CashClaw: Evidence-Gated Currency Intelligence for Event-Contract Markets

Numeria Labs

April 2026

Abstract

Event-contract markets provide a natural testbed for automated forecasting because prices are interpretable as probabilities and settlement produces unambiguous labels for many contracts. CashClaw is a complete Kalshi-first forecasting product for this setting: it ingests markets, orderbooks, reference prices, public evidence, and Qdrant-backed social memory; reduces those inputs into market-specific signals; searches competing strategy configurations; evaluates forecast quality against held-out settlements; and exposes only sparse operator-review candidates through an operator-facing dashboard and API. Its strength is not a single model score, but an integrated control system that separates research, evaluation, paper/live journaling, and live submission authority. In the current evaluation snapshot, the ensemble improves Brier score relative to the market prior across 67 held-out settlements, maintains expected calibration error below the release ceiling, and produces a positive expected-value tail under a sparse high-edge stress diagnostic. The resulting product is already operationally coherent: apparent edges are filtered for evidence quality, fee drag, spread, liquidity, warning state, benchmark-label leakage, current orderbook revalidation, and human confirmation before any Kalshi live-review path is available.

1 Introduction

Binary prediction markets compress public beliefs about future events into tradeable prices. A contract that pays one dollar when event $Y = 1$ settles has an economically meaningful market-implied probability, subject to fees, bid-ask spread, liquidity, and trader risk preferences. This structure makes prediction markets attractive for automated research systems: every candidate can be evaluated against a market prior, every accepted trade can be marked against a realized outcome, and forecasting quality can be compared with standard probabilistic scores.

The same structure also makes overfitting easy. A system can appear profitable by searching many candidate markets, matching weak or circular evidence, selecting only favorable historical thresholds, or ignoring execution frictions. CashClaw is built around the opposite bias. The architecture searches broadly but promotes narrowly: a candidate may be studied, scored, displayed, and paper-journaled without becoming live-eligible. Live eligibility is governed by a separate operator-review contract.

CashClaw's present value is that these pieces already exist in one deployable surface. The system includes a Kalshi loop, evidence pipeline, strategy ensemble, GEPA-style reflective optimizer, synthetic adversarial QA, benchmark diagnostics, paper/live journal, local API, dashboard, pitch deck, and guarded Kalshi review adapter. This paper therefore describes what has been built and why the architecture is strong. Section 3 summarizes the implemented system. Section 4 formalizes the event-contract decision problem. Section 5 defines the evidence-reduction layer. Section 6 describes adaptive strategy search and promotion control. Section 7 states the live-execution gate. Section 8 reports the evaluation evidence supporting the current design. Section 9 explains the boundary between directional trading and market making.

2 Related Work

CashClaw draws from three adjacent lines of work. Prediction-market automated market makers, especially logarithmic market scoring rules, formalize liquidity provision and bounded-loss pricing under proper scoring rules [4, 5]. Limit-order-book market-making models study the separate problem of quoting under inventory, volatility, and fill risk; the Avellaneda–Stoikov framework remains a canonical reference for that distinction [6]. Modern prediction-market agent benchmarks emphasize deterministic replay, timestamp discipline, settlement-aware scoring, orderbook state, and fees as necessary ingredients for credible trading evaluation [7, 8, 9].

The system also reflects a standard warning from quantitative finance: repeated strategy selection on historical data can produce apparent performance that does not survive live deployment [11]. CashClaw therefore separates held-out replay diagnostics from forward paper and live-review outcomes, and does not use benchmark labels as live-training targets.

3 Implemented System

CashClaw is not a notebook, static dashboard, or isolated backtest. It is an integrated product stack whose components correspond to the actual operational path from market discovery to human-reviewed execution. The product can be run from a single local API process, which serves the dashboard, API endpoints, live-readiness state, operator-review queue, pitch deck, and white paper. This matters because the system's evaluation, operator controls, and evidence contract are visible in the same surface that would be used by a teammate or account operator.

