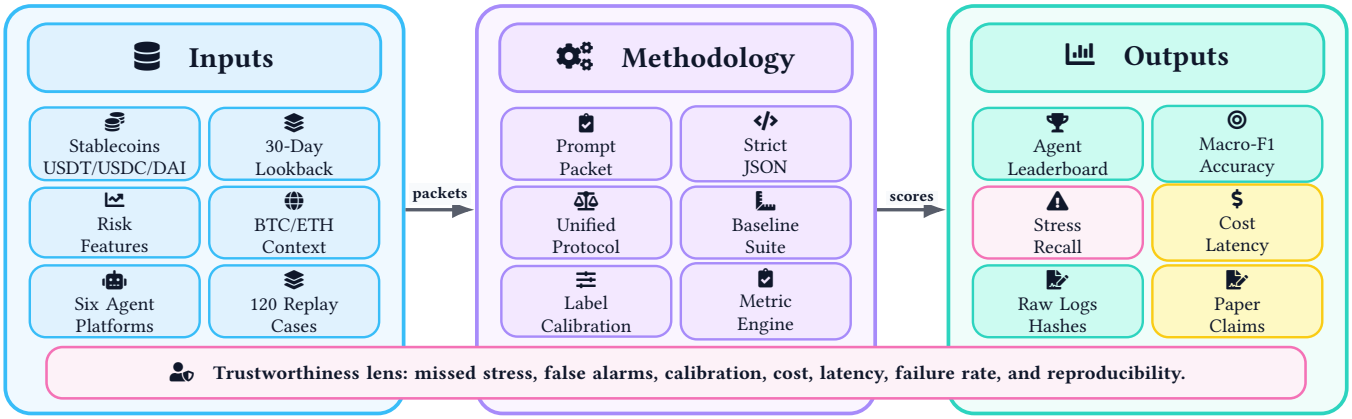


Figure 2: Experiment design of StableEval Arena: inputs, unified evaluation protocol, and paper-reportable outputs.

OpenClaw, NanoClaw, Hermes AI, EvoAgentX, LangGraph) receives only the hourly information available up to t and must predict the stablecoin's price-stability condition over days $t+1$ to $t+7$. This historical-replay design prevents the agent from observing future prices or labels while allowing objective evaluation after inference. Formally, let $\mathscr{C} = \{\text{USDT}, \text{USDC}, \text{DAI}\}$ denote the target stablecoin universe and let $\mathscr{Y} = \{\texttt{stable}, \texttt{watch}, \texttt{stress}\}$ denote the stability-label space. A benchmark case $\xi_i \in \mathscr{D}$ is defined as

$$\xi_i = (\chi_i, t_i, \mathscr{L}_i, \mathscr{H}_i), \qquad \chi_i \in \mathscr{C},$$

where χ_i is the target stablecoin, t_i is the observation time, $\mathscr{L}_i$ is the frozen lookback evidence window, and $\mathscr{H}_i$ is the hidden future horizon. The two temporal windows are

$$\mathscr{L}_i = \{\boldsymbol{x}_{\chi_i,\tau} : \tau = t_i - 719, \ldots, t_i\},$$

$$\mathscr{H}_i = \{p_{\chi_i,\tau} : \tau = t_i + 1, \ldots, t_i + 168\}.$$

where $\boldsymbol{x}_{\chi,\tau} \in \mathbb{R}^d$ contains price, volume, peg-deviation, volatility, liquidity, and market-context features available at hour τ, and $p_{\chi,\tau}$ is the hourly close price.

The required agent output is a structured JSON object corresponding to the prediction vector

$$f_{\theta_a}(\xi_i) \mapsto \widehat{\boldsymbol{y}}_{a,i} = (\widehat{s}_{a,i}, \widehat{\omega}_{a,i}, \widehat{\Delta}_{a,i}, \widehat{\phi}_{a,i}, \widehat{\kappa}_{a,i}, r_{a,i}),$$

where $a \in \mathscr{A}$ is an evaluated agent, θ_a denotes its platform and backend configuration, $\widehat{s}_{a,i} \in \mathscr{Y}$ is the predicted stability class, $\widehat{\omega}_{a,i} \in [0,1]$ is the predicted next-window depeg-risk probability, $\widehat{\Delta}_{a,i} \in \mathbb{R}_{\geq 0}$ is the predicted maximum absolute peg deviation in basis points, $\widehat{\phi}_{a,i} \in \{-1, 0, +1\}$ is the predicted deviation direction, $\widehat{\kappa}_{a,i} \in [0,1]$ is the reported confidence, and $r_{a,i}$ is a short evidence-based rationale. This output schema forces each agent to provide both a categorical risk judgment and a calibrated numerical estimate, while keeping outputs easy to parse and compare across agent frameworks. (i) a three-class price-stability prediction, `stable`, `watch`, or `stress`; (ii) a probability of depeg risk in the next seven days; (iii) the predicted maximum absolute deviation from the one-dollar peg in basis points; (iv) the predicted direction of deviation; (v) a confidence score; and (vi) a short evidence-based rationale. This output schema forces each agent to provide both a categorical risk judgment and a calibrated numerical estimate, while keeping outputs easy to parse and compare across agent frameworks.

Unlike standard forecasting, the evaluated object is an agentic decision system rather than a trained time-series model; therefore, we evaluate structured prediction quality together with stress-specific failure, calibration, cost, and reliability.

3.3 Dataset and Case Construction

StableEval Arena focuses on USDT, USDC, and DAI, covering centralized issuer-backed and decentralized collateral-backed stablecoins. BTC and ETH are included only as context assets to capture broader crypto-market stress. Each case is constructed from hourly price-volume data and daily market-structure features. The core raw fields include open, high, low, close, base volume, and USD volume. From these data, the benchmark cleans and derives risk-sensitive features such as latest close price, latest peg deviation, 7-day and 30-day maximum absolute deviation, duration above deviation thresholds, deviation volatility, recent high-low range, volume changes, market-cap or supply changes when available, and BTC/ETH market context. These features are computed only from the 30-day (720 hours) lookback window.

Dollar-peg deviation is measured as absolute distance from the one-dollar reference value. For stablecoin χ at hour τ, we define

$$\delta_{\chi,\tau} = 10^4 |p_{\chi,\tau} - 1|,$$

where $p_{\chi,\tau}$ is the hourly close price. For case ξ_i, the hidden-horizon maximum deviation and threshold-duration statistic are

$$\Delta_i = \max_{\tau \in \mathscr{H}_i} \delta_{\chi_i,\tau}, \qquad d_i(\vartheta) = \sum_{\tau \in \mathscr{H}_i} \mathbf{1}\{\delta_{\chi_i,\tau} \geq \vartheta\},$$

where ϑ is a pre-declared deviation threshold in basis points. We also record a strict sustained-depeg indicator

$$z_i = \mathbf{1}\left\{\exists \tau \in \mathscr{H}_i : \delta_{\chi_i,\tau}, \delta_{\chi_i,\tau+1}, \delta_{\chi_i,\tau+2} \geq 100\right\}.$$

The final hidden label $\ell_i \in \mathscr{Y}$ is assigned by the frozen evaluator:

$$\ell_i = \begin{cases} \texttt{stress}, & z_i = 1 \text{ or } \Delta_i \geq \vartheta_{\text{stress}}, \\ \texttt{watch}, & \vartheta_{\text{watch}} \leq \Delta_i < \vartheta_{\text{stress}} \text{ or } d_i(\vartheta_{\text{watch}}) \geq h_{\text{watch}}, \\ \texttt{stable}, & \text{otherwise}, \end{cases}$$

where ϑ_{watch}, $\vartheta_{\text{stress}}$, and h_{watch} are fixed before evaluation. This design avoids labeling every small fluctuation as a depeg while preserving sensitivity to rare but consequential stress events.

For the intermediate validation benchmark, we construct a 120-case set consisting of 54 stable cases, 45 watch cases, and 21 stress cases. The set includes all enhanced stress cases and all strict sustained-depeg cases, together with near-stress stable controls and natural stable controls. Hidden future-horizon labels and horizon statistics are used only for case selection and evaluation; they are never included in the agent-facing prompt packets. This construction supports more informative evaluation than a purely natural distribution, where severe stress cases are rare, while still preserving a reproducible historical-replay protocol.

3.4 Agents Evaluated and Baselines

StableEval Arena compares six agentic systems: Nanobot, LangGraph, Hermes AI, EvoAgentX, OpenClaw, and NanoClaw. All systems receive the same input packet, follow the same output schema, and are evaluated by the same scripts. To ensure transparency, the benchmark records each platform's execution mode. Nanobot, LangGraph, Hermes AI, and EvoAgentX are treated as the main local or framework-level agent comparisons. OpenClaw and NanoClaw are included as adapter-based systems unless native CMDOP execution is verified; We report this distinction explicitly to avoid overstating execution depth.

The benchmark also includes non-agent baselines. These include an always-stable baseline, a historical-persistence baseline, a threshold-rule baseline, and a simple machine-learning baseline where feasible. Baselines are necessary because agentic systems should not be evaluated only against each other. Moreover, a central question is whether agentic execution provides additional value beyond simple rules that exploit the strong base-rate stability of major stablecoins. Following benchmark practice, the same cases and labels are used for both agentic systems and baselines.

