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

Sean Wan[†]
Duke Kunshan University
Suzhou, China

Dongping Liu[§]
Tenorshare
Hong Kong, China

Luyao Zhang[†‡]
Duke Kunshan University
Suzhou, China

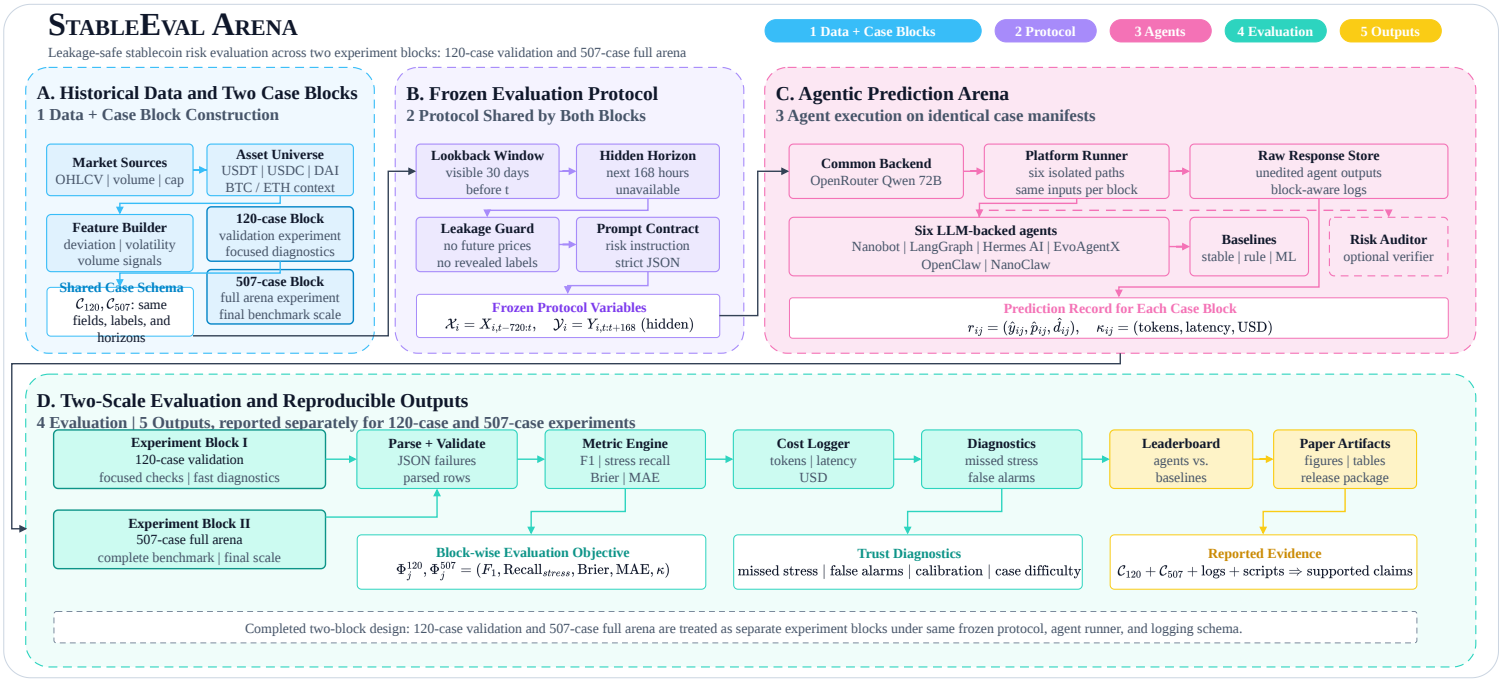

Figure 1: Overview of StableEval Arena.

Abstract

We introduce StableEval Arena, a cost-aware benchmark framework for evaluating agentic AI systems on stablecoin peg-risk prediction. StableEval Arena evaluates LLM-backed agentic systems on diagnosing peg stress and forecasting deviations from the one-dollar peg over a hidden seven-day horizon, using leakage-safe historical replay with exchange price-volume data and market-context features. We report two complementary experiment blocks: a 120-case stress-enriched validation block and a 507-case natural-distribution full-arena evaluation block. Across six LLM-backed agent configurations and baselines, StableEval Arena measures prediction quality, calibrated-label behavior, structured-output reliability, latency, token consumption, and estimated inference cost. Rather than ranking agents by accuracy alone, the framework treats trustworthiness as a joint property of forecast quality, operational reliability, and computational cost. The results show a gap between protocol-following reliability and financial-risk reliability: agents reliably produce valid structured outputs at modest measured cost, but still miss most rare severe-stress and sustained-depeg cases. To support auditing and replication, we release the benchmark dataset on Hugging Face and the source code on GitHub.

Keywords

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

1 Introduction

Stablecoins are core digital finance infrastructure for trading, lending, payments, collateral, liquidity, and settlement. Their utility rests on a fragile promise: a dollar-pegged stablecoin should remain near one U.S. dollar. When the promise weakens, applications may face pricing errors, liquidity stress, payment failure, and settlement uncertainty. Recent stablecoin research therefore calls for transparent metrics, stakeholder-oriented evaluation, and reproducible benchmarks [26]. Yet stablecoin risk assessment differs from ordinary cryptocurrency forecasting: the goal is not to predict the full future price path, but to judge whether a stablecoin will remain stable, enter a warning state, or experience peg stress over a fixed

[†]**Acknowledgments**: Sean Wan is grateful for the support from the Summer Research Scholar Program at Duke Kunshan University, supervised by Prof. Luyao Zhang.
[§]Dongping Liu contributed to this research in his personal capacity and as an independent scholarly endeavor. Tenorshare did not provide funding, resources, institutional support, or endorsement for this work.
[‡]The Corresponding author: Email: lz183@duke.edu, Digital Innovation Research Center and Social Science Division, Duke Kunshan University, Address: Duke Avenue No.8, Kunshan, Suzhou, Jiangsu, China, 215316.

future window. Such judgments require bounded evidence about peg deviation, volatility, volume, market context, and stablecoin mechanisms. Cryptocurrency-forecasting work highlights multimodal financial time-series inputs, including market metrics and sentiment signals [11], but stablecoin evaluation must separate normal peg noise from rare, high-consequence stress events.

This evaluation problem becomes more demanding as agentic AI systems enter financial decision-support workflows, including autonomous prediction-market trading [25]. These systems increasingly support reasoning, tool use, workflow execution, and decision support, yet trustworthy evaluation requires more than final-answer accuracy. Benchmarks must also measure protocol compliance, output validity, runtime cost, reliability, and failure modes. Recent agent and prediction-market benchmarks motivate standardized execution, fixed budgets, process instrumentation, and efficiency metrics for autonomous systems [3, 25]. Related forecasting and tabular benchmarks emphasize unified evaluation protocols, public leaderboards, reproducible code, and maintained benchmark infrastructure [8, 19]. However, existing benchmarks do not directly test whether agentic AI systems can serve as stablecoin risk-assessment agents under cost and reliability constraints.

We address this gap with StableEval Arena, a historical-replay benchmark for evaluating agentic stablecoin risk assessment. Each case provides a frozen 30-day lookback window and asks for a structured next-7-day risk assessment: stable/watch/stress label, depeg probability, maximum peg-deviation estimate, direction, confidence, and rationale. We compare agentic systems and non-agent baselines under one protocol, measuring prediction quality, stress recall, missed-depeg risk, calibration, valid-output rate, latency, token usage, and inference cost. The benchmark has two settings: a 120-case stress-enriched validation block for stress diagnosis and a 507-case natural-distribution scaling block for full-pool evaluation.

We organize the study around three research questions:

- **RQ1 (comparative performance).** Under a shared historical-replay protocol, do agentic AI frameworks improve stablecoin risk forecasts over simple baselines, or reproduce same errors?
- **RQ2 (reliability under rare stress).** When depeg stress is uncommon, how do systems trade off missed stress events, false alarms, calibration, structured-output validity, latency, token usage, and inference cost?
- **RQ3 (benchmark sensitivity).** What does comparing a stress-enriched validation block with a natural-distribution scaling block reveal that a single aggregate leaderboard would obscure?

The results show a gap between procedural reliability and financial-risk detection. The benchmark is diagnostic rather than agent-promotional: agents produce valid structured outputs under shared protocol, but do not consistently outperform simple baselines. In stress-enriched validation block, calibrated agents improve aggregate Macro-F1 relative to simple baselines; in the 507-case natural-distribution block, simple baselines remain strong and raw accuracy is shaped by stable-dominant class balance. Across both settings, rare-stress and sustained-depeg detection remain weak. To support replication, we release the dataset on Hugging Face[1] and the source code on GitHub[2] as open-access resources.

[1] *https://huggingface.co/datasets/SeanWan05/stableeval-arena*
[2] *https://github.com/SeanWan514/StableEval-Arena*

2 Background and Related Work

2.1 Stablecoin Risk Monitoring

Stablecoin risk monitoring asks whether a token's peg is supported by mechanisms that remain reliable under stress, not whether its recent price has stayed close to par. Prior work shows that stablecoin stability depends on market design and arbitrage access [21], reserve or collateral quality, redemption mechanisms, liquidity, user confidence, and regulatory design [2, 13, 16]. These factors matter because a token can appear stable in ordinary conditions while still being exposed to liquidity shocks, redemption frictions, or confidence-driven runs. Monitoring therefore requires distinguishing routine peg noise from warning or stress conditions that may affect payments, liquidation thresholds, collateral valuation, and settlement reliability. Recent work further argues that stablecoins should be evaluated as programmable financial infrastructure shaped by stakeholder needs, governance, regulation, and transparent benchmarking [26]. This motivates StableEval Arena's framing of monitoring as an evidence-bounded risk-evaluation task: given limited information, a system should identify salient risk signals, classify the severity of peg conditions, and produce reproducible assessments grounded in the available evidence.

2.2 Agentic Financial Prediction and Evaluation

Agentic financial prediction extends beyond conventional time-series forecasting because the system is often asked to synthesize bounded evidence into a structured judgment that can support downstream decisions. Time-series benchmarks emphasize fixed protocols, standardized dataset handling, and comparable evaluation settings [19], while cryptocurrency forecasting research shows that multimodal evidence can add value beyond price-only modeling [11]. In financial-risk settings, however, prediction quality alone is insufficient: the reliability of the decision process also depends on whether the system follows instructions, respects the evaluation protocol, produces valid structured outputs, and behaves consistently under resource constraints. This concern connects financial prediction to agent-evaluation work, where benchmarks such as AgentBench highlight reasoning, decision-making, and instruction following as persistent challenges [20], and Agent2 RL-Bench emphasizes fixed-budget evaluation, runtime recording, failure diagnosis, and post-hoc analysis [3]. These strands motivate evaluating agentic financial systems not only by predictive performance, but also by protocol compliance, structured-output reliability, failure behavior, latency, and resource use.

