LiveCodeBench [16] is a benchmark designed to evaluate the capabilities of large language models (LLMs) in code-related tasks, with an emphasis on holistic and contamination-free evaluation. Unlike traditional code-based benchmarks that often rely on static datasets or tasks, LiveCodeBench constructs diverse challenges that test LLMs on real-world programming scenarios, such as bug detection, code completion, and refactoring. The benchmark ensures that the test set is free from contamination by pre-existing model outputs, providing a more reliable assessment of LLM performance. Evaluation is based on a range of metrics, including accuracy, code efficiency, and robustness, to assess models' generalization across various programming domains.

InvestorBench [17] is a benchmark framework proposed for evaluating AI agents' capabilities to forecast and reason about financial markets and investment decisions. It frames evaluation as a forecasting task where models must predict asset price movements, financial outcomes, or market trends over defined horizons. Unlike static datasets, InvestorBench's tasks are forward-looking: at prediction time, the ground truth has not yet been realized. Performance is evaluated according to probabilistic forecasts, calibration, and decision quality, capturing not just whether a model can guess the right outcome, but how well it can reason under financial uncertainty, assess risk, and update predictions over time. This benchmark explicitly targets financial-market reasoning, bridging the gap between traditional AI evaluation (e.g., language, reasoning, classification) and real-world decision tasks where stakes, uncertainty, and temporal dynamics matter.

MIRAI [18] is an evaluation framework (and associated environment) aimed at assessing LLMs in interactive, agentic, and decision-making contexts. Rather than evaluating models through static prompts and single-run outputs, MIRAI emphasizes agentic execution, allowing LLM-based agents to interact with external tools or environments, perform multi-step reasoning or actions, and adapt over time. The framework supports modular agent composition, enabling evaluation of not just the base language model, but the entire decision-making stack including tool usage, planning, and dynamic context handling.

Prophet Arena [19] is a dynamic benchmarking platform designed to evaluate AI agents' forecasting capabilities in real-world, uncertain environments. Unlike static evaluation datasets, Prophet Arena emphasizes forward-looking tasks where outcomes are not yet known at the time of prediction, ensuring that models are tested on their ability to reason under uncertainty rather than memorize historical data. The framework aggregates probabilistic predictions from AI agents across a variety of socially and economically relevant events, including political, cultural, and financial phenomena. Evaluation focuses on calibration, accuracy, and adaptability over time, enabling the assessment of both short-term and long-term forecasting performance.

Agent Market Arena (AMA)[20] proposes a market-based, multi-agent evaluation ecosystem where autonomous agents interact, negotiate, or trade in simulated or real markets. Under AMA, agents are evaluated not on static benchmark performance but on their strategic behavior, adaptability, and decision-making in a shared environment with other agents. The evaluation includes how agents respond to other agents' behavior, market dynamics, and evolving information, capturing strategic reasoning, temporal adaptation, and emergent group dynamics. AMA represents a shift from static, isolated evaluation toward interactive, socially grounded assessment, and provides a valuable perspective on how AI agents might behave in settings where their decisions influence and respond to others.

LiveTradeBench [21] is a benchmarking methodology designed to evaluate AI agents' performance in live trading environments, using real or simulated financial markets. Instead of evaluating agents on offline historical data or synthetic tasks, LiveTradeBench emphasizes real-time decision-making under uncertainty, requiring agents to consume live market data, make predictions or trading decisions, and manage risk dynamically. Evaluation metrics include not only forecast calibration and prediction quality, but also trade outcome, profit-and-loss (PnL), drawdowns, and risk-adjusted returns. This benchmark assesses an agent's ability to reason under uncertainty, update forecasts with fresh information, and make economically rational decisions in volatile, high-stakes environments, attributes that static benchmarks fail to capture. LiveTradeBench underlines the importance of temporal coherence, risk sensitivity, and decision-making stability in AI agents that operate in real-world financial domains.

Building on these prior efforts, TruthTensor integrates live prediction-market data to create a dynamic, continuously evolving evaluation environment. Unlike existing benchmarks, TruthTensor emphasizes real-time probabilistic reasoning and agentic decision-making, moving beyond recall or static prediction to measure how well AI models think, adapt, and forecast in high-entropy, socially relevant contexts.

As AI continues to evolve, it is clear that the field must leave behind the narrow scope of traditional metrics in favor of dynamic, flexible evaluation methodologies that can capture the true cognitive potential of these systems. The evaluation process must reflect the complexities of real-world interactions and decision-making, ensuring that AI systems are not only competent in completing tasks but are also equipped to handle the unpredictable nature of human interaction, novel scenarios, and evolving challenges.