3.5 Evaluation Protocol

All agentic systems are evaluated under a unified protocol. In the main intermediate validation setting, each platform uses the same 120 cases, same case order, same model backend, same temperature, same prompt template, same input schema, same output schema, and same evaluator. The current frozen protocol uses `qwen/qwen-2.5-72b-instruct` with temperature 0. Web browsing is disabled, and all prompts instruct agents to use only the provided lookback-window evidence.

The primary classification metrics are accuracy, macro-F1, balanced accuracy, per-class precision, and per-class recall. Macro-F1 and balanced accuracy are emphasized because stablecoin stress events are rare and raw accuracy can be dominated by stable cases. Stress recall is reported separately because a system that misses stress events may be operationally unsafe even if it performs reasonably on aggregate metrics. Let $N = |\mathscr{D}|$ and let $\widehat{s}_{a,i}$ and ℓ_i denote the predicted and hidden labels for agent a on case i. For class $y \in \mathscr{Y}$, precision, recall, and F1 are

$$P_y = \frac{\mathrm{TP}_y}{\mathrm{TP}_y + \mathrm{FP}_y}, \qquad R_y = \frac{\mathrm{TP}_y}{\mathrm{TP}_y + \mathrm{FN}_y}, \qquad F1_y = \frac{2P_yR_y}{P_y + R_y}.$$

Macro-F1 and balanced accuracy are then

$$\mathrm{MacroF1}_a = \frac{1}{|\mathscr{Y}|}\sum_{y\in\mathscr{Y}} F1_y, \qquad \mathrm{BalAcc}_a = \frac{1}{|\mathscr{Y}|}\sum_{y\in\mathscr{Y}} R_y.$$

For severe-risk evaluation, we define

$$\mathrm{MDR}_a = \frac{\sum_i \mathbf{1}\{z_i = 1, \widehat{s}_{a,i} \neq \texttt{stress}\}}{\sum_i \mathbf{1}\{z_i = 1\}},$$

$$\mathrm{FAR}_a = \frac{\sum_i \mathbf{1}\{z_i = 0, \widehat{s}_{a,i} = \texttt{stress}\}}{\sum_i \mathbf{1}\{z_i = 0\}}.$$

where MDR_a is the missed-depeg rate and FAR_a is the severe false-alarm rate. Severe-risk evaluation further includes missed-depeg rate and false-alarm rate.

The benchmark also evaluates probabilistic and continuous predictions. Depeg probability is evaluated using Brier score and also calibration-oriented analysis, following the broader forecasting principle that probabilistic predictions should be judged by both sharpness and reliability [10]. Predicted maximum deviation is evaluated using mean absolute error and root mean squared error in basis points. The probabilistic and continuous metrics are

$$\mathrm{Brier}_a = \frac{1}{N}\sum_{i=1}^{N}\left(\widehat{\varpi}_{a,i} - \mathbf{1}\{\ell_i = \texttt{stress}\}\right)^2,$$

$$\mathrm{MAE}^{\Delta}_a = \frac{1}{N}\sum_{i=1}^{N}|\widehat{\Delta}_{a,i} - \Delta_i|, \qquad \mathrm{RMSE}^{\Delta}_a = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\widehat{\Delta}_{a,i} - \Delta_i)^2}.$$

A pre-declared calibration rule maps model-reported probability and predicted deviation into the final stable/watch/stress label. The rule is fixed before evaluation.

3.6 Cost, Reliability, and Reproducibility

Because StableEval Arena targets trustworthy agentic AI, evaluation includes operational cost and reliability in addition to prediction quality. For every agent-case run, the benchmark records prompt tokens, completion tokens, total tokens, latency, estimated inference cost, cost per case, valid-JSON status, failure status, raw output, parsed output, model backend, execution mode, prompt version, and input-packet hash. These logs allow the evaluator to distinguish three kinds of failure: predictive error, malformed or invalid output, and operational failure. For each agent-case run, operational cost is formalized as

$$\Gamma_{a,i} = c_{\mathrm{in}} n^{\mathrm{in}}_{a,i} + c_{\mathrm{out}} n^{\mathrm{out}}_{a,i},$$

where $n^{\mathrm{in}}_{a,i}$ and $n^{\mathrm{out}}_{a,i}$ are prompt and completion tokens, and c_{in} and c_{out} are backend-specific token prices. Agent-level cost and latency are

$$\overline{\Gamma}_a = \frac{1}{N}\sum_{i=1}^{N}\Gamma_{a,i}, \qquad \overline{\tau}_a = \frac{1}{N}\sum_{i=1}^{N}\tau_{a,i}.$$

Let $\nu_{a,i} = 1$ indicate valid JSON output and $\omega_{a,i} = 1$ indicate execution failure. The valid-output and failure rates are

$$\rho_a = \frac{1}{N}\sum_{i=1}^{N}\nu_{a,i}, \qquad \Omega_a = \frac{1}{N}\sum_{i=1}^{N}\omega_{a,i}.$$

This cost-aware design is motivated by the fact that financial monitoring systems must be usable under budget and latency constraints. A high-performing agent that is too expensive, slow, unstable, or difficult to reproduce may be unsuitable for practical deployment. Conversely, an agent that produces valid outputs with

Table 2: Protocol, reliability, cost, and latency for the six agentic systems.

•Metric	Nano	LG	Hermes	Evo	OClaw	NClaw
Cases	120	120	120	120	120	120
Valid JSON (↑)	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Failure (↓)	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Inval output (↓)	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Prompt tok. (↓)	7,730	26,873	7,790	13,665	13,613	13,665
Completion tok. (↓)	239	657	241	432	377	365
Total tok. (↓)	7,969	27,530	8,031	14,097	13,989	14,030
Latency sec. (↓)	34.3	72.6	33.2	54.3	30.7	29.4
Total cost (↓)	$0.345	$1.212	$0.348	$0.611	$0.625	$0.627
Cost/case (↓)	$0.0029	$0.0101	$0.0029	$0.0051	$0.0052	$0.0052
Cost/correct (↓)	$0.0065	$0.0229	$0.0068	$0.0130	$0.0128	$0.0118

low cost but systematically misses rare stress events exposes a trustworthiness gap. Therefore, StableEval Arena treats cost, latency, valid-output rate, failure rate, and stress-detection behavior as part of the same evaluation framework.

For reproducibility, the benchmark preserves case lists, prompt packets, prompt templates, calibration rules, adapter metadata, raw predictions, cost logs, and evaluation scripts. The intended release package includes a GitHub repository for code and reproduction commands and a Hugging Face dataset package for case metadata, schemas, data cards, and evaluation artifacts where redistribution is permitted. This release-oriented design follows recent benchmark practice in which public code, dataset documentation, and transparent evaluation protocols are part of the benchmark contribution itself [5, 14].

4 Results

We report results from the 120-case intermediate validation set, which contains 54 stable, 45 watch, and 21 stress cases. This setting is designed to evaluate whether agentic systems can distinguish different levels of stablecoin price stability under a controlled historical-replay protocol. Unless otherwise stated, the main leaderboard uses the calibrated first-pass protocol, where each agent receives the same frozen 30-day lookback packet and predicts the next-7-day stability label and dollar-peg deviation. We report agent results together with non-agent baselines to assess whether agentic execution adds value beyond simple rules. Following benchmark-reporting practice, we focus on aggregate predictive quality, class-level stress detection, ablation behavior, and computational cost [5, 17].

4.1 Main Results

Table 2-5 summarize the main validation leaderboard. Among the agentic systems, Nanobot achieves the strongest calibrated first-pass result, with a Macro-F1 of 0.351, accuracy of 0.442, and balanced accuracy of 0.393. LangGraph follows with a Macro-F1 of 0.337, while NanoClaw, Hermes AI, OpenClaw, and EvoAgentX achieve Macro-F1 scores of 0.328, 0.321, 0.318, and 0.272, respectively. The strongest non-agent baseline is historical persistence, with a Macro-F1 of 0.285 and balanced accuracy of 0.300. Thus, the best calibrated agent outperforms the best baseline on Macro-F1 in the intermediate validation set.

Table 3: Main calibrated three-class classification results.

•Metric	Nano	LG	Hermes	Evo	OClaw	NClaw
Accuracy (↑)	0.442	0.442	0.425	0.392	0.408	0.442
Macro-F1 (↑)	0.351	0.337	0.321	0.272	0.318	0.328
Balanced acc. (↑)	0.393	0.381	0.369	0.348	0.368	0.385
Prec. stable (↑)	0.591	0.593	0.583	0.538	0.500	0.591
Recall stable (↑)	0.241	0.296	0.259	0.130	0.185	0.241
Prec. watch (↑)	0.400	0.400	0.387	0.375	0.389	0.415
Recall watch (↑)	0.844	0.800	0.800	0.867	0.822	0.867
Prec. stress (↑)	0.667	0.333	0.333	0.333	0.400	0.250
Recall stress (↑)	0.095	0.048	0.048	0.048	0.095	0.048

Table 4: Binary sustained-depeg diagnostics and continuous maximum-deviation prediction errors.