2.3 Cost-Aware and Trustworthy Benchmarking

Cost-aware and trustworthy benchmarking asks whether a system is reliable under the operational conditions in which its predictions would be used, not only whether it achieves high average accuracy. This distinction is important in financial settings, where a model can appear strong on aggregate metrics while missing rare but consequential stress events, producing malformed outputs, incurring excessive latency, or failing to reproduce results under a fixed protocol. Recent work on prediction-market agents reflects this broader view: Prediction Arena evaluates autonomous AI models using returns and win rates alongside token usage, cycle time, settlement behavior, exit patterns, and cross-platform differences [25].

Table 1: Agentic Systems Selected for StableEval Arena Evaluation

This table summarizes the six agentic systems included as StableEval Arena comparators, focusing on implementation role, deployment interface, memory/persistence, orchestration, and control features relevant to benchmark execution.

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

Note. Rows summarize system type, access, interface, memory, orchestration, and safety/control dimensions used to characterize comparable agentic AI systems. Categories reflect how recent surveys describe LLM-based agents as systems combining planning, memory, tool use, feedback, and evaluation/control.
Selection. Nanobot, LangGraph, Hermes AI, and EvoAgentX represent native/framework-level agent designs; OpenClaw and NanoClaw are treated as adapter-based OpenClaw-family comparators.
Source tags. NB-GH: Nanobot GitHub; LG-GH/LG-Docs: LangGraph GitHub/docs; HA-GH: Hermes Agent GitHub; EAX-GH: EvoAgentX GitHub; Wang et al. 2025: *EvoAgentX*; OC-GH: OpenClaw GitHub; NC-GH: NanoClaw GitHub.

For stablecoin risk monitoring, this perspective suggests that evaluation should account for both decision quality and the cost and reliability of obtaining that decision, especially when agents are queried repeatedly across assets and time windows.

Trustworthiness also requires diagnostics sensitive to imbalanced risk settings. In stable-dominant domains, high accuracy may reflect majority-class behavior while rare depeg or stress cases remain undetected. Evaluation therefore benefits from reporting stress recall, missed-depeg rates, false alarms, calibration, maximum-deviation error, inference cost, latency, and failure modes alongside aggregate performance. Benchmark-infrastructure work emphasizes reproducible, transparent documentation: TabArena highlights curated datasets, public code, leaderboard-style reporting, and explicit limitations [8], while Datasheets for Datasets motivates reporting dataset provenance, composition, intended uses, limitations, maintenance, and risks or harms [12]. These principles provide the background for StableEval Arena benchmark, where financial-agent evaluation should report prediction quality, cost, reproducibility, and stress-specific diagnostics.

3 StableEval Arena

3.1 Overview

StableEval Arena is a historical-replay benchmark for trustworthy, cost-aware evaluation of agentic AI systems in stablecoin risk assessment. Specifically, each system receives evidence available at an observation time, and must produce a structured next-window risk assessment: whether the target stablecoin remains stable, enters a warning state, or experiences peg stress, together with its estimated depeg probability, maximum dollar-peg deviation, deviation direction, confidence, and rationale. The benchmark is designed to support fair comparison, financial relevance, and auditability: all systems receive same agent-facing inputs and output schema; labels correspond to short-horizon stablecoin risk; each run records outputs, cost, latency, failures, and reproducibility metadata.

The completed StableEval Arena benchmark uses two complementary evaluation settings. We refer to the full natural-distribution setting as StableEval-507-Natural: a 507-case historical-replay pool used for robustness and full-pool evaluation. We refer to the stress-enriched diagnostic setting as StableEval-120-Enriched: a 120-case subset that concentrates rare watch and stress outcomes for sharper stress analysis. StableEval-507-Natural contains 441 stable, 45 watch, and 21 stress cases, while StableEval-120-Enriched contains 54 stable, 45 watch, and 21 stress cases. Thus, the natural distribution benchmark is centered on StableEval-507-Natural, and stress diagnosis is centered on StableEval-120-Enriched. As summarized in Figure 2, the two settings are complementary rather than interchangeable. StableEval-507-Natural

tests behavior under natural class balance, whereas StableEval-120-Enriched tests whether systems respond to sparse but consequential stress cases.

3.2 Task Formulation

Each benchmark case in StableEval Arena is defined by a target stablecoin, an observation time t, a frozen 30-day lookback window with hourly data, and a hidden 7-day prediction horizon. The target stablecoins are USDT, USDC, and DAI. The lookback window contains 720 hours of evidence, and the hidden horizon contains the next 168 hours. Notice that the agent is not asked to reproduce the full future hourly price path. Instead, it predicts a structured next-window risk assessment: stable/watch/stress class, depeg probability, maximum peg-deviation estimate, direction, confidence, and rationale.

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. Therefore, 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 lookback evidence window, and $\mathscr{H}_i$ is the hidden future horizon. The two temporal windows are

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

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

Here $\boldsymbol{x}_{\chi,\tau} \in \mathbb{R}^d$ 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 system 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 system, 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. Thus, StableEval Arena evaluates agentic decision systems rather than trained time-series forecasters: systems are scored by accuracy, schema validity, calibration, rare-stress behavior, cost, latency, and reliability.

3.3 Case Library and 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 derives risk-sensitive features such as latest close price, latest peg deviation, recent 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. All agent-facing features are computed only from the 30-day (720-hours) lookback window.

As defined above, StableEval-507-Natural is the full case library, while StableEval-120-Enriched is a stress-enriched diagnostic subset drawn from it. The subset retains all watch and stress cases and adds stable controls, so stress behavior can be examined without changing the underlying historical-replay protocol. This design lets the benchmark report both natural-distribution results and concentrated stress diagnostics. Hidden future labels and horizon statistics are used only for case selection and evaluation; they are never included in the agent-facing prompt packets.

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

$$z_i = \mathbf{1}\left\{\exists \tau : \tau, \tau+1, \tau+2 \in \mathscr{H}_i \text{ and } \min_{k \in \{0,1,2\}} \delta_{\chi_i,\tau+k} \geq 100\right\}.$$

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 rule avoids labeling ordinary peg noise as stress while preserving sensitivity to rare but consequential deviations.

3.4 Agent Systems

StableEval Arena evaluates the heterogeneous agentic systems summarized in Table 1 under a shared task interface. The systems of native/framework-level are Nanobot, LangGraph, Hermes AI, and EvoAgentX. These systems form the main agentic AI comparison in the 507-case study. OpenClaw and NanoClaw are using StableEval adapter executions. Since verified local CMDOP execution was not available for OpenClaw and NanoClaw, their results are interpreted as adapter-based -Claw extensions that center on testing protocol compatibility, output validity, and cost logging under the same StableEval interface.

The benchmark also includes non-agent baselines: an always-stable baseline, a historical-persistence baseline, a threshold-rule baseline, and a simple machine-learning baseline where feasible. Baselines are included to prevent agent-only comparison and to test whether agentic execution adds value beyond simple rules that exploit the strong base-rate stability of major stablecoins. Both StableEval-120-Enriched and StableEval-507-Natural use the same system grouping: non-agent baselines, four native/framework-level agents, and two adapter-based -Claw extensions. Each reported result includes an execution-mode label, allowing the executions to be interpreted accurately.

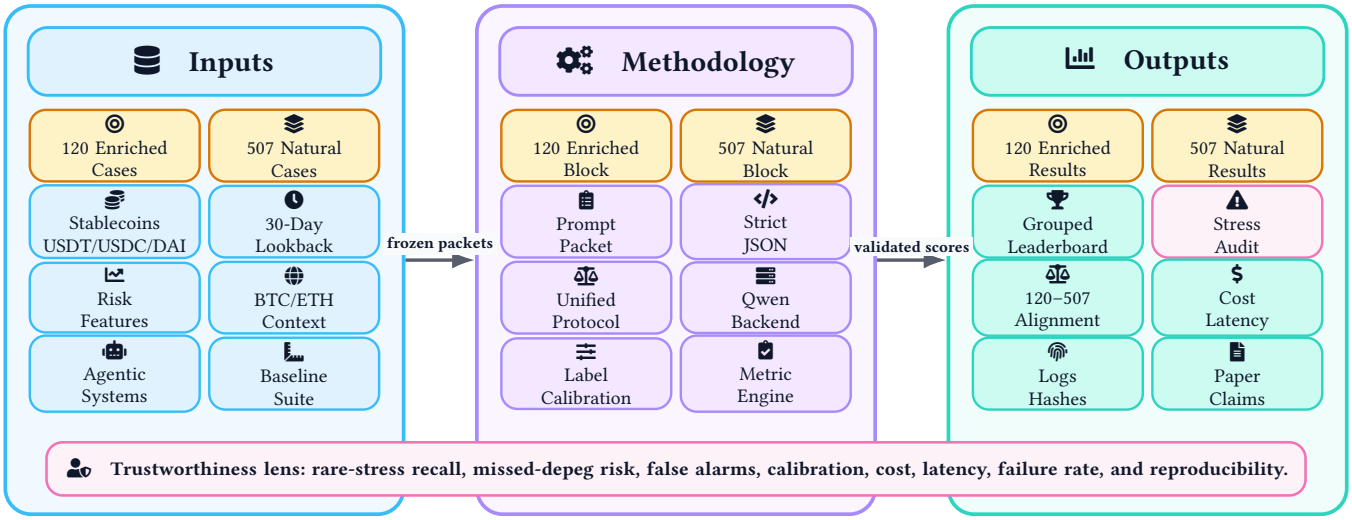

Figure 2: Experiment design of StableEval Arena: inputs, evaluation protocol, and reported outputs.

3.5 Evaluation Protocol, Outputs, and Metrics

All systems are evaluated under a unified protocol. They receive the same case packets, use the same prompt contract, follow the same strict JSON schema, and are parsed and scored by the same evaluator. The common backend is `qwen/qwen-2.5-72b-instruct` via OpenRouter with temperature 0. Web browsing is disabled, and prompts instruct systems to use only the provided lookback-window evidence. The protocol is fixed across the 120-case stress-enriched validation and the 507-case natural-distribution study.