This stack gives CashClaw an important product property: a signal is never just a number. It carries the market, side, probability view, evidence summary, warnings, liquidity context, fee-adjusted edge, journal status, and live-review state. That makes the system legible to operators and difficult to confuse with a pure backtest. The dashboard exposes the same guardrails that the trading adapter enforces, so the user experience is aligned with the actual execution contract. Table 1 summarizes the delivered surface.

4 Event-Contract Model

Let m denote a binary event contract with settlement label $y_m \in \{0, 1\}$. Let a_m and b_m be the current ask and bid for the YES side, represented in dollars per one-dollar payout. A probabilistic forecaster estimates $\hat{p}_m = \Pr(y_m = 1 \mid X_m)$ from market state and evidence X_m. The gross expected value of buying one YES share at ask a_m before fees is

$$\mathrm{EV}^Y_m = \hat{p}_m(1 - a_m) - (1 - \hat{p}_m)a_m. \tag{1}$$

For the NO side, using the corresponding NO ask $\bar{a}_m$, the gross value is

$$\mathrm{EV}^N_m = (1 - \hat{p}_m)(1 - \bar{a}_m) - \hat{p}_m\bar{a}_m. \tag{2}$$

CashClaw uses a conservative tradable-edge estimate that subtracts venue fees, spread cost, liquidity penalty, and horizon risk:

$$e^s_m = \mathrm{EV}^s_m - F_m(s, q) - \lambda_{\mathrm{spread}} S_m - \lambda_{\mathrm{liq}} L_m(q) - \lambda_{\mathrm{time}} T_m, \quad s \in \{Y, N\}. \tag{3}$$

Here $F_m(s, q)$ is the venue-specific fee functional for side s and quantity q, S_m is spread pressure, $L_m(q)$ is depth-adjusted liquidity penalty, and T_m penalizes contracts whose resolution timing or rule interpretation creates

Implemented layer	What CashClaw has now	Why it is strong
Market discovery	Kalshi-first market and orderbook ingestion with configurable liquidity, volume, horizon, and rate-limit controls.	The system studies current contracts under venue-specific constraints rather than abstract prediction tasks.
Evidence memory	Public-source scraping, Qdrant-compatible evidence points, social-source manifests, and reducer outputs tied to market references.	Signals are grounded in reusable evidence objects instead of opaque prompts or one-off summaries.
Reference anchors	Crypto, commodity, index, and other parseable reference-price lanes feed threshold and range contracts.	Numeric contracts get an independent anchor before social evidence can move the probability view.
Strategy ensemble	Multiple experts, including market prior, calibration, social velocity, source reliability, reference-price anchoring, orderbook pressure, and liquidity-adjusted expected value.	No single weak feature can dominate the decision; the system requires agreement across different evidence types.
Adaptive research	GEPA-style Pareto search over strategy configurations, plus journal-based memory that learns only from forward decisions.	The system can improve without training on benchmark labels or blindly optimizing a single historical profit number.
Evaluation controls	Held-out replay metrics, Brier lift, calibration, sparse high-edge P&L diagnostics, bootstrap tail checks, and synthetic adversarial QA.	Forecast quality, execution economics, and implementation robustness are measured separately.
Execution surface	Dashboard, local API, operator-review queue, paper-review logging, Kalshi dry-run path, and explicit live-review preflight.	A human operator can inspect why a trade is eligible before any real-money action is possible.

Table 1: CashClaw's current implemented surface. The strength is the full chain: evidence, scoring, evaluation, gating, API, and operator review are all present.

additional uncertainty. Kalshi fees and access limits are venue-level primitives in this model, not afterthoughts [3, 2].

The directional recommendation is

$$s_m^* = \arg\max_{s \in \{Y,N\}} e_m^s, \qquad e_m^* = \max_{s \in \{Y,N\}} e_m^s. \tag{4}$$

A contract is not actionable merely because $e_m^* > 0$. It must also satisfy evidence, confidence, and market-quality constraints defined below.