•Metric	Nano	LG	Hermes	Evo	OClaw	NClaw
Binary sustained-depeg detection						
AUROC (↑)	0.514	0.474	0.286	0.425	0.490	0.432
Avg. precision (↑)	0.031	0.033	0.022	0.023	0.029	0.024
Brier (↓)	0.047	0.046	0.047	0.050	0.048	0.050
ECE (↓)	0.125	0.118	0.107	0.124	0.117	0.119
False alarm (↓)	0.9%	0.9%	1.7%	1.7%	1.7%	1.7%
TN (↑)	116	116	115	115	115	115
FP (↓)	1	1	2	2	2	2
Continuous maximum-deviation errors						
MAE bps (↓)	56.6	58.7	57.9	66.9	62.0	66.4
RMSE bps (↓)	167.2	168.6	175.0	199.2	180.3	199.8
Error p10 bps	-63.4	-66.1	-67.0	-67.0	-63.4	-67.0
Error median bps	-2.0	-3.0	0.0	0.0	0.0	-1.0
Error p90 bps	36.2	35.1	40.0	40.0	40.0	40.0

Notice that this result should be interpreted as a benchmark outcome rather than a claim that stablecoin stress prediction is solved. The best agent improves aggregate three-class classification relative to the baselines, but stress-class recall remains low. Nanobot and OpenClaw obtain the highest agent stress recall under the calibrated first-pass protocol, at 0.095, while historical persistence reaches 0.333. Therefore, the main validation result shows a split pattern: agentic systems improve aggregate warning classification, but severe or rare stress detection remains difficult, whose potential reasons will be under discussion in 5.

4.2 Agent–Baseline Comparison

This comparison is central to StableEval Arena because the benchmark evaluates not only which agent ranks highest, but also whether agentic systems provide value beyond simple operational heuristics. The always-stable and simple logistic baselines both achieve an accuracy of 0.450 but a Macro-F1 of only 0.207, reflecting the limitation of majority-class behavior in a three-class stability

Table 5: Calibration, risk-auditor verifier, and anonymization ablation results.

•Metric	Nano	LG Hermes	Evo OClaw	NClaw
Raw first-pass and calibration ablation				
Raw acc. (↑)	0.400	**0.433**	0.425	0.383	0.417	0.425
Raw Macro-F1 (↑)	0.266	**0.301**	0.281	0.237	0.281	0.286
Δ Macro-F1 (↑)	**0.085**	0.036	0.040	0.035	0.037	0.042
Δ bal. acc. (↑)	**0.050**	0.013	0.004	0.015	0.011	0.021
Δ stress recall (↑)	**0.095**	0.048	0.048	0.048	**0.095**	0.048
Risk-auditor verifier ablation				
Verifier F1 (↑)	**0.349**	0.312	0.348	0.266	0.277	0.315
Verifier bal. acc. (↑)	**0.388**	0.342	0.380	0.328	0.326	0.359
Verifier stress rec. (↑)	**0.143**	0.095	**0.143**	0.095	0.095	0.095
Verifier cost (↓)	**$0.978**	$1.840	$0.982	$1.246	$1.253	$1.257
Verifier latency (↓)	64.0	97.9	68.1	83.6	68.1	**63.7**
Anonymization ablation for name/date sensitivity				
Label-change (↓)	25.0%	33.3%	**8.3%**	16.7%	**8.3%**	25.0%
Mean $\lvert\Delta p_{\text{dep}}\rvert$ (↓)	0.042	0.062	0.029	**0.017**	0.029	0.050
Mean $\lvert\Delta \hat{d}\rvert$ bps (↓)	8.5	8.4	**4.6**	7.9	8.3	14.2

task. The threshold-rule baseline improves Macro-F1 to 0.278, and historical persistence further improves it to 0.285.

Compared with these baselines, five of the six agentic systems exceed the best baseline on Macro-F1 under the calibrated first-pass protocol. Nanobot performs best overall, followed by LangGraph, NanoClaw, Hermes AI, and OpenClaw. EvoAgentX remains below the historical-persistence baseline. OpenClaw and NanoClaw are included as adapter-based systems in this validation, while Nanobot, LangGraph, Hermes AI, and EvoAgentX are treated as the primary local framework-level comparisons. This distinction is maintained in the execution metadata and is discussed further in the limitations section.

4.3 Ablation

We next compare three protocol variants: raw first-pass outputs, calibrated first-pass outputs, and verifier-based outputs. Calibration improves the best agent Macro-F1 from 0.301 in the raw first-pass setting to 0.351 in the calibrated setting. It also raises the best stress recall from 0.000 to 0.095. Such result indicates that post-hoc label mapping based on predicted risk probability and maximum expected deviation helps convert continuous risk estimates into more useful three-class stability labels.

The verifier layer produces a different tradeoff. Under the verifier-calibrated protocol, the best stress recall rises to 0.143, achieved by Nanobot and Hermes AI. However, the best verifier-calibrated Macro-F1 is 0.349, slightly below the best calibrated first-pass Macro-F1 of 0.351. The verifier also increases computational cost substantially: for Nanobot, total cost rises from $0.345 under the calibrated first-pass protocol to $0.978 under the verifier-calibrated protocol. Thus, verifier-based auditing improves stress recall modestly, but does not improve aggregate Macro-F1 while further imposing a higher inference cost.

We also evaluate a small anonymization ablation to probe sensitivity to real asset names and dates. The maximum label-change rate is 0.333, while mean probability changes range from 0.017 to 0.062. This suggests that agent outputs exhibit some sensitivity to identifiers, but the tested subset does not provide strong evidence that the validation results depend primarily on memorized real-world names or dates. We treat this as a reproducibility caveat rather than a failure of the benchmark.

4.4 Cost and Reliability

Table 2 and 3 report computational cost and reliability. All main agentic systems achieve a valid JSON rate of 1.0, a failure rate of 0.0, and an invalid-output rate of 0.0 under the frozen protocol. This indicates that the benchmark can reliably collect structured predictions from heterogeneous agentic systems.

Cost and latency vary substantially across platforms. Nanobot and Hermes AI are the least expensive main systems, with total costs of approximately $0.345 and $0.348, corresponding to about $0.0029 per case. LangGraph is the most expensive, with total cost of approximately $1.212, or $0.0101 per case, and also exhibits higher average latency. The adapter-based OpenClaw and NanoClaw systems fall between these extremes, with total costs of approximately $0.625 and $0.627, respectively. These differences show that computational cost is not uniform across agentic frameworks even when the prediction task, backend model, and output schema are fixed.

Together, the results establish three main empirical observations. First, calibrated agentic systems can outperform simple baselines on aggregate three-class Macro-F1. Second, stress detection remains weak, especially for strict sustained-depeg cases. Third, structured-output reliability is high, but cost and latency differ across platforms and increase when verifier-style auditing is added. The next section analyzes the trustworthiness implications of these findings.

5 Discussion and Analysis

StableEval Arena results reveal a central tension: agentic systems improve aggregate stable/watch/stress classification, but remain weak at rare stress detection. We also show that current agents can process structured historical evidence, produce valid outputs, and outperform simple baselines on Macro-F1, yet still miss high-consequence stress events. This gap between average performance and rare-event reliability is where evaluation, agentic AI, and trustworthiness intersect.

5.1 Aggregate Performance vs. Stress Reliability

The calibrated agents outperform the strongest simple baseline on Macro-F1 because they can synthesize more evidence than fixed rules. Historical persistence and threshold baselines react to limited past-deviation patterns, while LLM-backed agents can jointly consider peg behavior, volatility, volume, market context,

and stablecoin-specific notes. This explains why several agents exceed the historical-persistence baseline on aggregate classification, supporting the value of a unified agent benchmark similar in spirit to recent evaluation systems [5, 17].

However, Macro-F1 improvement does not imply reliable stress detection. The historical-persistence baseline has lower aggregate Macro-F1 but higher stress recall. This suggests that agents are conservative: because stablecoins usually remain near the peg, agents tend to choose stable or watch when evidence is ambiguous. That behavior can improve average classification while weakening rare-event detection. For stablecoin monitoring, this distinction matters because missing stress is not just another classification error; it can delay response to liquidity, payment, or confidence risks. StableEval therefore shows why agentic evaluation must report both aggregate quality and stress-specific metrics.

5.2 Hard-to-detect Stress

Stablecoin stress is difficult because it is rare, short-lived, and often driven by mechanisms not fully visible in compact price windows. A depeg may depend on reserves, redemption channels, exchange liquidity, collateral composition, governance, oracle behavior, or broader market confidence. A 30-day hourly historical packet improves reproducibility, but necessarily compresses this environment. Such compression is important in DeFi settings because oracle networks and cross-chain information flows help connect external data, smart contracts, and protocol-level risk; recent evidence links oracle integration to interoperability, risk-sharing, and resilience during turbulent crypto-market periods [4].

This helps explain why calibration improves Macro-F1 but does not solve strict depeg detection. Calibration makes agents less conservative by mapping risk probabilities and predicted deviations into more useful labels. However, it cannot recover missing information. Stablecoin research emphasizes that stablecoins are programmable monetary infrastructure requiring stakeholder-oriented metrics, not price-only monitoring [18]. Cryptocurrency forecasting work similarly suggests that market behavior often depends on multimodal signals beyond a single financial time series [7]. Thus, weak stress recall reflects both agent limitations and the difficulty of representing fragmented Web3 risk in a compact benchmark packet.

5.3 Reliability instead of Trustworthiness

All evaluated agents achieve 100% valid JSON and 0% failure rate, showing that the benchmark infrastructure can reliably collect structured outputs across heterogeneous agent systems. However, it should be noted that this is procedural reliability instead of financial trustworthiness. An agent can follow the schema and remain low-cost while still missing rare risk events.

This distinction is central to agentic AI evaluation. Recent benchmarks emphasize that agent evaluation should include runtime behavior, diagnostics, and failure modes, not only final scores [2]. Realistic agent environments also show that current agents can struggle despite appearing capable in narrower settings [19]. StableEval extends this concern to decentralized finance: valid outputs are necessary for deployment, but insufficient for trustworthy risk monitoring.