The JSON output schema requires the predicted class $\widehat{s}_{a,i}$, depeg-risk probability $\widehat{\omega}_{a,i}$, maximum deviation estimate $\widehat{\Delta}_{a,i}$, deviation direction $\widehat{\phi}_{a,i}$, confidence $\widehat{\kappa}_{a,i}$, and rationale $r_{a,i}$. A pre-declared calibration rule maps model-reported probability and predicted deviation into the final stable/watch/stress label. Raw and calibrated outputs are retained separately. Accuracy is reported, but it is not the primary measure under the stable-dominant 507-case distribution; Macro-F1, balanced accuracy, stress recall, missed-depeg rate, calibration, and deviation error are the primary reported metrics.

Let $N = |\mathscr{D}|$ and let $\widehat{s}_{a,i}$ and ℓ_i denote the predicted and hidden labels for system 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.

The benchmark also evaluates probabilistic and continuous predictions. Depeg probability is scored with Brier score, and calibration is assessed following modern neural-network calibration practice [14]. Predicted maximum deviation is evaluated using mean absolute error and root mean squared error in basis points:

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

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

3.6 Cost, Reliability, and Reproducibility

Since StableEval Arena evaluates agentic systems in a risk-sensitive setting, it records operational behavior in addition to prediction quality. For every agent-case run, the benchmark logs 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 distinguish predictive errors from malformed outputs and execution failures.

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

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

The 120-case validation artifacts are retained as historical validation outputs. The 507-case scaling study is additive: it uses the same prompt packets, backend, output schema, calibration rule,

evaluator, and metric definitions, while writing agent-specific outputs, logs, metadata, and reports under the scaling artifact tree. Raw and calibrated predictions are preserved separately, and the canonical 507 leaderboard records execution-mode grouping for baselines, native/framework-level agents, and adapter-based -Claw extensions.

For reproducibility, the benchmark preserves case lists, prompt packets, prompt templates, leakage checks, calibration rules, execution metadata, raw predictions, calibrated predictions, cost logs, evaluation scripts, and versioned leaderboards. This release-oriented design follows representative benchmark and dataset-documentation practice in which public code, documented data artifacts, transparent protocols, and reproducible evaluation outputs are part of the contribution itself [8, 12, 19]. The purpose is not only to rank agents, but to make agentic AI evaluation auditable: a system that produces valid low-cost outputs but misses rare stress events exposes a financial-risk failure mode rather than a deployment-ready financial monitor.

4 Results

Table 2 reports results for the two benchmark blocks defined above: StableEval-120-Enriched for rare-risk diagnostic validation, and StableEval-507-Natural for full-pool robustness and scaling. To keep comparisons clear and consistent, both result blocks separate baselines, native/framework-level agents, and the adapter-based -Claw variants. We emphasize Macro-F1, balanced accuracy, stress recall, missed-depeg rate, valid-output reliability, and cost, since raw accuracy alone is insufficient under stablecoin class imbalance.

4.1 120-Case Stress-Enriched Validation

StableEval-120-Enriched evaluates system behavior under a stress-enriched distribution, with 45 watch and 21 stress cases providing sharper rare-risk diagnostic coverage than the natural 507-case pool. Among baselines, historical persistence is strongest, with Macro-F1 of 0.285 and balanced accuracy of 0.300. The always-stable and simple logistic baselines both reach accuracy of 0.450, but only Macro-F1 of 0.207, demonstrating why accuracy alone is not an adequate metric for stable/watch/stress evaluation.

Among native/framework-level agents, Nanobot performs best under the calibrated first-pass protocol, with accuracy of 0.442, Macro-F1 of 0.351, and balanced accuracy of 0.393. LangGraph follows with Macro-F1 of 0.337, while Hermes AI and EvoAgentX obtain Macro-F1 scores of 0.321 and 0.272. For adapter-based -Claw group, NanoClaw obtains Macro-F1 of 0.328 and OpenClaw obtains 0.318. Overall, five of the six agentic systems exceed the historical-persistence baseline on Macro-F1, indicating that calibrated agentic outputs improve aggregate three-class classification in the stress-enriched setting.

Class-level results further qualify this aggregate improvement. Agent gains are concentrated more in stable/watch discrimination than in severe stress detection. Nanobot and OpenClaw obtain the highest calibrated first-pass agent stress recall, at 0.095, while historical persistence reaches 0.333. Thus, StableEval-120-Enriched implies agentic systems can improve aggregate warning classification, yet still struggle with the rarest and most consequential stress outcomes. This distinction is central to the trustworthiness motivation of the benchmark. The 120-case validation does not support a claim that current agents reliably solve stablecoin stress prediction; instead, it demonstrates that a unified agentic evaluation protocol can distinguish aggregate classification performance from stress reliability. Thus, the 120-case block remains useful for diagnosing stress behavior that is diluted in the natural 507-case distribution.

4.2 507-Case Natural-Distribution Scaling

StableEval-507-Natural applies the same protocol to the full natural-distribution pool, where 441 of 507 cases are stable and raw accuracy is therefore especially misleading. The always-stable baseline reaches accuracy of 0.870, but has zero watch recall and zero stress recall. The strongest overall baseline is simple ML/logistic, with Macro-F1 of 0.4019, balanced accuracy of 0.4930, and stress recall of 0.3810. Historical persistence also remains strong, with Macro-F1 of 0.3867, balanced accuracy of 0.4500, stress recall of 0.3333, and missed-depeg rate of 0.6667.

Among the agentic systems, LangGraph is strongest on the calibrated 507 leaderboard, with Macro-F1 of 0.3044 and balanced accuracy of 0.4404, followed closely by NanoClaw adapter (Macro-F1 0.3041, balanced accuracy 0.4352) and Hermes AI (Macro-F1 0.3011, balanced accuracy 0.4328). Nanobot obtains Macro-F1 of 0.2874 and balanced accuracy of 0.4275, EvoAgentX obtains 0.2626 and 0.4138, and OpenClaw adapter obtains 0.2485 and 0.4032. Unlike in the 120-case stress-enriched validation, no agentic system exceeds the simple ML/logistic baseline on Macro-F1 in the natural-distribution setting.

The 507-case results also clarify the kind of behavior the agents learn under natural class balance. Most agentic systems remain reasonably sensitive to watch cases, but stress recall is still low. Nanobot and Hermes AI achieve the highest calibrated agent stress recall, at 0.0952, while LangGraph, EvoAgentX, OpenClaw, and NanoClaw each reach 0.0476. Thus, full-pool results confirms that StableEval Arena can collect valid JSON across heterogeneous systems on the 507-case pool, but rare-stress and sustained-depeg detection remain the main remaining limitation.

4.3 Cross Alignment and Stress Diagnostics

The 120-vs-507 alignment audits compare each StableEval-120-Enriched result with the corresponding StableEval-507-Natural run restricted to the same 120 case identifiers. The restricted 507 results are generally close to or better than the 120 results, indicating that the scaling runs do not contradict the original stress-enriched validation. The lower or different full-pool metrics in StableEval-507-Natural mainly reflect the stable-dominant natural distribution rather than a protocol inconsistency.

The contrast between the two benchmark blocks is substantive rather than contradictory. StableEval-120-Enriched shows that calibrated agents can improve aggregate classification when watch and stress cases are concentrated. StableEval-507-Natural shows that under the full stable-dominant historical distribution, simple baselines remain strong and agentic systems do not yet dominate classical methods. Together, the two blocks provide a complete evaluation: one emphasizes stress sensitivity, while the other tests full-pool robustness and execution behavior.

The stress-case audit gives the main risk signal. There are 21 true stress cases in the 507-case pool. The four native-level agents and the two adapter-based -Claw variants miss 19 out of 21. Only 2 stress cases are caught by at least one system. No agentic system reduces missed-depeg below 1.0000; historical persistence is the

Table 2: Main results for Two Completed StableEval Benchmark Blocks

Group	System	Execution	Cases	Acc. (↑)	Macro-F1 (↑)	Bal. Acc. (↑)	Stress R (↑)	MDR (↓)	Valid (↑)	Cost/case (↓)
StableEval-120-Enriched: 120-case stress-enriched validation, 54 stable / 45 watch / 21 stress										
Baseline	Always stable	deterministic	120	**0.450**	0.207	0.333	0.000	1.000	1.00	$0
Baseline	Historical persistence	deterministic	120	0.292	0.285	0.300	**0.333**	**0.667**	1.00	$0
Baseline	Threshold rule	deterministic	120	0.283	0.278	0.295	**0.333**	1.000	1.00	$0
Baseline	Simple ML/logistic	ML baseline	120	**0.450**	0.207	0.333	0.000	1.000	1.00	$0
Native	Nanobot	native/framework	120	0.442	**0.351**	**0.393**	0.095	1.000	1.00	**$0.0029**
Native	LangGraph	native/framework	120	0.442	0.337	0.381	0.048	1.000	1.00	$0.0101
Native	Hermes AI	native/framework	120	0.425	0.321	0.369	0.048	1.000	1.00	**$0.0029**
Native	EvoAgentX	native/framework	120	0.392	0.272	0.348	0.048	1.000	1.00	$0.0051
Adapter	OpenClaw	StableEval adapter	120	0.408	0.318	0.368	0.095	1.000	1.00	$0.0052
Adapter	NanoClaw	StableEval adapter	120	0.442	0.328	0.385	0.048	1.000	1.00	$0.0052
StableEval-507-Natural: 507-case natural-distribution scaling, 441 stable / 45 watch / 21 stress										
Baseline	Always stable	deterministic	507	**0.870**	0.310	0.333	0.000	1.000	1.00	$0
Baseline	Historical persistence	deterministic	507	0.673	0.387	0.450	0.333	**0.667**	1.00	$0
Baseline	Threshold rule	deterministic	507	0.645	0.376	0.446	0.333	1.000	1.00	$0
Baseline	Simple ML/logistic	ML baseline	507	0.641	**0.402**	**0.493**	**0.381**	1.000	1.00	$0
Native	Nanobot	native/framework	507	0.394	0.287	0.428	0.095	1.000	1.00	**$0.0029**
Native	LangGraph	native/framework	507	0.467	0.304	0.440	0.048	1.000	1.00	$0.0098
Native	Hermes AI	native/framework	507	0.408	0.301	0.433	0.095	1.000	1.00	**$0.0029**
Native	EvoAgentX	native/framework	507	0.381	0.263	0.414	0.048	1.000	1.00	$0.0050
Adapter	OpenClaw	StableEval adapter	507	0.353	0.248	0.403	0.048	1.000	1.00	$0.0050
Adapter	NanoClaw	StableEval adapter	507	0.471	0.304	0.435	0.048	1.000	1.00	$0.0050

Notes. Agent rows report calibrated outputs. "Valid" is the structured-output rate; for LLM-backed systems this is valid JSON rate. Stress R is recall on the enhanced stress class ($n = 21$ per block; one case = 0.0476). MDR is missed strict-depeg rate for sustained ≥100 bps depegs over 3h ($n = 3$ per block; one case = 0.333), computed from binary depeg probability and not equal to 1 − Stress R. Cost/case reports estimated model/API token cost only; deterministic and ML baselines have zero measured API cost. Accuracy is contextual; Macro-F1, balanced accuracy, Stress R, and MDR are the primary trustworthiness metrics. Full raw/calibrated rows and diagnostic artifacts are reported in the appendix.

only reported row with a lower missed-depeg rate, at 0.6667. Hence, adding more agentic systems improves benchmark coverage and execution completeness, but does not materially resolve rare-stress detection. Thus, weak rare-stress detection is consistent across both evaluation blocks and remains the central limitation exposed by StableEval Arena.