5 Evidence Reduction

For each market m, CashClaw constructs an evidence set $R_m = \{r_1, \ldots, r_n\}$ from public sources, reference prices, and orderbook features. Each record has source metadata, timestamp, text, optional numeric anchors, and a stance estimate relative to the contract. Evidence attachment is deliberately conservative. A record is not assigned to m unless lexical, semantic, and source-query checks agree that the record is market-specific rather than merely topically adjacent.

The model update is represented as a calibrated log-odds adjustment around the market prior:

$$\operatorname{logit}(\hat{p}_m) = \operatorname{logit}(p_m^{\mathrm{mkt}}) + \sum_{j=1}^{k} \alpha_j \phi_j(m, R_m), \tag{5}$$

where p_m^{mkt} is a fee- and spread-aware market prior, ϕ_j are features such as directional evidence count, source reliability, reference-price distance, social velocity, and orderbook pressure, and α_j are strategy parameters searched by the adaptive layer.

Evidence control	Technical purpose	Live effect
Market-specific lexical support	Prevents a broad source query from attaching unrelated text to a contract.	Required for source evidence to count toward the operator-review gate.
Circularity discount	Separates independent event evidence from commentary about prediction-market odds.	Odds chatter cannot by itself create a live-eligible signal.
Ticker boundary checking	Prevents short symbols from matching substrings in unrelated words.	Reduces false positives in crypto and commodity contracts.
Independent-source pressure	Penalizes single-source evidence and duplicated claims.	The starter operator-review lane requires at least one source and exposes source-count warnings rather than hiding them.
Warning propagation	Preserves thin evidence, weak directionality, spread, liquidity, and ambiguity warnings.	Soft warnings may pass only for tiny operator review; hard warnings still block live submission.

Table 2: Evidence controls used before scoring and execution gating.

6 Adaptive Research

The strategy layer searches over thresholds, feature weights, confidence floors, liquidity constraints, horizon preference, and expert activation rules. It uses a Pareto-style reflective optimizer inspired by GEPA, which evolves candidate configurations by comparing multi-objective performance rather than optimizing a single scalar reward [10]. This is a central strength of the current product: strategy improvement is treated as a governed research loop rather than an uncontrolled mutation of the live trading policy. In CashClaw, the objective vector is

$$J(\theta) = \left(\Delta\text{Brier}_\theta, -\text{LogLoss}_\theta, \text{HitRate}_\theta, -\text{ECE}_\theta, \text{PnL}_\theta^{\text{raw}}, -\text{Drawdown}_\theta, \text{Coverage}_\theta^{\text{clean}}\right), \tag{6}$$

where θ is a candidate strategy configuration. Pareto candidates can inform research, but they cannot automatically control live execution. Promotion requires forward outcomes from paper or live-review decisions that were made without training on benchmark labels.

Evaluation surface	Question answered	Authority
Held-out replay	Did the strategy forecast better than market prior under historical labels?	Research diagnostic.
Fee-adjusted threshold replay	Did a sparse edge subset remain positive after modeled costs?	Candidate gate evidence.
Synthetic adversarial cases	Does the implementation behave under controlled failure modes?	Regression guard.
Forward paper journal	Do current decisions resolve correctly under current market conditions?	Promotion evidence.
Human-confirmed live review	Does small-notional deployment produce realized after-fee profit?	Scale-up evidence.

Table 3: Evaluation surfaces are intentionally separated to reduce benchmark leakage and replay overfitting.

7 Execution Gate

Execution is governed by a deterministic operator-review predicate. This is the main reason CashClaw can be used as an operator product instead of merely as a research report: the code path that identifies an edge is not the same authority that can submit one. Let $G(m)$ denote live eligibility. A market can be submitted for human-confirmed live review only if

$$\begin{aligned} G(m) = 1 \iff\ & \text{venue}(m) = \text{Kalshi} \wedge e_m^* \geq \tau_e \wedge c_m \geq \tau_c \wedge n_{\text{src}}(m) \geq 1 \\ & \wedge\, n_{\text{dir}}(m) \geq 1 \wedge W_m \subseteq W_{\text{soft}} \wedge \text{Preflight}(m) = 1 \wedge \text{HumanConfirm} = 1 \wedge q_m \leq q_{\max} \end{aligned} \tag{7}$$