The verifier ablation reinforces this point. The verifier makes agents slightly more sensitive to stress, but does not improve Macro-F1 and substantially increases cost. This suggests that self-checking helps only when the relevant risk evidence is present. Without richer domain data, the verifier mainly reweights the same packet rather than discovering new signals.

5.4 Cross-Agent Differences

Since all agents use the same backend model, temperature, case order, and output schema, performance differences likely come from framework-level execution: how each system packages state, formats instructions, manages intermediate reasoning, and returns structured predictions. Nanobot achieves the best Macro-F1 while remaining low-cost, suggesting that lightweight orchestration can potentially work well when the evidence packet is already organized. LangGraph performs competitively but costs more and has higher latency, indicating that more complex orchestration does not automatically yield better financial prediction. Hermes AI is cheap but less accurate, while EvoAgentX underperforms among the main local frameworks, suggesting that current integration quality matters. OpenClaw and NanoClaw are competitive but adapter-based in this validation, so their scores should be interpreted as exploratory rather than full native CMDOP results. Overall, the results show that the best agent is not simply the highest-scoring one. A useful stablecoin agent must balance Macro-F1, stress recall, calibration, reliability, latency, and inference cost. This cost-aware view follows the broader logic of financial agent benchmarks.

In sum, StableEval Arena reveals that current agentic systems are promising but incomplete stablecoin risk agents. They can improve aggregate warning classification and produce reliable structured outputs, but they remain conservative under rare stress and limited by the evidence available in the benchmark packet. StableEval, as an evaluation framework, has hence become valuable for turning these weaknesses into measurable, reproducible trustworthiness findings.

6 Industry Implications and Future Work

The significance of StableEval Arena also points to a broader industrial question of whether agentic AI can become a trustworthy coordination layer for stablecoin-based payment systems. Stablecoins are increasingly discussed as programmable monetary infrastructure for payments, settlement, DeFi, and tokenized commerce, yet their deployment depends on more than price stability alone. Such framing aligns with DeFi research that treats blockchain-based finance as programmable and composable financial infrastructure, while also emphasizing risks from smart contracts, governance, oracle dependence, custody, scalability, and regulation [11]. Prior work emphasizes that stablecoin evaluation must consider stakeholder-specific metrics, regulation, governance, interoperability, and reproducible benchmarking [18]. Our results reinforce this view: agents can produce structured outputs reliably, but weak stress recall suggests that current agentic systems are not yet sufficient for high-consequence payment automation. Furthermore, we also provide three industrial implications as motivation for future work adding on to the benchmark StableEval creates.

A first industrial implication concerns recurring payments and delegated authorization. Web2 subscription systems rely on mature card, banking, and platform rails with established authorization, dispute, and refund mechanisms. By contrast, Web3 subscription payments require users to delegate payment authority to applications, smart contracts, or agentic services. This creates a trust problem: automatic stablecoin deduction may improve programmability, but it also raises unresolved questions about user consent, privacy, revocation, and liability. Recent work on stablecoins in retail payments similarly argues that stablecoin arrangements differ from card networks in transaction lifecycle, recourse, and risk allocation [13, 18]. StableEval suggests that agentic monitors could help decide when to execute, pause, or reroute payments, but only if they can detect peg stress and settlement risk more reliably than current systems.

A second implication concerns service-level agreements (SLAs), clearing, and settlement. For stablecoins to support large-scale application payments, systems must specify when a payment is considered final, how reconciliation is performed, what happens when settlement fails, and who bears the cost of delay. International policy discussions similarly treat stablecoin arrangements as payment infrastructures requiring consistent oversight, governance, and risk management [3, 6]. Our results show why SLA design should include agent reliability metrics. A system that returns valid JSON with zero runtime failures may still be unsafe if it misses stress events. Future benchmarks should therefore evaluate detection latency, reconciliation mismatch risk, settlement delay risk, and escalation behavior, not only next-window price classification.

A third implication concerns pricing, taxation, and cross-border allocation of payment risk. Even small deviations from the one-dollar peg can redistribute value among users, application developers, platforms, exchanges, and payment intermediaries. In subscription or cross-border payment settings, this raises practical questions: whether the user, application, or platform absorbs price fluctuation; how fees and value-added taxes are calculated; and how jurisdiction is determined when users, platforms, and IP-based location signals span multiple countries. Regulatory work on global stablecoin arrangements highlights the need for consistent cross-jurisdictional supervision and legal clarity [6]. StableEval can be extended to model these frictions by attaching business-cost labels to each prediction: false alarms may interrupt legitimate payments, while missed stress events may cause unfavorable settlement, underpayment, or compliance disputes.

These implications suggest three future directions. First, the benchmark should incorporate richer on-chain and off-chain data, including liquidity, redemption, exchange depth, oracle feeds, wallet-level transaction flows, and settlement records. Second, evaluation should move from prediction quality alone to stakeholder-aware cost functions, including user loss, developer revenue risk, platform compliance burden, and public regulatory cost. Third, future versions should evaluate multi-agent payment orchestration, where user agents, application agents, exchange agents, and compliance agents negotiate execution under explicit SLA and tax constraints. In this direction, StableEval Arena can evolve from a price-stability benchmark into a broader testbed for trustworthy agentic payment infrastructure.

7 Limitations

The current version of StableEval Arena has several limitations. First, its main empirical evidence comes from the 30-case pilot and 120-case validation; the full 507-case benchmark has not yet been evaluated across all agents. Thus, current results support a reproducible benchmark prototype and validation study, but not a complete characterization of agent behavior under the full natural historical distribution.

Second, severe stablecoin stress is rare. Although the 120-case validation includes all strict sustained-depeg cases, their small number makes strict binary depeg detection difficult to estimate robustly. Missed-depeg results should therefore be read as a warning signal rather than a final statistical conclusion. This limitation is structural and intrinsic: stablecoins usually remain near peg, while consequential failures occur in short, unusual windows.

Third, the benchmark intentionally prioritizes reproducible historical replay over exhaustive stablecoin surveillance. Its evidence packets cover price, volume, market-structure, and contextual signals, while leaving higher-cost integrations such as reserve attestations, redemption flows, oracle and mempool data, cross-chain liquidity, DeFi liquidations, and regulatory/news feeds to future extensions. Such design avoids the risks and irreproducibility of live-capital deployment, but does not yet test real-time tool use, adaptive memory, or feedback-driven execution. OpenClaw and NanoClaw should therefore be interpreted as adapter-based baselines rather than native CMDOP implementations.

Finally, residual parametric-memory effects cannot be fully ruled out because some packets retain real stablecoin names and dates, even with hidden future labels and an anonymization ablation. Future releases should broaden anonymized evaluation and add dataset cards, documentation, and reproducibility artifacts aligned with established practice [5, 8]. These boundaries clarify the benchmark's present scope and define a concrete path toward more reproducible, domain-aware agentic evaluation.

8 Conclusion

We introduce StableEval Arena, a cost-aware agentic benchmark for stablecoin price-stability prediction. Instead of performing ordinary cryptocurrency forecasting, StableEval motivates a historical-replay evaluation task in which agents receive frozen 30-day evidence windows and predict next-7-day stable/watch/stress status, dollar-peg deviation, direction, confidence, and risk probability. By comparing multiple agentic systems with simple baselines under a unified protocol, the benchmark evaluates not only predictive quality, but also calibration, stress recall, missed-depeg risk, valid-output rate, latency, token usage, and inference cost. The results show that calibrated agentic systems can outperform simple baselines on aggregate Macro-F1, while still remaining weak at rare stress and strict sustained-depeg detection. This gap between structured operational reliability and financial-risk trustworthiness is the central finding of StableEval Arena. Overall, the benchmark provides a reproducible foundation for evaluating trustworthy, domain-aware, and cost-efficient financial agents, while future work should expand the case pool, integrate richer on-chain and off-chain data, improve native agent execution, and extend the framework toward live payment, settlement, and stablecoin-risk monitoring scenarios.

References

[1] Douglas W. Arner, Raphael Auer, and Jon Frost. 2020. *Stablecoins: Risks, Potential and Regulation.* BIS Working Paper 905. Bank for International Settlements. doi:10.2139/ssrn.3979495

[2] Wanyi Chen, Xiao Yang, Xu Yang, Tianming Sha, Qizheng Li, Zhuo Wang, Bowen Xian, Fang Kong, Weiqing Liu, and Jiang Bian. 2026. Agent^2 RL-Bench: Can LLM Agents Engineer Agentic RL Post-Training? arXiv:2604.10547 [cs.AI] https://arxiv.org/abs/2604.10547

[3] Committee on Payments and Market Infrastructures. 2023. *Considerations for the Use of Stablecoin Arrangements in Cross-border Payments.* Technical Report. Bank for International Settlements. https://www.bis.org/cpmi/publ/d220.pdf

[4] Lin William Cong, Eswar S. Prasad, and Daniel Rabetti. 2025. *Financial and Informational Integration Through Oracle Networks.* Working Paper 33639. National Bureau of Economic Research. doi:10.3386/w33639

[5] Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. 2025. TabArena: A Living Benchmark for Machine Learning on Tabular Data. arXiv:2506.16791 [cs.LG] https://arxiv.org/abs/2506.16791 NeurIPS 2025 Track on Datasets and Benchmarks.