To contextualize the contributions of TruthTensor relative to existing AI evaluation frameworks, Table 1 summarizes the key features, evaluation types, focus areas, and limitations of several representative benchmarks. As shown, while prior benchmarks such as Humanity's Last Exam, ForecastBench, Futurex, and Chatbot Arena provide valuable insights into general intelligence, probabilistic forecasting, or multi-turn conversational performance, TruthTensor uniquely integrates real-time prediction-market data to evaluate reasoning, adaptation, and probabilistic decision-making under high-entropy social conditions.

2.4. Market Prediction: Human Imitation Ground Truth. Prediction markets and live event feeds provide a natural source of such future-grounded tasks. Platforms like Polymarket [22] continuously aggregate human forecasts into probabilistic predictions about real-world outcomes. By tapping these markets, TruthTensor obtains a dynamic, crowd-sourced scoring target:

Table 1. Comparison of AI evaluation approaches and frameworks.

Framework	Evaluation Type	Primary Focus	Dynamic	Notes
ARC	Static QA benchmark	Grade-school science reasoning	No	Fixed multiple-choice dataset; closed-world evaluation
GSM8K	Static QA benchmark	Multi-step mathematical reasoning	No	Arithmetic word problems; vulnerable to contamination
HumanEval	Static code benchmark	Program synthesis and correctness	No	Small fixed task set; unit-test based
Chatbot Arena	Human preference ranking	Dialogue quality and consistency	Partial	Interactive but not outcome-grounded
Humanity's Last Exam	Multi-domain exam	General reasoning and abstraction	No	Broad coverage; static task design
CORE-Bench	Agentic reproducibility eval	Scientific reproducibility	No	Focused on research workflows, not reasoning drift
LiveCodeBench	Live code benchmark	Real-world programming tasks	Yes	Contamination-resistant; code-only domain
ForecastBench	Probabilistic forecasting	Future event prediction	Yes	Forward-looking; no live market grounding
Futurex	Probabilistic forecasting	Uncertainty-aware future reasoning	Yes	Synthetic or planned events
InvestorBench	Financial forecasting	Investment decision-making	Yes	Financial domain specific
MIRAI	Agentic forecasting	Tool-augmented event prediction	Yes	Interactive agents; no market ground truth
Agent Market Arena (AMA)	Multi-agent market simulation	Strategic interaction and adaptation	Yes	Simulated markets; limited real stakes
LiveTradeBench	Live trading benchmark	Trading performance and risk	Yes	High data-access and infra cost
Prophet Arena	Forecasting arena	Real-world event forecasting	Yes	Dynamic evaluation; limited drift analysis
TruthTensor (ours)	Market-grounded agentic eval	Human imitation, drift, calibration	Yes	Live prediction markets; longitudinal drift tracking

each event's current market odds serve as a calibrated probability forecast that reflects aggregated human reasoning, risk assessment, and narrative interpretation.

This creates a moving performance yardstick that evolves with new information. Since the market's probability estimates encode the wisdom of the crowd, they tend to be well-calibrated and aggregate diverse insights [23]. In this setup, an AI agent's output is compared against the evolving market consensus, so that performance reflects how closely the model's reasoning, confidence, and narrative coherence match human market participants. In practice, this means benchmarking by human imitation: models are scored not on binary correctness but on how well they replicate human-like probabilistic reasoning, calibration, and narrative stability.

2.5. Drift: The Central Evaluation Dimension. Beneath the impressive capabilities of LLMs lies vulnerabilities that threatens their long-term effectiveness and reliability. One such vulnerabilities is drift, the tendency for model outputs to shift in ways that diverge from human-like reasoning patterns. Key among drift manifestations that are fundamental to LLMs include narrative, temporal, and confidence. Narrative drift refers to inconsistent reasoning about the same event over time. It occurs as a result of semantic priming, where stylistic linguistic cues prompt an LLM to transition from providing factual summaries to simulating reality leading to synthetic evidence generation (the fabrication of facts) and claim escalation (the sensationalization of concepts), both of which contribute to a gap in Epistemic Integrity [24]. Narrative drift arises when a model's reasoning about the same event changes in ways that are inconsistent with human-like updating patterns. For example, a model might initially assign high probability to an election outcome based on polling data, then shift to low probability based on the same data without any new information arriving. Such shifts reveal that the model's reasoning process is unstable or inconsistent, failing to maintain narrative coherence in the way humans do. TruthTensor places drift at the center of its evaluation framework. It measures narrative drift by tracking how model probability estimates and reasoning traces evolve over time for the same event. We compute drift metrics including:

- Probability Volatility: Quantifies the magnitude of probability shifts that cannot be explained by new information arrival.
- Reasoning Trace Divergence: Compares reasoning traces at different time points, measuring how much the underlying narrative has shifted.

Models that exhibit high narrative drift are poor human imitators, as they fail to maintain the coherent, stable reasoning patterns that characterize human probabilistic reasoning.