4.4 Calibration, Ablation, and Cost-Reliability Tradeoffs

Calibration improves the 120-case first-pass results by converting continuous risk estimates and maximum-deviation predictions into more useful stable/watch/stress labels. The best agent Macro-F1 rises from 0.301 in the raw setting to 0.351 after calibration, and the best agent stress recall rises from 0.000 to 0.095. This gain is modest but shows that part of the performance gap comes from label mapping rather than only from the underlying evidence packet.

The verifier layer produces a cost-sensitive tradeoff. It raises the best stress recall to 0.143, achieved by Nanobot and Hermes AI, but the best verifier-calibrated Macro-F1 is 0.349, slightly below the best calibrated first-pass Macro-F1 of 0.351. It also increases inference cost. For Nanobot, total cost rises from $0.345 under calibrated first-pass inference to $0.978 under verifier-calibrated inference. Thus, verifier-style auditing improves stress sensitivity, but the gain is limited and comes with a clear cost penalty.

The anonymization ablation probes whether results depend strongly on real stablecoin names or dates. The maximum label-change rate is 0.333, and mean probability changes range from 0.017 to 0.062. This indicates some identifier sensitivity, but not enough to suggest that the benchmark is driven primarily by memorized names or calendar effects, which we therefore treat as a reproducibility caveat rather than a protocol failure.

Executional reliability is consistently high. All main 120-case agent runs achieve valid JSON rate of 1.0 and failure rate of 0.0, and completed 507-case runs show that structured-output collection remains reliable at full-pool scale. This reliability result distincts from predictive performance: systems can follow the protocol and produce valid outputs, though rare-stress detection remains weak.

4.5 Summary

The results show four main patterns. First, in the 120-case stress-enriched validation, calibrated agents improve aggregate Macro-F1 over simple baselines. Second, in the 507-case natural-distribution, the benchmark runs successfully across all cases, but simple ML/logistic remains stronger than all agentic systems on Macro-F1 and balanced accuracy. Third, 507 results show why accuracy alone is unreliable under stable-dominant class balance. Fourth, rare-stress and sustained-depeg detection remain the main limitation: agents usually follow the output schema, but still miss most stress cases.

5 Discussion and Analysis

The main result of StableEval Arena is a separation between procedural reliability and financial-risk reliability. The evaluated systems can execute a shared historical-replay protocol and return valid structured predictions across the full pool. However, this reliability does not translate into dependable detection of rare peg stress or sustained depeg. StableEval Arena should therefore be read not as evidence that current agentic systems solve stablecoin stress prediction, but as evidence that shared agentic evaluation can expose a gap between protocol-following reliability and rare-event financial trustworthiness.

5.1 Two-Block Evaluation

The two benchmark blocks shown in Table 2 answer complementary questions. StableEval-120-Enriched asks whether agents can reason about windows where watch and stress behavior is frequent enough to measure. By concentrating non-stable cases, it prevents stress behavior from being washed out by the stablecoin base rate. In this setting, calibrated agents improve Macro-F1 relative to the strongest simple baseline, indicating that structured evidence

packets and calibration can support useful stable/watch/stress judgments. However, this result should be interpreted narrowly: low stress recall shows that better aggregate classification is not the same as reliable severe-risk detection.

StableEval-507-Natural asks whether the same protocol remains informative over the full pool under natural class balance. Because 441 of 507 cases are stable, accuracy becomes a weak indicator of risk sensitivity. The always-stable baseline makes this explicit: it achieves high accuracy while recovering no stress cases. The full-pool result also changes the comparison with baselines. Simple ML/logistic remains strongest by Macro-F1 and balanced accuracy, so agentic execution alone does not dominate classical methods once the natural stable majority is restored.

Together, the two-block design avoids two symmetric overreadings. The enriched block prevents the benchmark from concluding that systems are reliable simply because most cases are stable. The natural block prevents stress-enriched gains from being mistaken for deployment-like robustness. Thus, StableEval Arena should be interpreted less as a single leaderboard and more as an evaluation framework: shared prompts, backend, schema, calibration, metrics, and execution logs support reproducible diagnostic comparison across stress-enriched and natural-distribution settings [8, 25].

5.2 Reliability vs. Financial Trustworthiness

Operationally, the systems are reliable; financially, stress detection remains weak. The six evaluated systems complete the protocol with valid outputs across 507 cases, and the completed 507-case runs show that heterogeneous agents can be compared through one historical-replay interface. This matters because malformed outputs, missing predictions, runtime failures, and unlogged costs can otherwise make agent comparisons difficult to reproduce.

The negative finding is financial. Valid, low-cost JSON outputs are not equivalent to trustworthy monitoring. In the 507-case stress audit, all six systems miss 19 of 21 true stress cases. On the sustained-depeg audit, no agentic system in Table 2 reduces missed-depeg rate below 1.0000. A system that appears reliable by execution metrics can remain unreliable for event type that matters most. In financial monitoring, missed stress and false alarms often have asymmetric costs, so evaluation should not treat all classification errors as equally important. For this reason, evaluation must report stress recall, missed-depeg rate, false alarms, calibration, and deviation error alongside accuracy, latency, cost, and valid-output rate.

This distinction aligns with agent-evaluation work emphasizing runtime behavior, diagnostics, and failure modes beyond final-answer accuracy [3]. For StableEval Arena, the primary failure mode is not inability to follow the output contract. It is the ability to follow the contract while still missing rare financial risk.

5.3 Why Rare Stress Remains Difficult

Rare-stress detection is difficult partly because the input evidence is intentionally bounded. The 30-day hourly lookback makes the benchmark reproducible and leakage-safe, but compact price and market evidence cannot fully represent reserves, redemption frictions, exchange depth, collateral composition, governance behavior, oracle conditions, cross-chain liquidity, or market confidence. This limitation is consistent with stablecoin research showing that peg stability depends on market design, governance, liquidity, and stakeholder context, not price history alone [2, 16, 21, 26].

The results also suggest conservative decision behavior under ambiguity. Since most windows remain near peg, agents often choose stable or watch unless the packet contains unusually strong stress evidence. That tendency can help aggregate performance in enriched 120-case block, but it is poorly aligned with high-consequence monitoring, where missing a rare stress event may be more costly than issuing some false alarms. Calibration and verifier layers can change the threshold between risk scores and labels, but they cannot recover signals that the packet does not contain.

This makes the benchmark's contribution precise. It does not show that current agents solve stablecoin risk prediction. It shows where their reliability stops: they can process bounded evidence and produce structured, reproducible assessments, but rare-stress detection still requires richer signals, more risk-sensitive decision rules, or calibration objectives that explicitly penalize missed depegs. This also points to future extensions using multimodal market evidence, stablecoin-arrangement infrastructure evidence, and oracle-network/interoperability evidence [5, 6, 11].

6 Industry Implications and Future Work

StableEval Arena connects agentic AI evaluation to a broader industry question: whether agentic systems can become reliable monitoring layers for stablecoin-based payment infrastructure. StableEval-120-Enriched and StableEval-507-Natural show that agents can produce structured outputs reliably, yet weak rare-stress detection remains a barrier to automated payment execution. Thus, procedural reliability does not imply reliable financial-risk detection: an agent may satisfy interface requirements while still missing rare events that affect payment safety. This finding aligns with DeFi and stablecoin research that frames blockchain-based finance as programmable infrastructure while emphasizing smart-contract, governance, oracle, regulatory, interoperability, and stakeholder-specific risks [15, 26]. We therefore highlight three industry implications as motivation for future extensions of StableEval.

The first industry implication concerns delegated authorization and interoperability for agentic and recurring payments. Web2 subscription systems rely on mature card, banking, and platform rails with established authorization, dispute, and refund mechanisms. By contrast, stablecoin-based recurring payments require users to delegate payment authority to applications, smart contracts, or agentic services [18]. Recent academic and policy work instead frames stablecoin payments as an unsettled design space involving authorization, interoperability, recourse, settlement, and risk allocation [4, 18, 24]. However, these efforts remain competing emerging directions rather than settled standards or broadly adopted market practice. Their significance is that they make user consent, authorization scope, spending limits, revocation, auditability, and liability concrete engineering requirements for agentic payment systems. StableEval Arena 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 with higher stress recall than observed here.

The second industry implication concerns SLA design, clearing, and economic settlement finality. For stablecoins to support large-scale application payments, systems must specify when a payment is final, how reconciliation is performed, how failed or delayed settlement is escalated, and who bears the resulting loss. Regulatory

discussions increasingly frame stablecoin arrangements as payment and financial-market infrastructure, applying concepts such as governance, risk management, settlement finality, money settlement, issuer reserves, and supervision [4, 5, 9, 10, 23]. These documents describe a regulatory target state, but operational settlement practice is not yet mature in deployed systems. In practice, on-chain confirmation alone does not establish economic finality: a stablecoin payment becomes economically final only when the recipient can use the funds, liquidity is available, compliance checks have cleared, reconciliation is complete, and responsibility for failure or delay is assigned [4, 5]. This gap is directly connected to StableEval Arena: our results show that an agent can produce valid JSON with zero runtime failures while still missing rare stress events, which in a payment SLA could lead to delayed settlement, failed reconciliation, or unsafe automatic execution. Future benchmarks should therefore evaluate not only next-window price classification, but also detection latency, reconciliation mismatch risk, settlement-delay risk, escalation behavior, and the cost of missed stress under explicit SLA rules.