[6] Financial Stability Board. 2023. *High-level Recommendations for the Regulation, Supervision and Oversight of Global Stablecoin Arrangements.* Technical Report. Financial Stability Board. https://www.fsb.org/2023/07/high-level-recommendations-for-the-regulation-supervision-and-oversight-of-global-stablecoin-arrangements-final-report/

[7] Yihang Fu, Mingyu Zhou, and Luyao Zhang. 2024. DAM: A Universal Dual Attention Mechanism for Multimodal Timeseries Cryptocurrency Trend Forecasting. In *2024 IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom).* IEEE, Piscataway, NJ, USA, 73–80. doi:10.1109/MetaCom62920.2024.00025

[8] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for Datasets. *Commun. ACM* 64, 12 (2021), 86–92. doi:10.1145/3458723

[9] Gary B. Gorton and Jeffery Y. Zhang. 2023. Taming Wildcat Stablecoins. *University of Chicago Law Review* 90, 3 (2023), 909–972. https://lawreview.uchicago.edu/print-archive/taming-wildcat-stablecoins

[10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70).* PMLR, Sydney, NSW, Australia, 1321–1330. https://proceedings.mlr.press/v70/guo17a.html

[11] Campbell R. Harvey and Daniel Rabetti. 2024. International Business and Decentralized Finance. *Journal of International Business Studies* 55, 7 (2024), 840–863. doi:10.1057/s41267-024-00705-7

[12] Ariah Klages-Mundt, Dominik Harz, Lewis Gudgeon, Jun-You Liu, and Andreea Minca. 2020. Stablecoins 2.0: Economic Foundations and Risk-based Models. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies (AFT '20).* Association for Computing Machinery, New York, NY, USA, 59–79. doi:10.1145/3419614.3423261

[13] Yuquan Li, Yuexin Xiang, Qin Wang, Tsz Hon Yuen, Andreas Deppeler, and Jiangshan Yu. 2026. SoK: Stablecoins in Retail Payments. arXiv:2601.00196 [cs.CR]

[14] Zhe Li, Xiangfei Qiu, Peng Chen, Yihang Wang, Hanyin Cheng, Yang Shu, Jilin Hu, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, and Bin Yang. 2025. TSFM-Bench: A Comprehensive and Unified Benchmark of Foundation Models for Time Series Forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* Association for Computing Machinery, New York, NY, USA, 5595–5606. doi:10.1145/3711896.3737442

[15] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024. AgentBench: Evaluating LLMs as Agents. In *The Twelfth International Conference on Learning Representations.* OpenReview.net, Vienna, Austria, 58 pages. https://openreview.net/forum?id=zAdUB0aCTQ

[16] Richard K. Lyons and Ganesh Viswanath-Natraj. 2023. What Keeps Stablecoins Stable? *Journal of International Money and Finance* 131 (2023), 102777. doi:10.1016/j.jimonfin.2022.102777

[17] Jaden Zhang, Gardenia Liu, Oliver Johansson, Hileamlak Yitayew, Kamryn Ohly, and Grace Li. 2026. Prediction Arena: Benchmarking AI Models on Real-World Prediction Markets. arXiv:2604.07355 [cs.LG] doi:10.48550/arXiv.2604.07355

[18] Luyao Zhang. 2025. SoK: Stablecoins for Digital Transformation: Design, Metrics, and Application with Real World Asset Tokenization as a Case Study. arXiv:2508.02403 [q-fin.GN] doi:10.48550/arXiv.2508.02403

[19] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2023. WebArena: A Realistic Web Environment for Building Autonomous Agents. arXiv:2307.13854 [cs.AI] doi:10.48550/arXiv.2307.13854

A Detailed StableEval Arena Protocol and Results

This appendix provides the audit trail behind the 120-case intermediate validation results. Its purpose is not only to add supporting tables, but to make the benchmark inspectable: how cases are constructed, what information agents receive, how outputs are parsed and calibrated, how baselines are defined, and how cost and reliability are measured. All reported agent results use the frozen validation protocol: 120 historical-replay cases, 30-day lookback windows, hidden 7-day prediction horizons, the revised risk-sensitive prompt packet, `qwen/qwen-2.5-72b-instruct` through OpenRouter, temperature 0, strict JSON outputs, and the same evaluator across platforms.

A.1 Glossary of Notation and Key Terms

Table 6: Glossary of key terms, abbreviations, and mathematical notation.

Term	Definition	Term	Definition
$\mathscr{D}$	Benchmark case set	ξ_i	One historical-replay case
$\mathscr{C}$	Target stablecoin universe	χ_i	Target stablecoin in case i
t_i	Observation time	$\mathscr{L}_i$	Frozen 30-day lookback window
$\mathscr{H}_i$	Hidden 7-day future horizon	$\boldsymbol{x}_{\chi,\tau}$	Feature vector at hour τ
$p_{\chi,\tau}$	Hourly close price	$\delta_{\chi,\tau}$	Peg deviation in basis points
Δ_i	Maximum hidden-horizon deviation	z_i	Strict sustained-depeg indicator
$\mathscr{Y}$	`stable`/`watch`/`stress` label space	ℓ_i	Hidden evaluator label
$\mathscr{A}$	Evaluated agent set	f_{θ_a}	Agent prediction function
$\widehat{s}_{a,i}$	Predicted stability label	$\widehat{\omega}_{a,i}$	Predicted depeg-risk probability
$\widehat{\Delta}_{a,i}$	Predicted max deviation	$\widehat{\phi}_{a,i}$	Predicted deviation direction
$\widehat{\kappa}_{a,i}$	Agent confidence score	$\Gamma_{a,i}$	Estimated inference cost
$\tau_{a,i}$	Latency	ρ_a	Valid-output rate
Ω_a	Failure rate	MDR/FAR	Missed-depeg and false-alarm rates

A.2 Dataset, Labels, and Validation Set

This section defines the benchmark data used for the validation results. Table 7 summarizes the conversion from hourly stablecoin market observations into historical-replay cases. The validation set contains 54 stable cases, 45 watch cases, and 21 stress cases. It includes all enhanced stress cases and all strict sustained-depeg cases available in the frozen pool, together with watch cases, near-stress stable controls, and natural stable controls. Hidden future-horizon labels are used only for case selection and evaluation; they are never included in the agent-facing packet.

Table 7: Dataset and case construction summary.

Component	Value
Stablecoins	USDT, USDC, DAI
Context assets	BTC, ETH
Source date range	2023-01-01 to 2026-05-01
Frequency	Hourly price/volume; daily market-structure metadata joined by date
Lookback	30 days
Prediction horizon	7 days
Full case pool	507 cases: 441 stable, 45 watch, 21 stress
Strict sustained-depeg cases	3 in the full frozen pool; all 3 are included in validation
Validation case mix	120 cases: 54 stable, 45 watch, 21 stress

Table 8: Stable/watch/stress and depeg label definitions.

Label or quantity	Definition
Deviation bps	$\lvert\text{hourly close} - 1.0\rvert \times 10000$
Stable	No meaningful hidden-horizon deviation under the enhanced label rule.

Label or quantity	Definition
Watch	Mild but meaningful hidden-horizon instability, including any hourly close deviation at or above 50 bps absent strict sustained depeg.
Stress	Large, persistent, or operationally meaningful instability.
Strict sustained depeg	Absolute hourly close deviation at or above 100 bps for at least three consecutive hours.

Table 9: 120-case validation label distribution.

Label	Cases
stable	54
watch	45
stress	21

A.3 Agentic Systems and Execution Metadata

This section documents the execution context for the evaluated agentic systems. StableEval Arena compares six platforms under a common model backend and shared prompt/evaluator protocol. Nanobot, LangGraph, Hermes AI, and EvoAgentX are treated as the main framework-level agent comparisons. OpenClaw and NanoClaw are included as adapter-based systems in this validation unless native CMDOP execution is separately verified. Because all platforms use the same LLM backend, the comparison focuses on platform-mediated execution, structured-output behavior, cost, latency, and reliability rather than independent foundation-model ability.

Table 10: StableEval Arena agent platform metadata.

Platform	Package	Native runtime	Execution mode	Caveat
Nanobot	nanobot-ai 0.2.0	True	Nanobot SDK OpenRouter agent loop	Real local Nanobot SDK framework path executed with OpenRouter LLM backend.
LangGraph	langgraph 1.2.0	True	Two-node StateGraph with OpenRouter LLM	Real local LangGraph StateGraph framework path executed with OpenRouter LLM backend.
Hermes AI	hermes-ai 0.3.20	True	Hermes Agent with LlamaIndex/OpenRouter	Real local Hermes Agent framework path executed with OpenRouter LLM backend.
EvoAgentX	evoagentx 0.1.0	True	EvoAgentX OpenRouter adapter	Real local EvoAgentX OpenRouter-backed workflow executed.
OpenClaw	openclaw 2026.3.20	False	StableEval adapter	CMDOP local runtime was not available; StableEval used an LLM-backed adapter, not native CMDOP execution.
NanoClaw	nanoclaw 2026.3.20	False	StableEval adapter	CMDOP local runtime was not available; StableEval used an LLM-backed adapter, not native CMDOP execution.

A.4 Leaderboard-Style Summary Figures

The following figures provide visual summaries of the main validation patterns before the full numeric tables. They are drawn directly from the reported validation metrics and use the same StableEval color palette as the main paper figures and tables.

