

# Look-Ahead-Bench: a Standardized Benchmark of Look-ahead Bias in Point-in-Time LLMs for Finance

Mostapha Benhenda <sup>\*</sup>

## Abstract

We introduce Look-Ahead-Bench, a standardized benchmark measuring look-ahead bias in Point-in-Time (PiT) Large Language Models (LLMs) within realistic and practical financial workflows. Unlike most existing approaches that primarily test inner lookahead knowledge via Q&A, our benchmark evaluates model behavior in practical scenarios. To distinguish genuine predictive capability from memorization-based performance, we analyze performance decay across temporally distinct market regimes, incorporating several quantitative baselines to establish performance thresholds.

We evaluate prominent open-source LLMs—Llama 3.1 (8B and 70B) and DeepSeek 3.2—against a family of Point-in-Time LLMs (Pitinf-Small, Pitinf-Medium, and frontier-level model Pitinf-Large) from PiT-Inference. Results reveal significant lookahead bias in standard LLMs, as measured with alpha decay, unlike Pitinf models, which demonstrate improved generalization and reasoning abilities as they scale in size. This work establishes a foundation for the standardized evaluation of temporal bias in financial LLMs and provides a practical framework for identifying models suitable for real-world deployment. Code is available on GitHub: <https://github.com/benstaf/lookaheadbench>.

## 1. Introduction

Large Language Models are revolutionizing quantitative finance. FinGPT (Yang et al., 2023), BloombergGPT (Wu et al., 2023), FinMem (Yu et al., 2023), FinRL-DeepSeek (Benhenda, 2025), and TradingAgents (Xiao et al., 2025), along with multi-agent architectures like Hedge-Agents (Li et al., 2025a) and the AI

<sup>\*</sup> Correspondence to: Mostapha Benhenda <[mostaphabenhenda@gmail.com](mailto:mostaphabenhenda@gmail.com)>.

Hedge Fund (Singh, 2025), have demonstrated remarkable capabilities in processing financial data, generating trading signals, and executing complex investment strategies. However, beneath these promising results lies a fundamental methodological challenge that threatens the validity of LLM-based financial applications: look-ahead bias.

Look-ahead bias occurs when models access information that would not have been available at the time of prediction, creating artificially inflated performance metrics that evaporate in real-world deployment. Consequently, back-tested returns often collapse once the model’s knowledge window ends and trading gets into genuinely unknown territory (Li et al., 2025b).

### 1.1. The Challenge of Temporal Bias in Financial LLMs

LLMs are pre-trained on web-scale corpora containing extensive financial data, including post-hoc explanations of market events, historical price movements, and retrospective analyses. When an LLM encounters a prompt about “NVIDIA’s performance in 2023,” it may have been trained on text explicitly stating “NVIDIA surged 190% in 2023 on AI boom” (Li et al., 2025b). In this case, the model does not learn predictive relationships; it memorizes outcomes and recites them during evaluation.

This pre-training contamination creates systematic biases that are particularly pernicious in finance, where temporal causality is paramount. Several studies have documented various manifestations of this issue:

- **Training Data Leakage:** Models recall specific stock prices, earnings figures, and market events directly from their training corpus (Lopez-Lira et al., 2025b).
- **Entity Memorization:** Even with masked identifiers, models can recognize companies from contextual clues and retrieve associated historical performance (Sarkar & Vafa, 2024).
- **Temporal Reasoning Failures:** Models struggle to distinguish between information available at different time points, leading to logically impossible predictions (Yan & Tang, 2026; Gao et al., 2025).

## 1.2. Limitations of Current Evaluation Approaches

Existing approaches to detecting lookahead bias have focused primarily on testing inner knowledge through question-answering tasks, or evaluating the memorization of specific facts. While valuable, these evaluation methods may not reflect the practical consequences of bias in real-world financial settings.

The pragmatic test of a financial model’s capability does not lie in its ability (or lack thereof) to recall historical patterns from its training data, but in its practical capacity to generalize to genuinely novel market conditions.

## 1.3. Our Contributions

We introduce Look-Ahead-Bench, a standardized benchmark that addresses these limitations through three key contributions:

- 1. Practical Trading Workflow Integration:** Unlike knowledge-based benchmarks, our evaluation uses realistic agentic trading systems that make actual portfolio decisions. We utilize the AI Hedge Fund framework (Singh, 2025), a widely adopted open-source project (+45k stars on GitHub), ensuring evaluation moves beyond toy problems to assess real-world deployment viability.
- 2. Point-In-Time LLMs Evaluation:** We explicitly evaluate Pitinf models—commercial LLMs specifically designed to effectively remove lookahead bias—alongside standard foundation models.
- 3. Alpha Decay Metric:** We introduce a rigorous metric for measuring bias by quantifying the performance drop between in-sample (training window) and out-of-sample (future) periods.