Temporal drift refers to a phenomenon in which the performance and accuracy of language models decline over time, driven by shifts in underlying data distributions, evolving linguistic patterns, and changes in the factual knowledge that the models were originally trained to capture [25]. It measures how well models update their probability estimates as new information arrives. Human market participants continuously incorporate new information, adjusting their probability estimates in ways that reflect Bayesian updating principles.

Models that fail to update appropriately, either by ignoring new information or by overreacting to noise, exhibit temporal drift. The challenge of temporal drift is especially pronounced for LLMs due to their scale and the extensive breadth of knowledge they are designed to encode. In contrast to specialized models, which operate within narrow domains where temporal changes may be more predictable or readily managed, LLMs aim to achieve general-purpose language understanding across nearly all domains of human knowledge and communication. This expansive scope renders temporal drift a multidimensional phenomenon, manifesting concurrently across interdependent aspects of data, language, and factual knowledge, thereby complicating systematic anticipation and mitigation. TruthTensor evaluates temporal drift by:

- Comparing model updates to market updates following information events (news releases, polling updates, etc.).
- Measuring the correlation between information arrival and probability shifts.
- Assessing whether updates are appropriately calibrated to information magnitude.

Models with low temporal drift—those that update in ways consistent with human market participants—are better human imitators than those that exhibit high temporal drift. A key characteristic of human-like reasoning is the recognition and acknowledgment of uncertainty, along with an associated level of confidence in the provided answers. While LLMs rely on statistical prediction techniques, the degree of confidence, both implicit and explicit, in their responses is not immediately apparent [26]. When seeking an expert opinion, it is generally expected that the response will be accompanied by an indication of confidence. This confidence measure is a standard component in statistical expert systems, where a validated correlation between the confidence level and the accuracy of the response is crucial for establishing the credibility of the system. Confidence drift measures the alignment between a model's stated confidence and its actual calibration. Human experts exhibit well-calibrated confidence: when they express high confidence, their predictions tend to be correct; when they express low confidence, their predictions are more uncertain. Models that exhibit confidence drift, which refers to overconfidence or underconfidence relative to their actual accuracy, fail to replicate human-like calibration patterns. TruthTensor measures confidence drift using:

- Calibration Error: The difference between stated confidence and actual accuracy across probability bins.
- Overconfidence Index: Measures the extent to which models express higher confidence than their accuracy warrants.
- Confidence-Reasoning Alignment: Assesses whether stated confidence correlates with reasoning quality and information availability.

Models that exhibit low confidence drift, well-calibrated confidence that aligns with reasoning quality, are better human imitators than those with high confidence drift.

2.6. Drift Measurement Methodology . TruthTensor measures drift through a systematic sampling protocol:

(1) Event Selection: Events are selected from prediction markets, categorized by risk profile, domain, and temporal horizon.
(2) Time-Series Sampling: For each event, models are queried at regular intervals (e.g., daily) until resolution.
(3) Drift Computation: At each time point, drift metrics are computed relative to baseline models and market-implied probabilities.
(4) Aggregation: Drift scores are aggregated across events, categories, and time horizons to produce comprehensive drift profiles.

This methodology enables systematic comparison of drift patterns across models, revealing which models best replicate human-like reasoning stability and coherence.

3. System Architecture

TruthTensor implements a modular, end-to-end architecture that enables the systematic evaluation of LLMs as human imitation systems operating in prediction markets. The system is designed to isolate model capabilities, enforce experimental repeatability through instruction locking, and measure holistic reasoning quality including drift patterns, calibration, and risk assessment. The full pipeline consists of four core stages: instruction locking and prompt specification, baseline construction, agent deployment, and market-linked execution with drift tracking.

3.1. Instruction Locking and Prompt Specification Layer. The evaluation process begins with a controlled prompt specification interface that implements instruction locking, a mechanism that ensures prompt templates are versioned, immutable, and reproducible. Researchers define a probability-query template that includes:

- a binary forecasting target which is either a yes or a no,
- a time horizon that reflects cycling and tuning (e.g. 466 cycle representing api call for each model),
- constraints on justification length, reasoning style, or tool usage,
- token budget limits (addressing token constraint awareness),
- the expected probability format (e.g., scalar 0–1), and
- metadata for versioning and reproducibility.

Once locked, prompt templates cannot be modified, preventing prompt engineering from masking model limitations or introducing test set contamination. This layer ensures that each model receives an identical and rigorously defined forecasting instruction, enabling fair comparison across models and replication across research teams.

TruthTensor stores each prompt configuration as a unique evaluation contract with a cryptographic hash, enabling verification that evaluations use the exact same prompts. This instruction locking mechanism distinguishes TruthTensor from benchmarks that allow prompt engineering, ensuring that evaluation focuses on model capabilities rather than prompt optimization.

3.2. Baseline Construction Layer. In this study, the baseline is defined exclusively as the market baseline, constructed directly from live prediction market prices. All baseline information used throughout the paper is derived solely from the market signals summarized in Table 2, which therefore func-