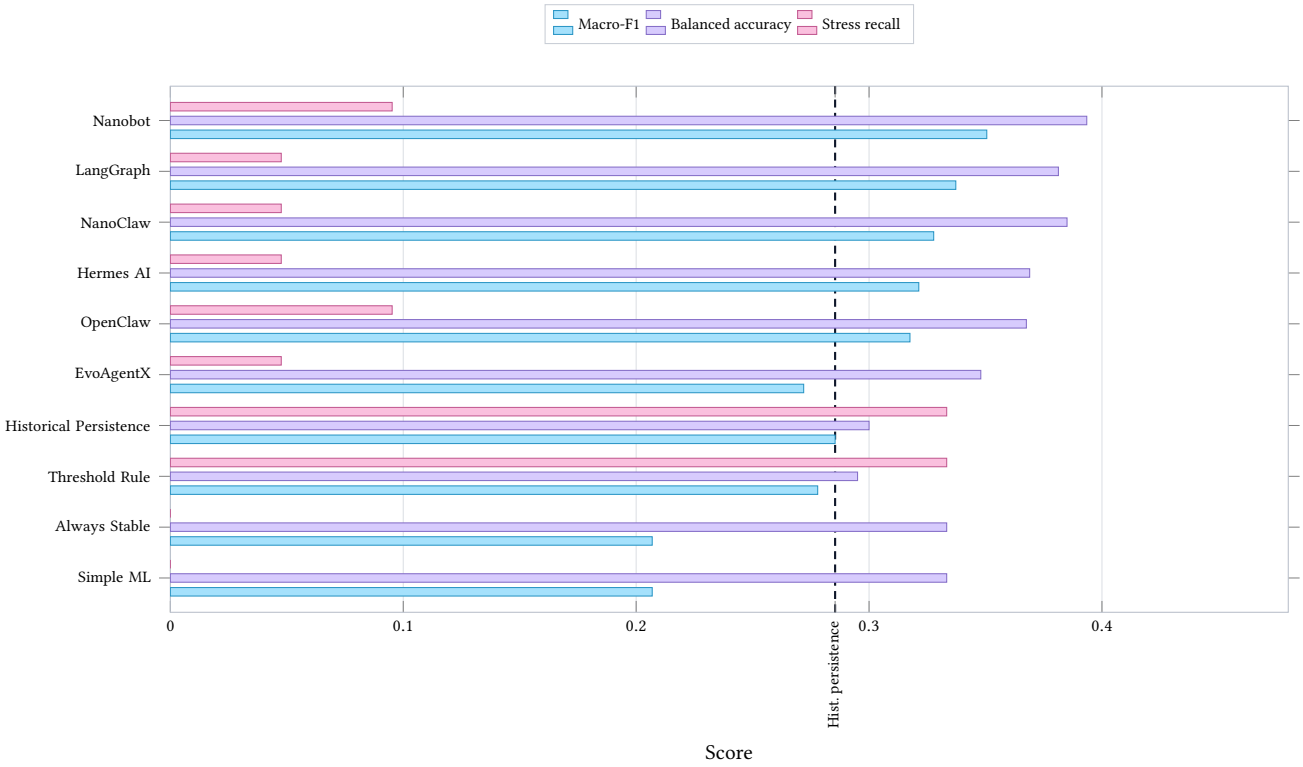


Figure 3: Validation leaderboard on the 120-case benchmark. Bars report Macro-F1, balanced accuracy, and stress recall from the calibrated first-pass protocol. The dashed reference line marks the Macro-F1 of the strongest non-agent baseline, historical persistence.

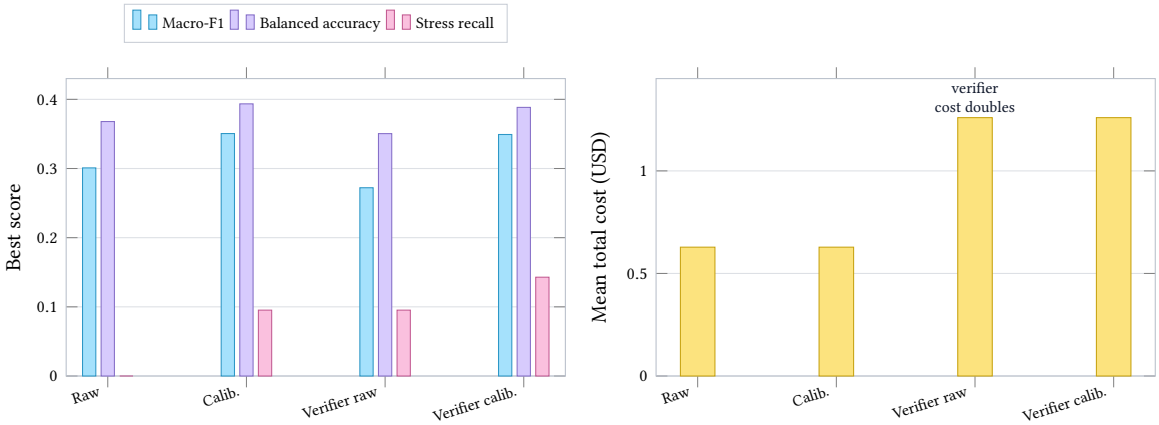


Figure 4: Ablation summary across raw first-pass, calibrated first-pass, and verifier protocols. Calibration improves the best aggregate scores, while the verifier increases stress recall but approximately doubles mean total cost.

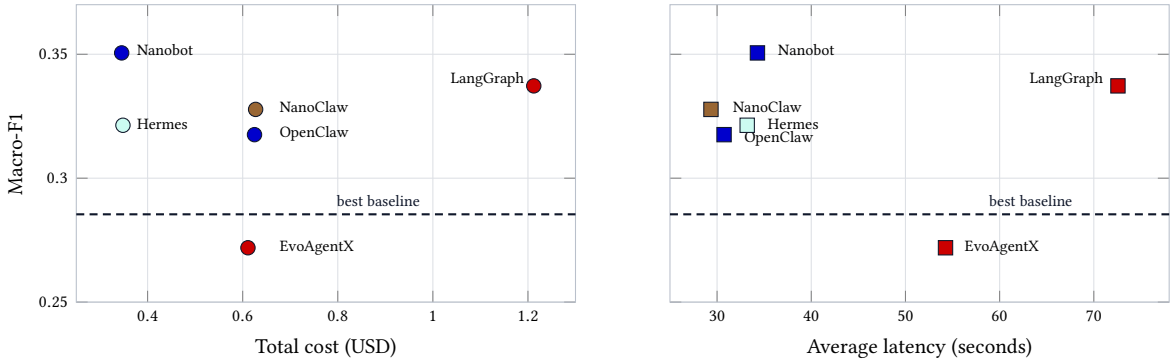


Figure 5: Cost-aware performance tradeoffs for the six agentic systems. Nanobot provides the strongest Macro-F1 at low cost, while LangGraph is competitive but more expensive and slower under the frozen validation protocol.

A.5 Validation Leaderboard Tables

The next tables report the main validation metrics in readable form. Tables 11 and 12 split the compact validation summary into predictive metrics and operational metrics. The wider cross-variant matrices are placed on explicit landscape pages so that the values remain readable without shrinking the text excessively.

Table 11: 120-case intermediate validation summary: predictive and risk metrics.

System	Type	Acc.	Macro-F1	BalAcc	Stress R	MDR	FAR
Nanobot	agent	0.442	0.351	0.393	0.095	1.000	0.009
LangGraph	agent	0.442	0.337	0.381	0.048	1.000	0.009
NanoClaw	agent	0.442	0.328	0.385	0.048	1.000	0.017
Hermes AI	agent	0.425	0.321	0.369	0.048	1.000	0.017
OpenClaw	agent	0.408	0.318	0.368	0.095	1.000	0.017
EvoAgentX	agent	0.392	0.272	0.348	0.048	1.000	0.017
Historical persistence	baseline	0.292	0.285	0.300	0.333	0.667	0.376
Threshold rule	baseline	0.283	0.278	0.295	0.333	1.000	0.282
Always stable	baseline	0.450	0.207	0.333	0.000	1.000	0.000
Simple ML logistic	baseline	0.450	0.207	0.333	0.000	1.000	0.000

Table 12: 120-case intermediate validation summary: continuous error and operational metrics.

System	Deviation MAE	Valid JSON	Failure	Cost USD
Nanobot	56.617	1.000	0.000	0.345
LangGraph	58.683	1.000	0.000	1.212
NanoClaw	66.358	1.000	0.000	0.627
Hermes AI	57.858	1.000	0.000	0.348
OpenClaw	62.000	1.000	0.000	0.625
EvoAgentX	66.917	1.000	0.000	0.611
Historical persistence	148.858	1.000	0.000	0.000
Threshold rule	148.890	1.000	0.000	0.000
Always stable	63.908	1.000	0.000	0.000
Simple ML logistic	113.500	1.000	0.000	0.000

Table 13: Appendix full leaderboard across validation variants: classification metrics.