The third industry implication concerns settlement pricing and tax accountability in automated stablecoin payments. Settlement pricing also creates accountability problems because stablecoin prices can diverge across venues, oracle feeds, and liquidity pools during market stress. Prior work on stablecoin depegging, market spillovers, and oracle risk shows that venue-specific prices and external data feeds can affect downstream liquidation, settlement, and payment outcomes [7, 17, 22]. When prices diverge across venues, the system must decide whether the user, application, platform, exchange, or another intermediary absorbs the spread. Tax handling creates a parallel accountability problem: systems must determine the applicable jurisdiction, calculate fees and value-added taxes, and assign responsibility for remittance. These pricing and tax questions reflect a broader payment-design gap, since stablecoin arrangements differ from card networks in transaction lifecycle, recourse mechanisms, and risk allocation [18]. Hence, StableEval Arena can be extended with business-cost labels for false interruptions, unfavorable settlement, spread losses, underpayment, tax misclassification, and compliance disputes.

These implications suggest several future directions. First, future versions should broaden the asset universe and incorporate richer on-chain, off-chain, and unstructured evidence, including liquidity, redemption, exchange depth, oracle feeds, reserve attestations, market news, wallet-level flows, authorization mandates, revocation logs, audit trails, and settlement records. Second, evaluation should move from prediction quality alone to stakeholder-aware and risk-sensitive cost functions, including missed-stress penalties, detection-latency costs, user loss, developer revenue risk, platform compliance burden, settlement-price error, spread allocation, tax-calculation risk, and jurisdictional compliance risk [1, 18, 26]. Third, future work should test multiple LLM backends, retain simple ML baselines as mandatory controls, add anonymized and counterfactual identifier checks, and evaluate multi-agent payment orchestration under explicit authorization, SLA, pricing, 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

We identify several limitations for StableEval Arena. First, severe stablecoin stress cases are rare. StableEval-507-Natural contains only 21 stress cases, while StableEval-120-Enriched intentionally concentrates them for diagnostic evaluation. Accordingly, stress recall and missed-depeg results should be read as risk warning signals rather than final statistical conclusions. This limitation is structural: stablecoins usually remain near peg, while consequential failures occur in short, unusual windows. The asset scope is also limited to USDT, USDC, and DAI, so future releases should test broader stablecoin designs and related payment assets.

Second, the benchmark prioritizes bounded, reproducible evidence packets over exhaustive real-world surveillance. The current packets include price, volume, market-structure, and contextual signals available in the frozen lookback window, but they do not include every high-cost or real-time risk signal that might matter in practice. Future extensions will add reserve attestations, redemption flows, cross-chain liquidity, DeFi liquidation activity, oracle feeds, regulatory events, and news while preserving leakage-safe historical replay. Additionally, using `qwen/qwen-2.5-72b-instruct` via OpenRouter improves comparability across agentic systems, but it does not test backend generality. Similarly, OpenClaw and NanoClaw are reported as StableEval adapter-based compatibility extensions rather than native CMDOP executions.

Finally, residual parametric-memory effects cannot be fully ruled out because some packets retain real stablecoin names and dates, even though future labels are hidden and anonymization checks are included. Future releases will broaden anonymized and counterfactual evaluation and strengthen dataset cards and artifact manifests, following benchmark and datasheet practices [8, 12]. These boundaries define the current benchmark setting and paths for extending trustworthy agentic AI evaluation in stablecoin risk monitoring.

8 Conclusion

We introduce StableEval Arena, a cost-aware agentic benchmark for stablecoin price-stability risk assessment. Each case provides a 30-day evidence window and requires structured next-7-day predictions of stable/watch/stress status, depeg probability, maximum peg deviation, direction, confidence, and rationale. The benchmark compares agentic systems with simple baselines under a unified protocol and evaluates prediction quality, stress recall, missed-depeg risk, calibration, valid-output rate, latency, token usage, and inference cost. The completed experiments combine a 120-case stress-enriched validation for rare-risk diagnosis with a 507-case natural-distribution scaling study for full-pool robustness and operational evaluation. Rather than showing current agents can act as deployment-ready stablecoin risk monitors, our results show that a leakage-safe, cost-aware agentic benchmark can reveal the gap between valid execution and trustworthy rare-event financial detection. Across both settings, systems produce valid structured outputs, but rare-stress and sustained-depeg detection remain weak. This gap between execution reliability and financial-risk detection is the central finding of StableEval Arena. Future work should broaden evidence packets and asset coverage, test backend sensitivity and more robust native agent execution, and evaluate richer payment, settlement, and stablecoin-risk monitoring scenarios.

References

[1] Iñaki Aldasoro, Jon Frost, and Hiro Ito. 2026. *The Impact of Stablecoins on the International Monetary and Financial System.* BIS Papers 170. Bank for International Settlements. https://www.bis.org/publ/bppdf/bispap170.htm

[2] Douglas W. Arner, Raphael Auer, and Jon Frost. 2020. *Stablecoins: Risks, Potential and Regulation.* BIS Working Paper 905. Bank for International Settlements. https://www.bis.org/publ/work905.htm

[3] 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] doi:10.48550/arXiv.2604.10547

[4] Committee on Payments and Market Infrastructures. 2023. *Considerations for the Use of Stablecoin Arrangements in Cross-border Payments.* CPMI Papers 220. Bank for International Settlements. https://www.bis.org/cpmi/publ/d220.htm

[5] Committee on Payments and Market Infrastructures and Board of the International Organization of Securities Commissions. 2022. *Application of the Principles for Financial Market Infrastructures to Stablecoin Arrangements.* CPMI-IOSCO final guidance 206. Bank for International Settlements. https://www.bis.org/cpmi/publ/d206.htm Accessed 2026-06-10.

[6] 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

[7] Chanelle Duley, Leonardo Gambacorta, Rodney Garratt, and Priscilla Koo Wilkens. 2023. *The Oracle Problem and the Future of DeFi.* BIS Bulletin 76. Bank for International Settlements. https://www.bis.org/publ/bisbull76.htm

[8] 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. In *NeurIPS 2025 Datasets and Benchmarks Track.* arXiv:2506.16791 [cs.LG] https://arxiv.org/abs/2506.16791 Spotlight.

[9] European Parliament and Council of the European Union. 2023. Regulation (EU) 2023/1114 of the European Parliament and of the Council of 31 May 2023 on Markets in Crypto-assets. https://eur-lex.europa.eu/eli/reg/2023/1114/oj Official Journal of the European Union, L 150, pp. 40–205.

[10] 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/

[11] 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. arXiv:2405.00522 [econ.GN] doi:10.1109/MetaCom62920.2024.00025

[12] 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

[13] Gary B. Gorton and Jeffery Y. Zhang. 2023. Taming Wildcat Stablecoins. *University of Chicago Law Review* 90, 3 (2023), 909–971. https://lawreview.uchicago.edu/print-archive/taming-wildcat-stablecoins

[14] 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, 1321–1330. https://proceedings.mlr.press/v70/guo17a.html

[15] 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

[16] 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

[17] Yi-Hsi Lee, Yu-Fen Chiu, and Ming-Hua Hsieh. 2025. Stablecoin Depegging Risk Prediction. *Pacific-Basin Finance Journal* 90 (2025), 102640. doi:10.1016/j.pacfin.2024.102640

[18] Yuquan Li, Yuexin Xiang, Qin Wang, Tsz Hon Yuen, Andreas Deppeler, and Jiangshan Yu. 2026. SoK: Stablecoins in Retail Payments. arXiv:2601.00196 [q-fin.GN] doi:10.48550/arXiv.2601.00196

[19] 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 V.2* (Toronto, ON, Canada) *(KDD '25).* Association for Computing Machinery, New York, NY, USA, 5595–5606. doi:10.1145/3711896.3737442

[20] 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. arXiv:2308.03688 [cs.AI] https://openreview.net/forum?id=zAdUB0aCTQ

[21] 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

[22] Baptiste Perez Riaza and Jean-Yves Gnabo. 2025. From Depegs to Jumps: The Role of Stablecoin Instabilities in Crypto Market Dynamics. *Journal of International Money and Finance* 155 (2025), 103339. doi:10.1016/j.jimonfin.2025.103339

[23] United States Congress. 2025. Guiding and Establishing National Innovation for U.S. Stablecoins Act. Public Law 119-27. https://www.govinfo.gov/app/details/PLAW-119publ27 Approved July 18, 2025.

[24] Gang Wang, Qin Wang, and Shiping Chen. 2023. Exploring Blockchains Interoperability: A Systematic Survey. *Comput. Surveys* 55, 13s, Article 290 (2023), 38 pages. doi:10.1145/3582882

[25] 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

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

A Shared Protocol, Data, and Execution Metadata

This appendix provides the audit trail behind two implementation-result blocks. StableEval-120-Enriched is the frozen 120-case stress-enriched validation block used for the main six-system comparison. StableEval-507-Natural is the full 507-case natural-distribution scaling block. Both blocks use 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 logic. The 507-case block is additive: it tests scaling and natural-distribution behavior, while the 120-case block remains the primary rare-risk validation because it is stress-enriched.

A.1 Glossary of Notation and Key Terms

Table 3: Glossary of key terms, abbreviations, and mathematical notation used in the appendix.

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 Case Construction

This section defines the shared benchmark data used by both result blocks. Table 4 summarizes the conversion from hourly stablecoin market observations into historical-replay cases. The same source universe, 30-day lookback, 7-day hidden horizon, and stable/watch/stress label definitions are used for both blocks. Hidden future-horizon labels are used only for case selection and evaluation; they are never included in the agent-facing prompt packet.

Table 4: Dataset and case-construction summary for the shared StableEval benchmark protocol.

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
Stride	7 days
Full frozen case pool	507 historical-replay cases
Strict sustained-depeg cases	3 in the full frozen pool
Implementation result blocks	StableEval-120-Enriched; StableEval-507-Natural

Table 5: Stable/watch/stress and depeg label definitions used by the evaluator.

Label or quantity	Definition
Deviation bps	$\lvert\text{hourly close} - 1.0\rvert \times 10000$
Stable	No meaningful next-7-day instability under the enhanced label rule.
Watch	Mild but meaningful next-7-day instability, including any hourly close deviation at or above 50 bps absent strict sustained depeg.
Stress	Large, persistent, or operationally meaningful instability, including strict sustained depeg, 100 bps close deviation, repeated 50 bps hours, or high average absolute deviation.
Strict sustained depeg	Absolute hourly close deviation at or above 100 bps for at least three consecutive hours.

