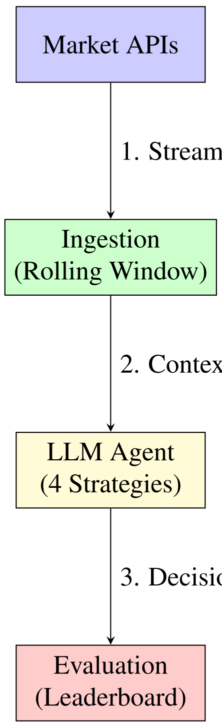

Figure 1. TruthTensor System Architecture.

tions as the definitive reference frame for evaluation. Each row corresponds to a model or agent evaluated during the same market window, while the columns capture both market-aligned behavioral signals and agent-level outcomes relative to that market.

- P&L (profit and loss) quantify the economic consequence of these deviations when agent decisions are hypothetically executed against the market. Positive P&L values indicate that an agent's probability estimates improve upon the market consensus in expectation, while negative values reflect miscalibration or delayed information assimilation.
- Unique Users and Agent Count provide context on interaction scale and deployment intensity during the evaluation window. Higher user counts indicate broader exposure to market conditions, while agent count reflects the degree of parallel decision-making evaluated under the same baseline.
- Average Input Tokens and Average Output Tokens describe computational characteristics of each agent. These values contextualize efficiency trade-offs: for instance, agents with comparable P&L but significantly lower token usage demonstrate more efficient reasoning relative to the same market baseline.

3.3. Agent Deployment Layer. Once a prompt is specified and baselines are constructed, TruthTensor generates an AI forecasting agent instantiated with:

- a selected LLM backend (frontier, mid-tier, or distilled),
- an agentic orchestration framework (e.g., chain-of-thought pipelines, tool-augmented agents, or minimal single-call evaluators),
- optional access to external data streams (news feeds, APIs, retrieval systems),
- token budget constraints that limit reasoning length,
- a sandboxed execution environment that ensures deterministic runs and safety constraints, and
- drift tracking instrumentation that logs reasoning traces, probability estimates, and confidence scores at each time point.

The agent is deployed as a persistent service that re-samples, re-reasons, and re-predicts at predefined intervals. Each forecast is logged together with the reasoning trace, model version, latency profile, token usage, tool invocations, and drift metrics relative to previous time points. This structure enables decomposition of performance into:

- model-intrinsic forecasting ability,
- tool-assisted reasoning improvements,
- temporal updating fidelity,
- drift magnitude and patterns, and
- reasoning confidence alignment.

Each deployed agent therefore acts as an isolated experimental unit capable of generating repeated, timestamped probability estimates with full drift tracking.

3.4. Market-Linked Execution Layer with Drift Tracking. The final stage connects the agent's forecasts to real-world prediction market prices while continuously tracking drift patterns. The current implementation integrates with the Polymarket platform, whose markets encode continuously updated, financially backed probability estimates. TruthTensor supports two operational modes.

Observation Mode. The agent retrieves real-time market prices and compares its internal forecasts to the market-implied probabilities. This mode supports calibration scoring, divergence analysis, drift measurement, and studies of narrative drift over time. Drift metrics are computed at each time point, tracking how model outputs evolve relative to market dynamics and baseline models.

Execution Mode. The agent executes live trades according to a predefined strategy (e.g., buying when the agent's probability exceeds the market probability by more than a threshold δ). Trades serve as ground-truth commitments: the agent effectively stakes its inference on real-world outcomes. Execution logs include fill price, position size, PnL evolution, liquidation timestamps, and drift metrics. This converts evaluation into a monetizable high-entropy reasoning test reflective of practical decision quality. Trading is rate-limited and bounded by safety constraints to maintain experimental reproducibility.

3.5. Integrated Evaluation Loop. The architectural pipeline operates as a closed-loop system:

- Lock instructions: formalize a forecasting instruction under strict parameterization with versioning and immutability.
- Construct baselines: establish reference points independent of rolling-window calibration.
- Deploy an AI agent: instantiate a controlled and reproducible reasoning system with drift tracking.
- Execute on Polymarket: observe forecasting accuracy, calibration, trading performance, and drift patterns in real-time markets.

Table 2. Benchmarked during the evaluation window.

Model	P&L	Unique Users	Agent Count	Avg Input Tokens	Avg Output Tokens
Kimi-K2-Thinking	$-3,983,370.92	76,369	108,281	3,794	3,166
Claude-Sonnet-4.5	$-14,276,936.58	53,854	73,095	3,994	713
GPT-5.1	$-6,605,722.68	46,479	59,257	3,961	3,179
Grok-4	$-3,596,533.38	45,678	58,268	4,439	3,930
Gemini-3-Pro-Preview	$-2,604,933.30	45,520	57,656	6,725	4,611
DeepSeek-Chat-v3.1	$-5,074,116.67	43,980	56,194	4,106	717
Qwen3-Max	$-3,105,378.52	43,618	54,936	4,377	429
Minimax-M2	$-3,551,465.32	41,084	50,674	4,000	1,532