Variant	System	Type	Acc.	Macro-F1	BalAcc	R stable	R watch	R stress
base-calibrated	Nanobot	agent	0.442	0.351	0.393	0.241	0.844	0.095
base-calibrated	LangGraph	agent	0.442	0.337	0.381	0.296	0.800	0.048
base-calibrated	NanoClaw	agent	0.442	0.328	0.385	0.241	0.867	0.048
base-calibrated	Hermes AI	agent	0.425	0.321	0.369	0.259	0.800	0.048
base-calibrated	OpenClaw	agent	0.408	0.318	0.368	0.185	0.822	0.095
base-calibrated	Historical persistence	baseline	0.292	0.285	0.300	0.278	0.289	0.333
base-calibrated	Threshold rule	baseline	0.283	0.278	0.295	0.241	0.311	0.333
base-calibrated	EvoAgentX	agent	0.392	0.272	0.348	0.130	0.867	0.048
base-calibrated	Always stable	baseline	0.450	0.207	0.333	1.000	0.000	0.000
base-calibrated	Simple ML logistic	baseline	0.450	0.207	0.333	1.000	0.000	0.000
base-raw	LangGraph	agent	0.433	0.301	0.368	0.259	0.844	0.000
base-raw	NanoClaw	agent	0.425	0.286	0.364	0.204	0.889	0.000
base-raw	Historical persistence	baseline	0.292	0.285	0.300	0.278	0.289	0.333
base-raw	Hermes AI	agent	0.425	0.281	0.365	0.185	0.911	0.000
base-raw	OpenClaw	agent	0.417	0.281	0.357	0.204	0.867	0.000
base-raw	Threshold rule	baseline	0.283	0.278	0.295	0.241	0.311	0.333
base-raw	Nanobot	agent	0.400	0.266	0.343	0.185	0.844	0.000
base-raw	EvoAgentX	agent	0.383	0.237	0.333	0.111	0.889	0.000
base-raw	Always stable	baseline	0.450	0.207	0.333	1.000	0.000	0.000
base-raw	Simple ML logistic	baseline	0.450	0.207	0.333	1.000	0.000	0.000
baselines-only	Historical persistence	baseline	0.292	0.285	0.300	0.278	0.289	0.333
baselines-only	Threshold rule	baseline	0.283	0.278	0.295	0.241	0.311	0.333
baselines-only	Always stable	baseline	0.450	0.207	0.333	1.000	0.000	0.000
baselines-only	Simple ML logistic	baseline	0.450	0.207	0.333	1.000	0.000	0.000
verifier-calibrated	Nanobot	agent	0.425	0.349	0.388	0.222	0.800	0.143
verifier-calibrated	Hermes AI	agent	0.417	0.348	0.380	0.241	0.756	0.143
verifier-calibrated	NanoClaw	agent	0.400	0.315	0.359	0.204	0.778	0.095
verifier-calibrated	LangGraph	agent	0.383	0.312	0.342	0.241	0.689	0.095
verifier-calibrated	Historical persistence	baseline	0.292	0.285	0.300	0.278	0.289	0.333
verifier-calibrated	Threshold rule	baseline	0.283	0.278	0.295	0.241	0.311	0.333
verifier-calibrated	OpenClaw	agent	0.358	0.277	0.326	0.148	0.733	0.095
verifier-calibrated	EvoAgentX	agent	0.358	0.266	0.328	0.111	0.778	0.095
verifier-calibrated	Always stable	baseline	0.450	0.207	0.333	1.000	0.000	0.000
verifier-calibrated	Simple ML logistic	baseline	0.450	0.207	0.333	1.000	0.000	0.000
verifier-raw	Historical persistence	baseline	0.292	0.285	0.300	0.278	0.289	0.333
verifier-raw	Threshold rule	baseline	0.283	0.278	0.295	0.241	0.311	0.333
verifier-raw	LangGraph	agent	0.383	0.272	0.341	0.130	0.844	0.048
verifier-raw	NanoClaw	agent	0.367	0.271	0.335	0.111	0.800	0.095
verifier-raw	EvoAgentX	agent	0.375	0.262	0.345	0.074	0.867	0.095
verifier-raw	Hermes AI	agent	0.392	0.261	0.350	0.093	0.911	0.048
verifier-raw	Nanobot	agent	0.375	0.253	0.336	0.093	0.867	0.048
verifier-raw	OpenClaw	agent	0.367	0.241	0.319	0.111	0.844	0.000
verifier-raw	Always stable	baseline	0.450	0.207	0.333	1.000	0.000	0.000
verifier-raw	Simple ML logistic	baseline	0.450	0.207	0.333	1.000	0.000	0.000

Table 14: Appendix full leaderboard across validation variants: risk, error, and cost metrics.

Variant	System	MDR	FAR	Brier	MAE	RMSE	Cost	Cost/correct
base-calibrated	Nanobot	1.000	0.009	0.047	56.617	167.172	0.345	0.007
base-calibrated	LangGraph	1.000	0.009	0.046	58.683	168.627	1.212	0.023
base-calibrated	NanoClaw	1.000	0.017	0.050	66.358	199.767	0.627	0.012
base-calibrated	Hermes AI	1.000	0.017	0.047	57.858	174.954	0.348	0.007
base-calibrated	OpenClaw	1.000	0.017	0.048	62.000	180.325	0.625	0.013
base-calibrated	Historical persistence	0.667	0.376	0.276	148.858	309.496	0.000	0.000
base-calibrated	Threshold rule	1.000	0.282	0.244	148.890	309.496	0.000	0.000
base-calibrated	EvoAgentX	1.000	0.017	0.050	66.917	199.181	0.611	0.013
base-calibrated	Always stable	1.000	0.000	0.025	63.908	148.390	0.000	0.000
base-calibrated	Simple ML logistic	1.000	0.000	0.024	113.500	368.523	0.000	0.000
base-raw	LangGraph	1.000	0.009	0.046	58.683	168.627	1.212	0.023
base-raw	NanoClaw	1.000	0.017	0.050	66.358	199.767	0.627	0.012
base-raw	Historical persistence	0.667	0.376	0.276	148.858	309.496	0.000	0.000
base-raw	Hermes AI	1.000	0.017	0.047	57.858	174.954	0.348	0.007
base-raw	OpenClaw	1.000	0.017	0.048	62.000	180.325	0.625	0.012
base-raw	Threshold rule	1.000	0.282	0.244	148.890	309.496	0.000	0.000
base-raw	Nanobot	1.000	0.009	0.047	56.617	167.172	0.345	0.007
base-raw	EvoAgentX	1.000	0.017	0.050	66.917	199.181	0.611	0.013
base-raw	Always stable	1.000	0.000	0.025	63.908	148.390	0.000	0.000
base-raw	Simple ML logistic	1.000	0.000	0.024	113.500	368.523	0.000	0.000
baselines-only	Historical persistence	0.667	0.376	0.276	148.858	309.496	0.000	0.000
baselines-only	Threshold rule	1.000	0.282	0.244	148.890	309.496	0.000	0.000
baselines-only	Always stable	1.000	0.000	0.025	63.908	148.390	0.000	0.000
baselines-only	Simple ML logistic	1.000	0.000	0.024	113.500	368.523	0.000	0.000
verifier-calibrated	Nanobot	1.000	0.017	0.052	59.992	168.163	0.978	0.019
verifier-calibrated	Hermes AI	1.000	0.017	0.051	60.608	176.005	0.982	0.020
verifier-calibrated	NanoClaw	1.000	0.017	0.054	71.158	200.912	1.257	0.026
verifier-calibrated	LangGraph	1.000	0.017	0.054	65.442	185.971	1.840	0.040
verifier-calibrated	Historical persistence	0.667	0.376	0.276	148.858	309.496	0.000	0.000
verifier-calibrated	Threshold rule	1.000	0.282	0.244	148.890	309.496	0.000	0.000
verifier-calibrated	OpenClaw	1.000	0.017	0.053	70.850	200.598	1.253	0.029
verifier-calibrated	EvoAgentX	1.000	0.017	0.053	69.217	199.638	1.246	0.029
verifier-calibrated	Always stable	1.000	0.000	0.025	63.908	148.390	0.000	0.000
verifier-calibrated	Simple ML logistic	1.000	0.000	0.024	113.500	368.523	0.000	0.000
verifier-raw	Historical persistence	0.667	0.376	0.276	148.858	309.496	0.000	0.000
verifier-raw	Threshold rule	1.000	0.282	0.244	148.890	309.496	0.000	0.000
verifier-raw	LangGraph	1.000	0.017	0.054	65.442	185.971	1.840	0.040
verifier-raw	NanoClaw	1.000	0.017	0.054	71.158	200.912	1.257	0.029
verifier-raw	EvoAgentX	1.000	0.017	0.053	69.217	199.638	1.246	0.028
verifier-raw	Hermes AI	1.000	0.017	0.051	60.608	176.005	0.982	0.021
verifier-raw	Nanobot	1.000	0.017	0.052	59.992	168.163	0.978	0.022
verifier-raw	OpenClaw	1.000	0.017	0.053	70.850	200.598	1.253	0.028
verifier-raw	Always stable	1.000	0.000	0.025	63.908	148.390	0.000	0.000
verifier-raw	Simple ML logistic	1.000	0.000	0.024	113.500	368.523	0.000	0.000

Table 15: Cost and reliability summary.