A.3 Agentic Systems and Execution Metadata

StableEval Arena compares six platforms under a common model backend and shared prompt/evaluator protocol. Nanobot, LangGraph, Hermes AI, and EvoAgentX are native or framework-level agent comparisons. OpenClaw and NanoClaw are included as StableEval adapter executions. Because all LLM-backed systems use the same backend, the comparison measures platform-mediated execution, structured-output behavior, cost, latency, and reliability under a shared interface rather than independent foundation-model ability.

Table 6: 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 Computing Environment and Resource Configuration

StableEval primarily uses API-based LLM inference through OpenRouter. Local hardware affects orchestration, prompt-packet construction, logging, JSON parsing, calibration, scoring, analysis, and figure/table generation, while the LLM inference itself is served remotely.

Table 7: Execution-environment facts that are recorded or directly supported by the release artifacts.

Field	Recorded value or reproducibility note
Execution setting	API-backed LLM inference plus local orchestration and evaluation.
LLM backend	`qwen/qwen-2.5-72b-instruct` through OpenRouter.
Decoding temperature	0.
Web browsing	Disabled in the reported protocol.
Local GPU/TPU use	No local GPU/TPU is required for saved-output recomputation; LLM inference is remote.
Local computation responsibilities	Build prompt packets, call platform wrappers, log outputs, parse strict JSON, apply calibration, compute metrics, and generate tables/figures.
Package/runtime metadata	`results/stableeval_120_enriched/agent_execution_metadata.csv`

Backend metadata	`results/stableeval_120_enriched/model_backend.json`
	`results/stableeval_507_natural/metadata/model_backend.json`
	`protocols/stableeval_507_natural_protocol.json`

B StableEval-120-Enriched Validation Results

The 120-case stress-enriched validation block 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. This composition makes the 120-case block the primary comparison for warning quality and rare-event trustworthiness.

B.1 120-Case Leaderboard-Style Summary Figures

The following figures summarize the main 120-case validation patterns before the numeric tables. They are drawn from the reported validation metrics and use the StableEval color palette used in the main paper.

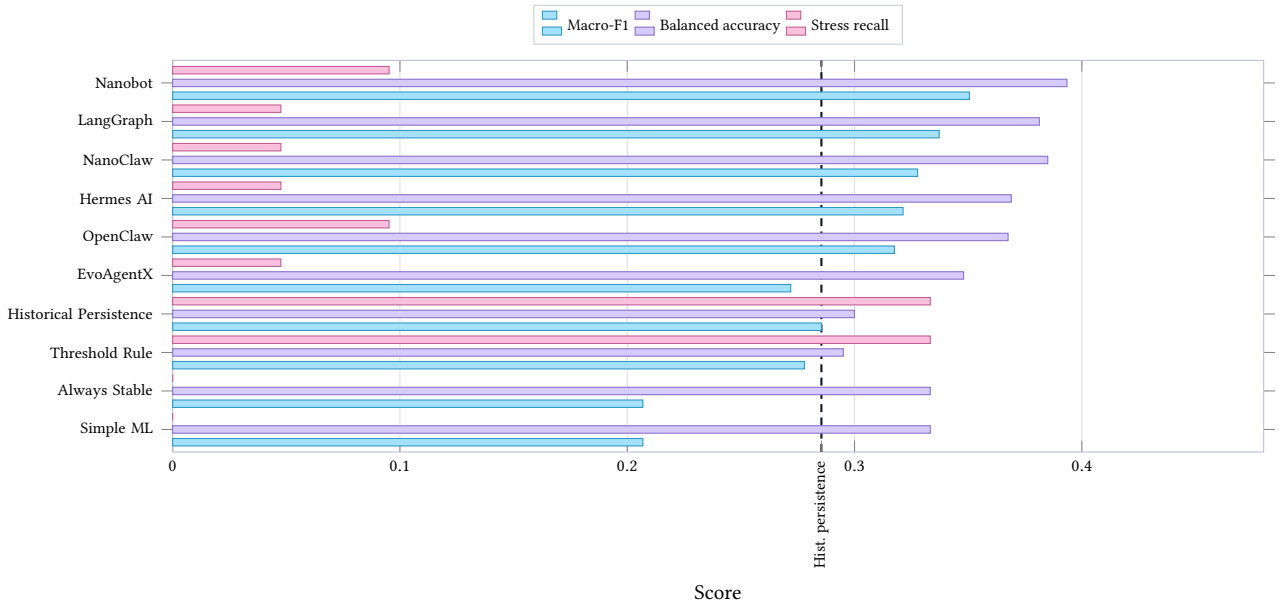

Figure 3: Validation leaderboard on the 120-case stress-enriched benchmark. Bars report Macro-F1, balanced accuracy, and stress-class recall from the calibrated first-pass protocol. The dashed reference line marks the Macro-F1 of the strongest non-agent baseline, historical persistence.

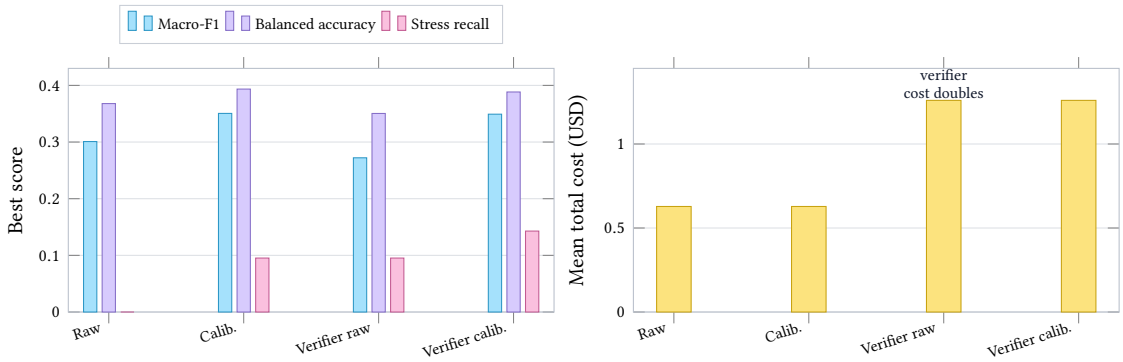

Figure 4: 120-case ablation summary across raw first-pass, calibrated first-pass, and verifier protocols. Calibration improves the best aggregate scores. The verifier increases best stress recall but approximately doubles mean total cost.

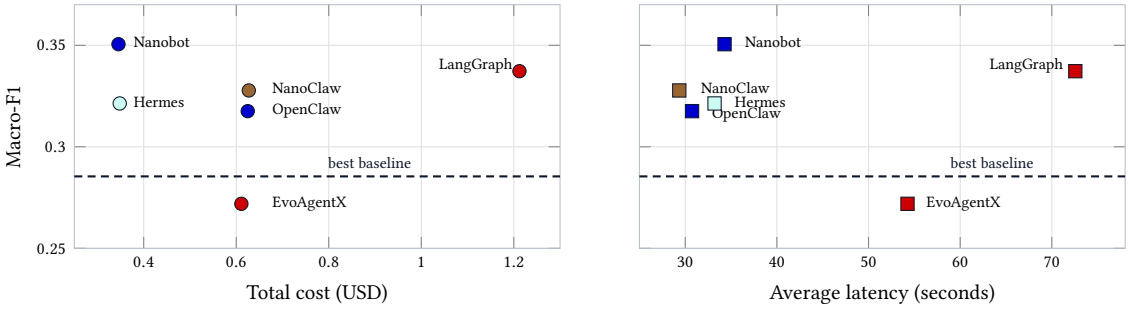

Figure 5: Cost-aware 120-case performance tradeoffs for the six agentic systems. Nanobot has the strongest Macro-F1 at low total cost, while LangGraph is competitive but more expensive and slower under the frozen validation protocol.

B.2 120-Case Leaderboard Tables

Tables 8 and 9 provide the compact calibrated 120-case result. The wider cross-variant matrices are retained at the end of this section because they are the most direct way to audit raw, calibrated, baseline-only, and verifier variants from the saved outputs.

Table 8: 120-case calibrated 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 9: 120-case calibrated validation summary: continuous error, reliability, latency, token, and cost metrics.

System	MAE	Valid JSON	Failure	Avg. latency	Avg. tokens	Cost/case	Total cost
Nanobot	56.617	1.000	0.000	34.287	7968.908	0.003	0.345
LangGraph	58.683	1.000	0.000	72.562	27529.942	0.010	1.212
NanoClaw	66.358	1.000	0.000	29.351	14029.575	0.005	0.627
Hermes AI	57.858	1.000	0.000	33.198	8031.433	0.003	0.348
OpenClaw	62.000	1.000	0.000	30.742	13989.475	0.005	0.625
EvoAgentX	66.917	1.000	0.000	54.254	14096.733	0.005	0.611
Historical persistence	148.858	1.000	0.000	0.000	0.000	0.000	0.000
Threshold rule	148.890	1.000	0.000	0.000	0.000	0.000	0.000
Always stable	63.908	1.000	0.000	0.000	0.000	0.000	0.000
Simple ML logistic	113.500	1.000	0.000	0.000	0.000	0.000	0.000

B.3 120-Case Stress and Ablation Diagnostics

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 10: Frozen calibrated-label and verifier rules used for the 120-case ablation and reused for post-hoc calibration in the 507-case block.

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

Table 11: 120-case stress-class recall plus strict sustained-depeg TP/FN audit. Binary TP/FN refer only to the three strict sustained-depeg cases, while stress recall is the three-class stress-label recall. Stress R is recall on the three-class stress label. MDR and Binary TP/FN are computed only over the three strict sustained-depeg cases; FAR is computed over non-strict-depeg cases.

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

Table 12: 120-case validation ablation summary. Calibration improves the best aggregate metrics without additional calls; the verifier improves best stress recall but raises cost.

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

Table 13: 120-case 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: 120-case full leaderboard across validation variants: risk, continuous-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

C StableEval-507-Natural Scaling Results

The 507-case natural-distribution block contains 441 stable cases, 45 watch cases, and 21 stress cases. It uses the same stablecoin universe, 30-day lookback window, hidden 7-day horizon, risk-sensitive prompt contract, OpenRouter Qwen backend, temperature-zero decoding, strict JSON parsing, and frozen post-hoc calibration rule used in the 120-case block. The stable-heavy composition makes this block an operational scaling and robustness check rather than a replacement for the stress-enriched 120-case validation.