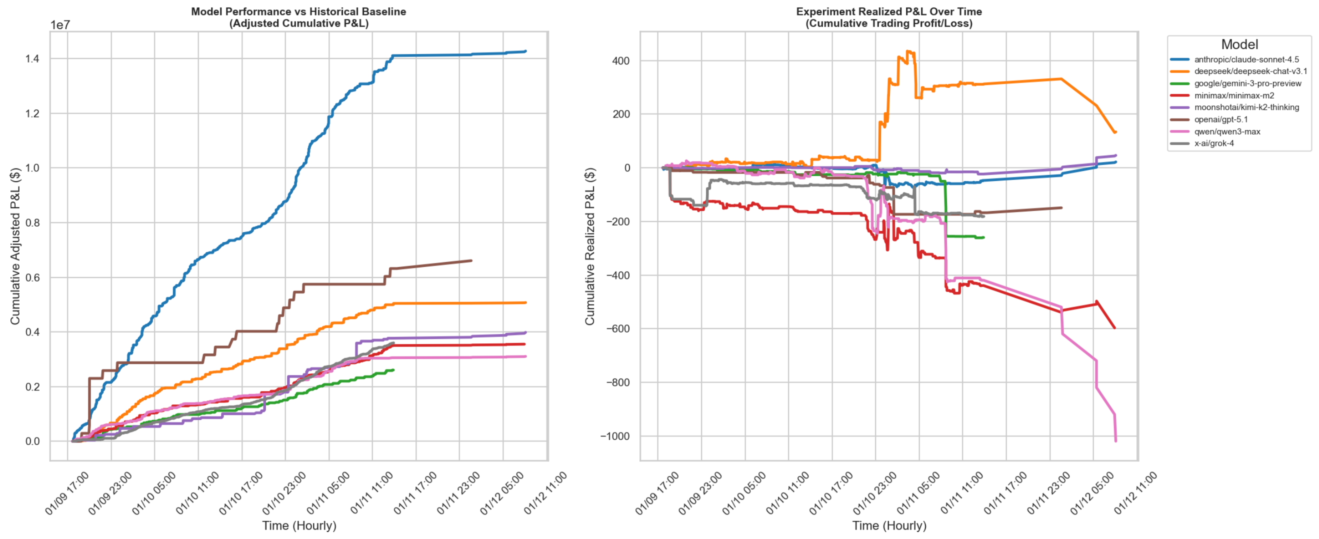

Figure 2. Adjusted cumulative P&L versus historical baseline (left) and realized cumulative trading P&L (right).

- Measure drift: compute narrative, temporal, and confidence drift metrics relative to baselines and market dynamics.

This end-to-end design provides a principled mechanism for measuring LLM human imitation capabilities, free from static dataset contamination and anchored to real-world probabilistic outcomes. TruthTensor therefore transforms forecasting into a scientific probe of model reasoning, temporal coherence, narrative stability, and drift patterns—essential dimensions for assessing human imitation capabilities.

4. Evaluation Methodology

TruthTensor employs a holistic evaluation methodology that measures correctness, risk assessment, temporal coherence, calibration, and drift patterns. The framework categorizes events by risk profile, domain, and temporal horizon, enabling targeted analysis of model strengths and weaknesses across different contexts.

4.1. Event Categorization. Events are categorized along multiple dimensions:

- Risk Profile: Low-risk (high-probability outcomes), medium-risk (balanced probabilities), high-risk (low-probability, high-impact events).
- Domain: Political (elections, policy outcomes), economic (market movements, indicators), cultural (awards, trends), technological (product launches, breakthroughs).
- Temporal Horizon: Short-term (resolves within days), medium-term (weeks to months), long-term (months to years).
- Market Liquidity: High-liquidity (active trading), medium-liquidity (moderate activity), low-liquidity (limited trading).

This categorization enables targeted evaluation: models may excel in low-risk, short-term political events but struggle with high-risk, long-term technological predictions. By measuring performance across categories, TruthTensor provides a comprehensive view of model capabilities.

4.2. Specified Evaluation Metrics. TruthTensor employs a comprehensive set of specified evaluation metrics.

Correctness Metrics

- Brier Score: Measures the accuracy of probability forecasts, with lower scores indicating better accuracy [27].
- Log-Likelihood: Evaluates the probability assigned to the actual outcome, rewarding well-calibrated forecasts.
- Accuracy: Binary correctness rate for thresholded predictions.

Calibration Metrics.

- Expected Calibration Error (ECE): Measures the difference between predicted confidence and actual accuracy across probability bins.