System	Type	Valid JSON	Failure	Avg. latency	Avg. tokens	Cost/case	Total cost
Nanobot	agent	1.000	0.000	34.287	7968.908	0.003	0.345
Hermes AI	agent	1.000	0.000	33.198	8031.433	0.003	0.348
EvoAgentX	agent	1.000	0.000	54.254	14096.733	0.005	0.611
OpenClaw	agent	1.000	0.000	30.742	13989.475	0.005	0.625
NanoClaw	agent	1.000	0.000	29.351	14029.575	0.005	0.627
LangGraph	agent	1.000	0.000	72.562	27529.942	0.010	1.212
Historical persistence	baseline	1.000	0.000	0.000	0.000	0.000	0.000
Threshold rule	baseline	1.000	0.000	0.000	0.000	0.000	0.000
Always stable	baseline	1.000	0.000	0.000	0.000	0.000	0.000
Simple ML logistic	baseline	1.000	0.000	0.000	0.000	0.000	0.000

A.6 Agent-Baseline Comparison

This section isolates the comparison between agentic systems and non-agent baselines. The main pattern is intentionally mixed: calibrated agents outperform the strongest baseline on aggregate Macro-F1, but the historical-persistence baseline remains stronger on stress recall. This distinction is important because StableEval evaluates both average warning classification and rare-event trustworthiness.

Table 16: Agent and baseline comparison on the 120-case validation set.

System	Type	Macro-F1	BalAcc	Stress R	MDR	FAR	Brier
Nanobot	agent	0.351	0.393	0.095	1.000	0.009	0.047
LangGraph	agent	0.337	0.381	0.048	1.000	0.009	0.046
NanoClaw	agent	0.328	0.385	0.048	1.000	0.017	0.050
Hermes AI	agent	0.321	0.369	0.048	1.000	0.017	0.047
OpenClaw	agent	0.318	0.368	0.095	1.000	0.017	0.048
EvoAgentX	agent	0.272	0.348	0.048	1.000	0.017	0.050
Historical persistence	baseline	0.285	0.300	0.333	0.667	0.376	0.276
Threshold rule	baseline	0.278	0.295	0.333	1.000	0.282	0.244
Always stable	baseline	0.207	0.333	0.000	1.000	0.000	0.025
Simple ML logistic	baseline	0.207	0.333	0.000	1.000	0.000	0.024

Table 17: Stress recall and missed-depeg metrics.

System	Type	Stress R	MDR	FAR	Binary TP	Binary FN
Historical persistence	baseline	0.333	0.667	0.376	1	2
Threshold rule	baseline	0.333	1.000	0.282	0	3
Nanobot	agent	0.095	1.000	0.009	0	3
OpenClaw	agent	0.095	1.000	0.017	0	3
LangGraph	agent	0.048	1.000	0.009	0	3
NanoClaw	agent	0.048	1.000	0.017	0	3
Hermes AI	agent	0.048	1.000	0.017	0	3
EvoAgentX	agent	0.048	1.000	0.017	0	3
Always stable	baseline	0.000	1.000	0.000	0	3
Simple ML logistic	baseline	0.000	1.000	0.000	0	3

A.7 Calibration, Verifier, and Anonymization Protocols

This section documents the post-processing and audit protocols used around the agent outputs. The calibrated-label variant applies a frozen rule defined before the 120-case intermediate validation. It changes only the mapped class label from the model's own predicted depeg probability and predicted maximum absolute deviation; it does not modify raw model outputs, validation labels, future prices, or hidden horizon outcomes. The verifier ablation adds one risk-auditor call that receives the same lookback packet and first prediction, and may revise labels or probabilities only from lookback-window evidence.

Table 18: Frozen calibrated-label and verifier rules.

Component	Rule
Stress rule 1	Predict `stress` if `predicted_max_abs_deviation_bps` ≥ 100.

Component	Rule
Stress rule 2	Predict `stress` if `p_depeg_next_7d` ≥ 0.35 and `predicted_max_abs_deviation_bps` ≥ 75.
Watch rule	Predict `watch` if `p_depeg_next_7d` ≥ 0.15 or `predicted_max_abs_deviation_bps` ≥ 40.
Stable rule	Predict `stable` otherwise.
Verifier input	Same 30-day lookback packet plus the first prediction.
Verifier constraint	May revise only from lookback-window evidence; no future data, hidden labels, or web browsing.

To probe possible parametric-memory leakage, the anonymization ablation compares predictions on the same numerical evidence with real stablecoin names and calendar dates versus anonymized coin aliases and relative time labels. This is an audit rather than proof of no leakage.

Table 19: Anonymization ablation summary on matched numerical evidence.

System	Cases	Label change	Mean p diff.	Mean bps diff.
EvoAgentX	12	0.1667	0.01667	7.917
Hermes AI	12	0.08333	0.02917	4.583
LangGraph	12	0.3333	0.06250	8.417
Nanobot	12	0.2500	0.04167	8.500
NanoClaw	12	0.2500	0.05000	14.170
OpenClaw	12	0.08333	0.02917	8.333

A.8 Ablation Results

The ablation study compares raw first-pass outputs, calibrated first-pass outputs, and verifier-calibrated outputs. Calibration improves Macro-F1 by mapping model-reported probabilities and predicted maximum deviations into the final three-class label. The verifier layer modestly improves stress recall, but it increases cost and does not improve the best aggregate Macro-F1. This makes the ablation useful as a trustworthiness diagnostic rather than a simple performance upgrade.

Table 20: Validation ablation summary.

Variant	Best Macro-F1	Best BalAcc	Best Stress R	Mean cost
base-calibrated	0.351	0.393	0.095	0.628
base-raw	0.301	0.368	0.000	0.628
verifier-calibrated	0.349	0.388	0.143	1.260
verifier-raw	0.272	0.350	0.095	1.260

A.9 Thirty-Case Pilot Context

The 30-case paid pilot is included only as design context, not as the final validation result. It motivated the improved prompt, richer input packet, calibration rule, and verifier ablation later tested on the 120-case validation set. Reporting it here makes the experimental sequence transparent while keeping the main claims tied to the larger frozen validation protocol.

Table 21: 30-case improved pilot summary: predictive and risk metrics.

System	Type	Acc.	Macro-F1	BalAcc	Stress R	MDR	FAR
OpenClaw	agent	0.500	0.434	0.500	0.100	1.000	0.000
Nanobot	agent	0.433	0.347	0.433	0.100	1.000	0.037
NanoClaw	agent	0.400	0.344	0.400	0.100	1.000	0.037
EvoAgentX	agent	0.400	0.330	0.400	0.100	1.000	0.037
LangGraph	agent	0.400	0.301	0.400	0.000	1.000	0.000
Hermes AI	agent	0.367	0.278	0.367	0.100	1.000	0.037
Historical persistence	baseline	0.333	0.333	0.333	0.200	0.667	0.370
Threshold rule	baseline	0.300	0.298	0.300	0.200	1.000	0.222
Simple ML logistic	baseline	0.300	0.260	0.300	0.000	1.000	0.000
Always stable	baseline	0.333	0.167	0.333	0.000	1.000	0.000

Table 22: 30-case improved pilot summary: continuous error and operational metrics.

System	Deviation MAE	Valid JSON	Failure	Cost USD
OpenClaw	117.967	1.000	0.000	0.151
Nanobot	153.133	1.000	0.000	0.087
NanoClaw	128.300	1.000	0.000	0.154
EvoAgentX	155.367	1.000	0.000	0.152
LangGraph	116.300	1.000	0.000	0.298
Hermes AI	155.467	1.000	0.000	0.087
Historical persistence	198.167	1.000	0.000	0.000
Threshold rule	197.975	1.000	0.000	0.000
Simple ML logistic	142.760	1.000	0.000	0.000
Always stable	140.767	1.000	0.000	0.000

A.10 Reproducibility Artifacts

This appendix is designed to function as an audit layer for the main paper. Rather than asking readers to trust only the reported tables, StableEval Arena preserves the intermediate artifacts needed to trace each reported result from case construction, to agent execution, to calibrated scoring, to final paper-facing metrics. The release package therefore prioritizes the artifacts that make the benchmark inspectable: frozen case lists, prompt packets, model and platform metadata, raw predictions, calibration rules, evaluator outputs, and cross-variant summaries.

Table 23: Main reproducibility artifacts preserved for StableEval Arena.

Artifact	Purpose
`data/benchmark/full_case_labels_507.csv`	Full frozen 507-case label file used as the source label table for benchmark evaluation.
`data/benchmark/full_507_prompt_packets.jsonl`	Full prompt-packet file containing lookback-only evidence and excluding hidden future labels.
`results/validation_120/validation_case_list.csv`	Selected 120-case intermediate validation set used for the main reported validation results.
`prompts/risk_sensitive_agent_prompt.md`	Frozen risk-sensitive agent prompt template shared across evaluated platforms.
`results/validation_120/model_backend.json`	Frozen model-backend metadata for the reported validation protocol.
`results/validation_120/calibration_rule.md`	Frozen calibrated-label mapping used to convert model risk estimates into stable/watch/stress labels.
`results/validation_120/agent_execution_metadata.csv`	Platform package, execution-mode, native-runtime, and adapter metadata.
`results/validation_120/predictions_*`	Saved raw and parsed predictions for each validation variant.
`results/validation_120/evaluation_*`	Evaluator outputs, including prediction quality, cost, reliability, and error metrics.
`results/validation_120/validation_variant_comparison.csv`	Cross-variant summary used for calibration and verifier ablation reporting.

Together, these artifacts make the benchmark claims traceable at the level most important for review: what cases were used, what information agents received, what each platform returned, how labels were calibrated, how metrics were computed, and how cost and reliability were recorded. Raw vendor snapshots are intentionally excluded from the release package by default; the reproducibility package instead preserves processed case metadata, labels, prompt packets, saved predictions, scripts, and aggregate evaluation outputs. In this sense, the appendix supports the paper's central claim: StableEval Arena is not only a leaderboard, but an auditable evaluation standard for cost-aware and trustworthy financial agents.