C.1 507-Case Leaderboard-Style Figures

The following figures mirror the 120-case summaries: calibrated leaderboard, raw-versus-calibrated ablation, and cost-aware performance tradeoffs. Because the full 507-case pool is stable-heavy, non-agent baselines are comparatively strong on aggregate Macro-F1; this is an expected distribution effect.

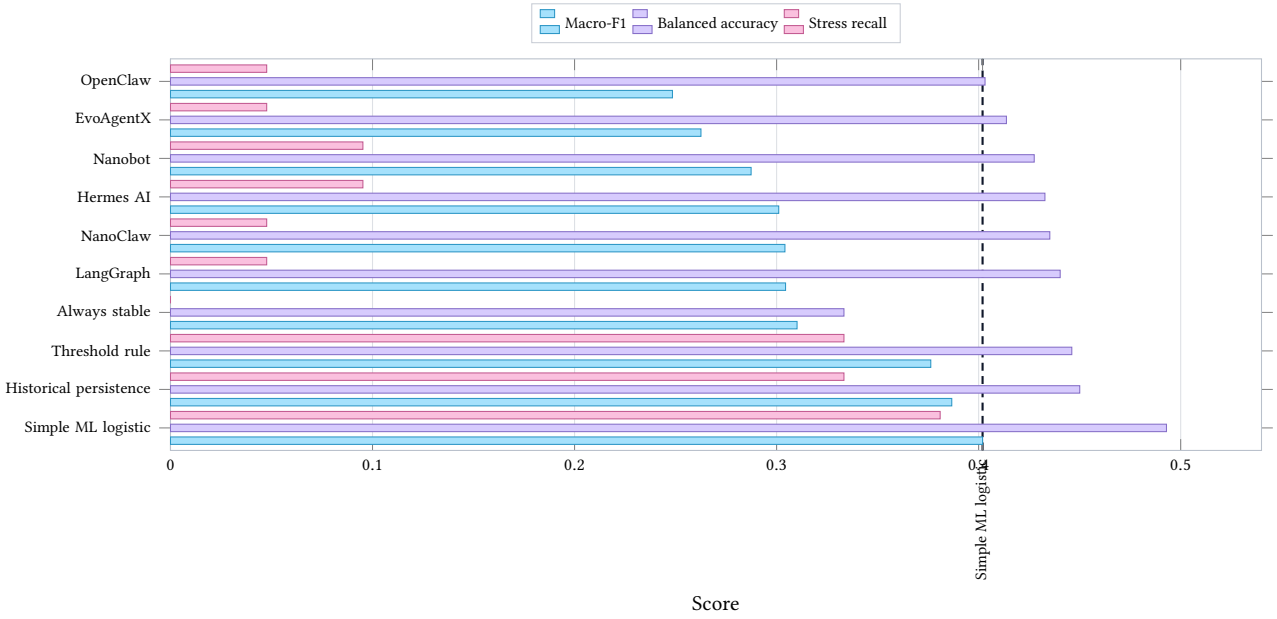

Figure 6: Full 507-case natural-distribution leaderboard. Bars report Macro-F1, balanced accuracy, and stress-class recall from the calibrated first-pass protocol. The dashed reference line marks the Macro-F1 of the strongest non-agent baseline, Simple ML logistic.

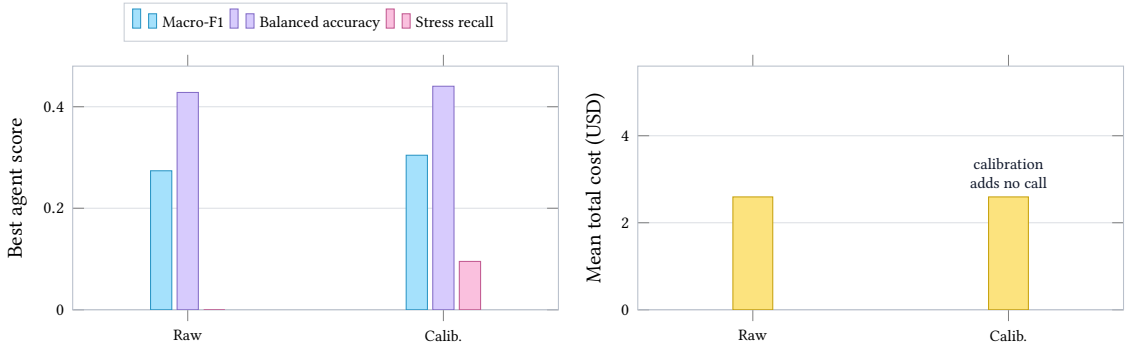

Figure 7: 507-case raw-versus-calibrated ablation. Calibration improves the best agent Macro-F1, balanced accuracy, and stress recall without increasing total cost because it remaps already returned risk estimates and predicted deviations.

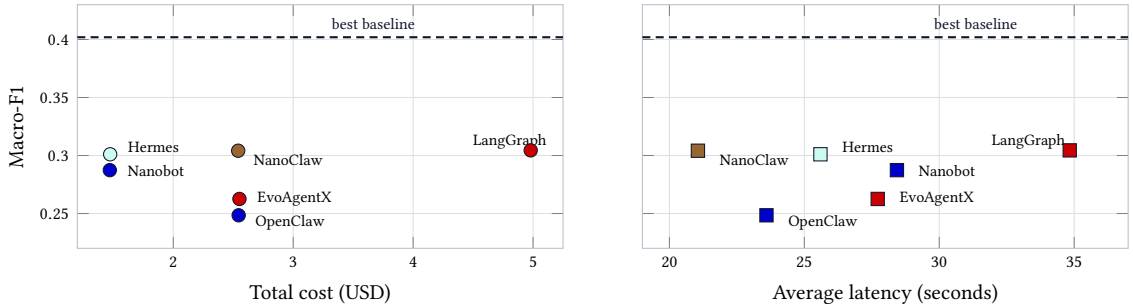

Figure 8: Cost-aware performance tradeoffs for the six agentic systems on the full 507-case natural-distribution scaling check. The agents remain operationally reliable, but the stable-heavy distribution makes the strongest non-agent baseline difficult to beat on aggregate Macro-F1.

C.2 507-Case Leaderboard Tables

Tables 15 and 16 report the calibrated 507-case summary. The raw/calibrated matrices remain in this section so reviewers can reconstruct the complete six-agent and four-baseline scaling result.

Table 15: 507-case natural-distribution scaling summary: predictive and risk metrics.

System	Type	Acc.	Macro-F1	BalAcc	Stress R	MDR	FAR
LangGraph	agent	0.467	0.304	0.440	0.048	1.000	0.004
NanoClaw	agent	0.471	0.304	0.435	0.048	1.000	0.004
Hermes AI	agent	0.408	0.301	0.433	0.095	1.000	0.004
Nanobot	agent	0.394	0.287	0.428	0.095	1.000	0.002
EvoAgentX	agent	0.381	0.263	0.414	0.048	1.000	0.004
OpenClaw	agent	0.353	0.248	0.403	0.048	1.000	0.004
Simple ML logistic	baseline	0.641	0.402	0.493	0.381	1.000	0.077
Historical persistence	baseline	0.673	0.387	0.450	0.333	0.667	0.149
Threshold rule	baseline	0.645	0.376	0.446	0.333	1.000	0.095
Always stable	baseline	0.870	0.310	0.333	0.000	1.000	0.000

Table 16: 507-case natural-distribution scaling summary: continuous error, reliability, latency, token, and cost metrics.

System	MAE	Valid JSON	Failure	Avg. latency	Avg. tokens	Cost/case	Total cost
LangGraph	31.063	1.000	0.000	34.835	27210.148	0.010	4.980
NanoClaw	31.672	1.000	0.000	21.057	13879.876	0.005	2.541
Hermes AI	34.375	1.000	0.000	25.597	8046.953	0.003	1.474
Nanobot	30.242	1.000	0.000	28.431	8031.456	0.003	1.471
EvoAgentX	32.680	1.000	0.000	27.720	13931.836	0.005	2.552
OpenClaw	34.655	1.000	0.000	23.597	13900.018	0.005	2.545
Simple ML logistic	17.067	1.000	0.000	0.000	0.000	0.000	0.000
Historical persistence	52.665	1.000	0.000	0.000	0.000	0.000	0.000
Threshold rule	52.688	1.000	0.000	0.000	0.000	0.000	0.000
Always stable	20.170	1.000	0.000	0.000	0.000	0.000	0.000

C.3 507-Case Stress and Calibration Diagnostics

The 507-case scaling run separates raw model labels from calibrated labels in the same way as the 120-case block. Calibration uses the frozen rule in Table 10 and does not alter model probabilities, predicted maximum deviations, costs, tokens, or latencies. On the full natural-distribution set, calibration improves every agent's Macro-F1 relative to its raw first-pass label, while non-agent baselines remain strong because the full set is stable-heavy.

Table 17: 507-case stress-class recall plus strict sustained-depeg TP/FN audit. Binary TP/FN refer only to the three strict sustained-depeg cases. Stress R is recall on the three-class stress label. MDR and Binary TP/FN are computed only over the three strict sustained-depeg cases; FAR is computed over non-strict-depeg cases.

System	Type	Stress R	MDR	FAR	Binary TP	Binary FN
Simple ML logistic	baseline	0.381	1.000	0.077	0	3
Historical persistence	baseline	0.333	0.667	0.149	1	2
Threshold rule	baseline	0.333	1.000	0.095	0	3
Hermes AI	agent	0.095	1.000	0.004	0	3
Nanobot	agent	0.095	1.000	0.002	0	3
NanoClaw	agent	0.048	1.000	0.004	0	3
LangGraph	agent	0.048	1.000	0.004	0	3
EvoAgentX	agent	0.048	1.000	0.004	0	3
OpenClaw	agent	0.048	1.000	0.004	0	3
Always stable	baseline	0.000	1.000	0.000	0	3

Table 18: 507-case raw-versus-calibrated ablation summary for agentic systems.

Variant	Best agent Macro-F1	Best agent BalAcc	Best agent Stress R	Mean cost
full-507-calibrated	0.304	0.440	0.095	2.594
full-507-raw	0.274	0.428	0.000	2.594

Table 19: Full 507-case leaderboard: classification metrics for baselines and raw/calibrated agent variants.