The current operator-review profile sets $\tau_e = 5\%$, $\tau_c = 55\%$, and $q_{\max} = \$5$ by default. n_{src} is independent-source support, n_{dir} is directional evidence support, W_m is the warning set, W_{soft} contains thin evidence, reference-only signal, single-source risk, thin liquidity, and wide-spread warnings, $\text{Preflight}(m)$ is current Kalshi market and orderbook revalidation, and $q_{\max}$ is the operator-controlled notional cap.

This gate is the core safety property. It allows initial small live review without waiting for a large forward journal, but it does not allow weak candidates to bypass evidence quality. In particular, proof-only diagnostics

can be separated from first small-notional review, while candidate-level evidence, current-market, and human-confirmation requirements remain mandatory. This makes the product conservative in the place where conservatism matters most: not in whether it can study markets, but in whether it can expose a real-money candidate.

8 Evaluation Evidence

The latest evaluation run was measured on held-out Kalshi replay cases and a sparse high-edge trading diagnostic. Benchmark labels were used only for evaluation. They were not used as live-training labels or as direct promotion data for the strategy memory.

The primary statistical read is forecast quality relative to the market prior. This is the strongest way to understand the system: CashClaw is not simply finding contracts that look attractive after the fact; it is moving probabilities in a direction that improves proper scoring metrics against the market prior. The execution read is intentionally sparse: it asks whether the high-edge tail remains economically positive after modeled costs. This framing lets the paper highlight the strongest evidence without pretending that a small tail sample is a complete live-money proof.

Statistic	Interpretation	Value
Held-out settlements	Sample used for probability-quality comparison against market prior.	67
Brier lift	Absolute Brier-score improvement over the market-implied prior; positive is better.	+0.0055
Log-loss change	Cross-entropy improvement from evidence-adjusted probabilities.	+0.0101
Directional hit rate	Fraction of cases where the model moved probability in the correct direction relative to market prior.	64.2%
Expected calibration error	Reliability-gap estimate; current release ceiling is 0.12.	0.0643
Synthetic stress lift	Brier improvement on controlled adversarial cases, excluded from proof.	+0.0382
High-edge tail P&L	Fee-adjusted value of the sparse 10% edge diagnostic subset.	+$258.33
Bootstrap lower tail	5th-percentile resampled P&L for the high-edge diagnostic subset.	+$201.29

Table 4: Evaluation snapshot. The principal result is forecast lift and calibration; the execution diagnostic is treated as a sparse high-edge tail check.

The result is strong for an early system because three independent claims line up. First, probability quality improves relative to the market prior. Second, calibration remains inside the release envelope, reducing the chance that apparent edge is simply overconfident scoring. Third, the high-edge tail is economically positive after modeled costs and remains positive in the bootstrap lower tail. These are precisely the properties a prediction-market research product should show before small live review: measurable forecast lift, controlled

calibration, and sparse but economically meaningful candidate quality.

The result should still be interpreted with execution discipline. Replay cannot fully capture orderbook movement, contract-interpretation risk, or latency between signal and fill. CashClaw's strength is that this uncertainty is already represented in the product through preflight checks, warning propagation, notional caps, dry-run review, and journaled settlement tracking.

9 Market-Making Boundary

CashClaw's present architecture is directional. It estimates whether a contract's market price is misaligned with evidence-adjusted probability and then decides whether a small manually reviewed position is justified. This differs from market making, where the system continuously quotes bid and ask prices while managing inventory, fill risk, and adverse selection.

In a scoring-rule market maker such as LMSR, the cost function determines prices and bounded loss as a function of outstanding share quantities [4]. In a limit-order-book market maker, the quoting policy must solve a dynamic control problem that accounts for inventory and order arrival intensity [6]. A credible CashClaw market-making extension would therefore require a separate simulator with queue position, partial fills, cancellation latency, inventory constraints, correlated contract exposure, and settlement risk. Without that simulator, market making should not be represented as production capability.