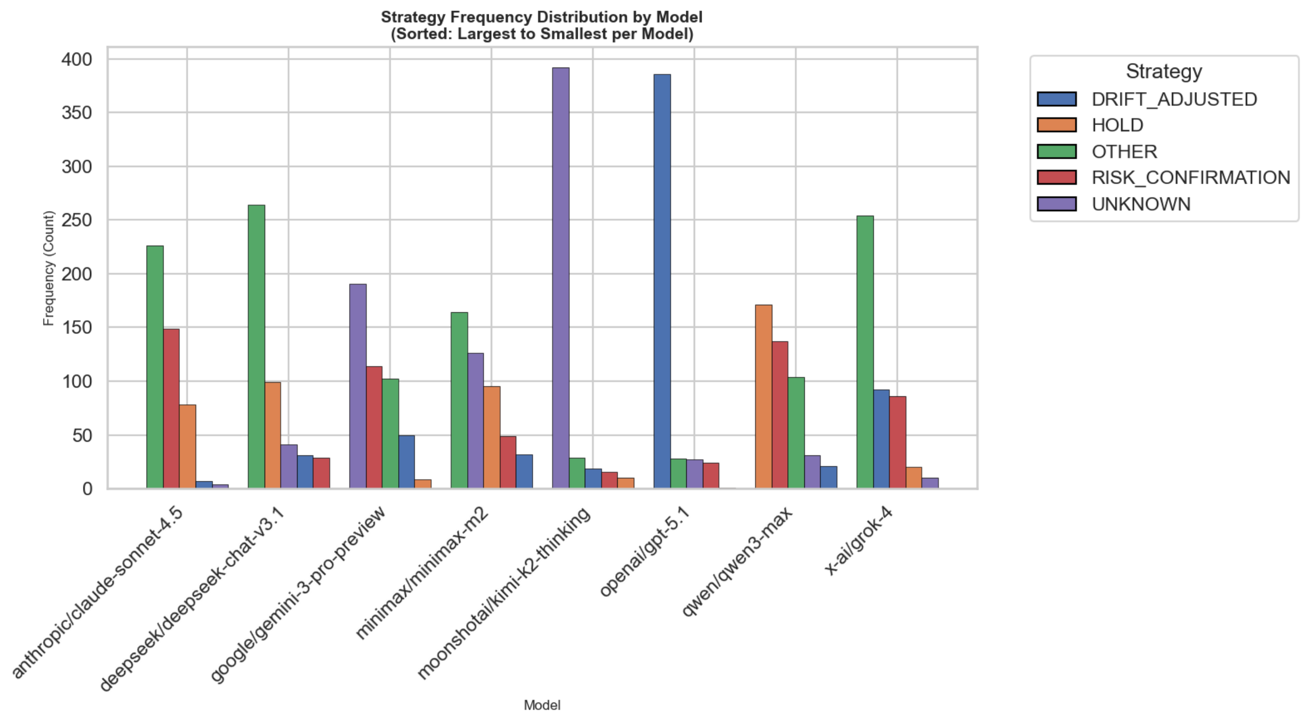

Figure 3. Distribution of strategy selection frequency by model.

- Maximum Calibration Error (MCE): Captures the worst-case calibration error.
- Reliability Diagrams: Visualize calibration across probability ranges.

Drift Metrics.

- Narrative Drift Score: Quantifies reasoning trace inconsistency over time.
- Temporal Drift Score: Measures update appropriateness relative to information arrival.
- Confidence Drift Score: Assesses confidence-reasoning alignment over time.
- Market Divergence: Tracks how model outputs diverge from market-implied probabilities over time.

Risk Assessment Metrics.

- Value at Risk (VaR): Measures potential losses under adverse scenarios.
- Conditional Value at Risk (CVaR): Assesses tail risk beyond VaR thresholds.
- Risk-Adjusted Returns: Evaluates performance relative to risk exposure.

Reasoning Confidence Metrics.

- Confidence-Reasoning Alignment: Measures correlation between stated confidence and reasoning quality.
- Confidence Calibration: Assesses whether confidence levels match actual accuracy.
- Confidence Stability: Tracks confidence consistency across time points.

Holistic Composite Metrics.

- Human Imitation Score: Weighted combination of correctness, calibration, drift, and risk metrics, measuring overall similarity to human reasoning patterns.
- Reasoning Quality Index: Integrates reasoning trace quality, confidence alignment, and narrative coherence.

4.3. Token Constraint Evaluation. TruthTensor explicitly evaluates how reasoning quality degrades under token budget constraints. Models are evaluated across multiple token budget levels (e.g., 500, 1000, 2000, 4000 tokens), measuring:

- whether probability estimates remain stable under constraints,
- how drift patterns vary with available reasoning capacity.

This evaluation reveals which models maintain human-like reasoning quality even under resource constraints, an important consideration for real-world deployment.

4.4. Baseline Comparison Protocol. Models are compared against baselines using a systematic protocol:

(1) Baseline Computation: Baselines are computed independently for each event category and time point.
(2) Model Evaluation: Models are evaluated using the same metrics as baselines.
(3) Statistical Testing: Significance tests determine whether model performance differs meaningfully from baselines.
(4) Drift Analysis: Drift patterns are compared across models and baselines, revealing which models best replicate human-like stability.

This protocol ensures fair comparison and prevents rolling-window calibration from masking model limitations.