Variant	System	Type	Acc.	Macro-F1	BalAcc	R stable	R watch	R stress
full-507-calibrated	LangGraph	agent	0.467	0.304	0.440	0.451	0.822	0.048
full-507-calibrated	NanoClaw	agent	0.471	0.304	0.435	0.458	0.800	0.048
full-507-calibrated	Hermes AI	agent	0.408	0.301	0.433	0.381	0.822	0.095
full-507-calibrated	Nanobot	agent	0.394	0.287	0.428	0.365	0.822	0.095
full-507-calibrated	EvoAgentX	agent	0.381	0.263	0.414	0.349	0.844	0.048
full-507-calibrated	OpenClaw	agent	0.353	0.248	0.403	0.317	0.844	0.048
full-507-raw	LangGraph	agent	0.458	0.274	0.428	0.440	0.844	0.000
full-507-raw	NanoClaw	agent	0.410	0.249	0.397	0.390	0.800	0.000
full-507-raw	EvoAgentX	agent	0.369	0.232	0.407	0.333	0.889	0.000
full-507-raw	Nanobot	agent	0.355	0.223	0.389	0.322	0.844	0.000
full-507-raw	Hermes AI	agent	0.335	0.214	0.388	0.297	0.867	0.000
full-507-raw	OpenClaw	agent	0.327	0.210	0.385	0.288	0.867	0.000
full-507-baselines	Simple ML logistic	baseline	0.641	0.402	0.493	0.676	0.422	0.381
full-507-baselines	Historical persistence	baseline	0.673	0.387	0.450	0.728	0.289	0.333
full-507-baselines	Threshold rule	baseline	0.645	0.376	0.446	0.694	0.311	0.333
full-507-baselines	Always stable	baseline	0.870	0.310	0.333	1.000	0.000	0.000

Table 20: Full 507-case leaderboard: risk, continuous-error, and cost metrics for baselines and raw/calibrated agent variants.

Variant	System	MDR	FAR	Brier	MAE	RMSE	Cost	Cost/correct
full-507-calibrated	LangGraph	1.000	0.004	0.021	31.063	99.662	4.980	0.021
full-507-calibrated	NanoClaw	1.000	0.004	0.020	31.672	100.153	2.541	0.011
full-507-calibrated	Hermes AI	1.000	0.004	0.019	34.375	99.966	1.474	0.007
full-507-calibrated	Nanobot	1.000	0.002	0.023	30.242	84.236	1.471	0.007
full-507-calibrated	EvoAgentX	1.000	0.004	0.022	32.680	100.002	2.552	0.013
full-507-calibrated	OpenClaw	1.000	0.004	0.023	34.655	100.163	2.545	0.014
full-507-raw	LangGraph	1.000	0.004	0.021	31.063	99.662	4.980	0.021
full-507-raw	NanoClaw	1.000	0.004	0.020	31.672	100.153	2.541	0.012
full-507-raw	EvoAgentX	1.000	0.004	0.022	32.680	100.002	2.552	0.014
full-507-raw	Nanobot	1.000	0.002	0.023	30.242	84.236	1.471	0.008
full-507-raw	Hermes AI	1.000	0.004	0.019	34.375	99.966	1.474	0.009
full-507-raw	OpenClaw	1.000	0.004	0.023	34.655	100.163	2.545	0.015
full-507-baselines	Simple ML logistic	1.000	0.077	0.067	17.067	72.335	0.000	0.000
full-507-baselines	Historical persistence	0.667	0.149	0.112	52.665	154.288	0.000	0.000
full-507-baselines	Threshold rule	1.000	0.095	0.093	52.688	154.287	0.000	0.000

Variant	System	MDR	FAR	Brier	MAE	RMSE	Cost	Cost/correct
full-507-baselines	Always stable	1.000	0.000	0.006	20.170	72.666	0.000	0.000

D Cross-Block Robustness, Alignment, and Diagnostic Audits

This section collects diagnostics that cut across result blocks or audit benchmark safety rather than reporting one block's main leaderboard. The purpose is to help reviewers distinguish aggregate performance, rare-stress behavior, calibration effects, leakage checks, and adapter caveats.

D.1 120-Versus-507 Scaling Context

Table 21 compares the calibrated agentic systems across the stress-enriched and natural-distribution blocks. The 507 block has lower stress prevalence and many more stable controls, so changes in Macro-F1 and balanced accuracy should be interpreted as distribution-sensitive scaling behavior rather than a replacement for the 120-case validation.

Table 21: Compact 120-case versus 507-case calibrated scaling context for agentic systems.

System	120 Macro-F1	507 Macro-F1	120 BalAcc	507 BalAcc	120 Stress R	507 Stress R	120 Cost	507 Cost
Nanobot	0.351	0.287	0.393	0.428	0.095	0.095	0.345	1.471
LangGraph	0.337	0.304	0.381	0.440	0.048	0.048	1.212	4.980
NanoClaw	0.328	0.304	0.385	0.435	0.048	0.048	0.627	2.541
Hermes AI	0.321	0.301	0.369	0.433	0.048	0.095	0.348	1.474
OpenClaw	0.318	0.248	0.368	0.403	0.095	0.048	0.625	2.545
EvoAgentX	0.272	0.263	0.348	0.414	0.048	0.048	0.611	2.552

Across both blocks, the central trustworthiness caveat is unchanged: agents can improve some aggregate metrics, but strict sustained-depeg detection remains weak. The binary TP/FN columns in Tables 11 and 17 make this limitation explicit.

D.2 Anonymization and Leakage-Safety Audit

The anonymization ablation probes possible parametric-memory leakage by comparing 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 signal rather than a proof of no leakage. The matched 12-case design preserves the numerical prompt evidence while changing identifying metadata.

Table 22: Anonymization ablation summary on matched numerical evidence. Lower label-change and numeric-difference rates indicate less sensitivity to coin/date identifiers, but this audit does not prove absence of leakage.

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

D.3 Calibration and Metric-Consistency Audits

The calibration rule is frozen and applied after model output parsing. It does not change probabilities, predicted maximum deviations, cost, latency, token counts, hidden labels, or future data. The 507 release also includes saved metric-consistency and alignment audits under `results/stableeval_507_natural/metrics/verification/` and `results/stableeval_507_natural/reports/`. These files are used as audit artifacts rather than additional leaderboard tables, because the paper-facing values are already reported in Sections B and C.

E Reproducibility Artifacts and Release Checklist

This appendix is designed to function as an audit layer for the main paper. 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 prioritizes frozen case lists, prompt packets, model and platform metadata, raw predictions, calibration rules, evaluator outputs, cost logs, and cross-variant summaries.

Table 23: Main reproducibility artifacts preserved for StableEval Arena. Paths are relative to the release root.

Artifact	Purpose
`data/benchmark/stableeval_507_natural/labels.csv`	Full frozen 507-case label file used as the source label table for benchmark evaluation.
`data/benchmark/stableeval_507_natural/case_list.csv`	Full 507-case natural-distribution case list.
`data/benchmark/stableeval_507_natural/prompt_packets.jsonl`	Full 507-case risk-sensitive prompt-packet file containing lookback-only evidence and excluding hidden future labels.
`data/benchmark/stableeval_507_natural/full_hourly_windows.csv.gz`	Expanded frozen lookback and horizon rows used for audit and reconstruction.
`data/benchmark/stableeval_120_enriched/prompt_packets.jsonl`	120-case stress-enriched prompt packets.
`results/stableeval_120_enriched/validation_case_list.csv`	Selected 120-case stress-enriched validation set used for the main reported validation results.
`prompts/templates/risk_sensitive_agent_prompt.md`	Frozen risk-sensitive agent prompt template shared across evaluated platforms.
`results/stableeval_120_enriched/model_backend.json`	Frozen model-backend metadata for the 120-case validation protocol.
`results/stableeval_120_enriched/calibration_rule.md`	Frozen calibrated-label mapping used to convert model risk estimates into stable/watch/stress labels.
`results/stableeval_120_enriched/agent_execution_metadata.csv`	Platform package, execution-mode, native-runtime, and adapter metadata.
`results/stableeval_120_enriched/predictions_base_raw/`	Saved first-pass 120-case raw agent predictions.
`results/stableeval_120_enriched/predictions_base_calibrated/`	Saved 120-case calibrated predictions used for the main validation table.
`results/stableeval_120_enriched/evaluation_base_calibrated/`	Evaluator outputs for the main 120-case validation result.
`results/stableeval_120_enriched/final_tables/`	Paper-facing 120-case summary, stress, cost, and ablation tables.
`results/stableeval_120_enriched/freeze_manifest/`	120-case freeze manifest and checksum summary used to audit that later scaling work did not alter frozen validation artifacts.
`results/stableeval_507_natural/raw_predictions/`	Saved raw 507-case agent predictions.
`results/stableeval_507_natural/calibrated_predictions/`	Saved calibrated 507-case agent predictions.
`results/stableeval_507_natural/baselines/`	507-case non-agent baseline predictions.
`results/stableeval_507_natural/final_tables/stableeval_507_natural_v6_leaderboard.csv`	Canonical full 507-case leaderboard containing all six agentic systems and four non-agent baselines.
`results/stableeval_507_natural/metrics/`	Per-system 507-case metric outputs and with-actuals files used to reconstruct predictive, risk, reliability, cost, and binary depeg metrics.
`results/stableeval_507_natural/costs/`	507-case cost, latency, and token accounting files for agentic systems and baselines.
`results/stableeval_507_natural/metadata/calibration_rule.md`	Frozen post-hoc calibration rule applied to 507-case agent predictions.
`results/stableeval_507_natural/metrics/verification/`	507-case case-label, calibration, metric-recomputation, and table-version consistency checks.
`results/stableeval_507_natural/reports/`	120-versus-507 alignment and scaling-run interpretation reports.
`protocols/stableeval_507_natural_protocol.json`	Frozen 507-case protocol descriptor, including backend, temperature, systems, prompt path, and canonical output paths.
`scripts/evaluate/recompute_stableeval_120_enriched.py`	No-API recomputation script for the main 120-case saved-output metrics.
`scripts/evaluate/recompute_stableeval_507_natural.py`	No-API recomputation script for the 507-case v6 leaderboard, stress audit, and cost/reliability summary.
`FILE_MANIFEST.md`	Release-facing file manifest and checksums.
`REPRODUCIBILITY.md`	Reproduction guide separating saved-output checks from optional API-backed reruns.