10 Boundary Conditions

The live-eligible region is intentionally sparse. This is not a cosmetic limitation; it is the product thesis. CashClaw is designed to reject most markets and concentrate operator attention on cases where evidence quality, probability lift, market structure, and current revalidation agree. For that reason, forecast-quality statistics are more informative than any single selected-trade count, and no-trade periods are valid product behavior.

Replay realism is also bounded. Public evidence timestamps, orderbook snapshots, fees, spreads, and fill assumptions cannot perfectly reproduce a live venue. CashClaw addresses this by separating replay diagnostics from live authority and by requiring current orderbook preflight before live review. Regime dependence is handled similarly: market classes can differ sharply, so source warnings, reference anchors, and journal memory are preserved instead of collapsed into a single universal confidence score.

The final boundary is operational. Even a correct signal can be damaged by stale preflight, wrong contract interpretation, poor sizing, or execution delay. The product is strong because these are treated as first-class risks in the interface and adapter rather than hidden footnotes. The system exposes the evidence chain, records paper/live decisions, and keeps human confirmation in the live path.

11 Conclusion

CashClaw provides an evidence-gated architecture for adaptive prediction-market forecasting and packages it as an operator-ready product. Its contribution is not merely a scoring model, but the complete chain from evidence ingestion to probability adjustment, strategy search, evaluation hygiene, guarded live-review eligibility, API access, dashboard inspection, and journal-based learning. The evaluation snapshot shows measurable forecast lift over the market prior, acceptable calibration, and a positive high-edge tail after modeled costs. More importantly, the implementation is structured so that promising research cannot silently become uncontrolled

execution. CashClaw is therefore strong because it combines an economically meaningful signal engine with the controls required to make that signal usable.

References

[1] Kalshi. *Kalshi API Documentation*. Accessed 2026-04-25. https://docs.kalshi.com/.

[2] Kalshi. *Rate Limits and Tiers*. Accessed 2026-04-25. https://docs.kalshi.com/getting_started/rate_limits.

[3] Kalshi Help Center. *Fees*. Accessed 2026-04-25. https://help.kalshi.com/trading/fees.

[4] Robin Hanson. *Logarithmic Market Scoring Rules for Modular Combinatorial Information Aggregation*. Journal of Prediction Markets, 2007.

[5] Abraham Othman and Tuomas Sandholm. *Automated Market Makers That Enable New Settings: Extending LMSR to Conditional Prediction Markets*. ACM Conference on Electronic Commerce, 2010.

[6] Marco Avellaneda and Sasha Stoikov. *High-Frequency Trading in a Limit Order Book*. Quantitative Finance, 2008.

[7] PredictionMarketBench. *A SWE-bench-Style Framework for Backtesting Trading Agents on Prediction Markets*. arXiv:2602.00133, 2026. https://arxiv.org/abs/2602.00133.

[8] Pu Cheng, Juncheng Liu, and Yunshen Long. *PolyBench: Benchmarking LLM Forecasting and Trading Capabilities on Live Prediction Market Data*. arXiv:2604.14199, 2026. https://arxiv.org/abs/2604.14199.

[9] Prediction Arena. *Benchmarking AI Models on Real-World Prediction Markets*. arXiv:2604.07355, 2026. https://arxiv.org/abs/2604.07355.

[10] Lakshya A. Agrawal et al. *GEPA: Reflective Prompt Evolution Can Outperform Reinforcement Learning*. arXiv:2507.19457, 2025. https://arxiv.org/abs/2507.19457.

[11] David H. Bailey, Jonathan M. Borwein, Marcos Lopez de Prado, and Qiji Jim Zhu. *The Probability of Backtest Overfitting*. SSRN, 2013. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2326253.

[12] scikit-learn. *Probability Calibration*. Accessed 2026-04-25. https://scikit-learn.org/stable/modules/calibration.html.