Our results demonstrate that standard LLMs exhibit significant lookahead bias, with alpha decay exceeding -15 percentage points between periods, while Pitinf models maintain stable performance. This work provides a diagnostic tool for identifying bias in financial LLMs.

## 2. Related Work

### 2.1. Lookahead Bias in Financial LLMs

The recognition of lookahead bias as a critical issue in financial LLM applications has grown rapidly in recent years. Glasserman & Lin (2023) provided one of the earliest systematic assessments, investigating both lookahead bias and distraction effects in GPT-based sentiment analysis for stock return prediction. Their work revealed that anonymizing company identifiers could actually improve

performance by reducing distraction effects, while highlighting the challenge of disentangling genuine sentiment analysis from memorized associations.

Levy (2025) emphasized implications for financial applications and proposed mitigation strategies including careful data cutoff management and entity embedding neutralization. Gao et al. (2025) developed a statistical test for detecting lookahead bias using Lookahead Propensity (LAP), a measure based on membership inference attack techniques. Their approach correlates model familiarity with training data (estimated through token probability analysis) with forecast accuracy. While innovative, this method focuses on individual prediction tasks rather than end-to-end trading system evaluation.

Other contributions have expanded both the empirical and methodological study of lookahead bias in financial language models, proposing standardized diagnostics, benchmark-style evaluations, and controlled temporal stress tests that reveal performance inflation caused by future data contamination in forecasting and trading settings (Noguer i Alonso, 2024; Lopez-Lira et al., 2025a).

### 2.2. Information Leakage and Memorization

The broader issue of information leakage in LLMs has received extensive attention. Li et al. (2025b) introduced FinLeak-Bench and the FactFin framework to systematically quantify information leakage across four dimensions, demonstrating that most published LLM-based agents fail to beat baselines once their knowledge cutoff is passed.

Lopez-Lira et al. (2025b) documented the “memorization problem” through extensive testing of GPT-4o’s recall capabilities, showing that models can recall exact S&P 500 closing prices with less than 1% error for dates within their training window, while errors “explode” for post-cutoff dates. Carlini et al. (2021; 2022) established theoretical and empirical foundations for understanding memorization in large language models, demonstrating that training data can be extracted from models through carefully designed queries.

Complementary studies further show that pretraining contamination and implicit temporal leakage can persist even without explicit date cues, and that leakage effects scale with model size unless explicitly mitigated (Kong et al., 2025; Drinkall et al., 2024).

### 2.3. Point-in-Time and Temporally Aware Models

The development of Point-in-Time (PiT) models represents a direct response to lookahead bias concerns. Models like Time Machine GPT (Drinkall et al., 2024), ChronoGPT (He et al., 2025) and DatedGPT (Yan & Tang, 2026) introduce time-aware frameworks that train models strictly on

pre-cutoff data to ensure temporal integrity. Their approach demonstrates that explicit temporal constraints can prevent future information leakage. Merchant & Levy (2025) proposed logit adjustment from specialized auxiliary models, allowing both verbatim and semantic knowledge to be removed without retraining large base models.

There are also proprietary Point-In-Time LLMs designed for backtesting, such as those from PiT-Inference (PiT-Inference, 2026). These are available in three sizes: Small size (<10B parameters) for low-latency, Medium size (<100B parameters), and Large size (>500B parameters) for frontier-grade reasoning tasks.

## 2.4. Agentic Trading Systems

The application of LLMs to agentic trading has produced numerous innovative systems. Yu et al. (2023; 2024) developed FinMem and FinCon, demonstrating sophisticated memory and multi-agent capabilities. Zhang et al. (2024b) created a multimodal foundation agent with tool augmentation, while Yang et al. (2024) introduced FinRobot for equity research and valuation.

The AI Hedge Fund open-source project (Singh, 2025) provides LLM-based trading agents and serves as the foundation for our benchmark implementation, due to its significant impact (+45k stars on GitHub) on the community.

## 3. Methodology

### 3.1. Dual-Period Design

Our evaluation compares model performance across two carefully selected periods to test models with knowledge cutoffs through 2023-2024, while maintaining similar market characteristics for fair comparison, like in FinLeak-Bench (Li et al., 2025b):

**Period P1 (In-Sample):** April 1, 2021 – September 30, 2021

- Duration: 6 months
- Buy-and-Hold Return: +25.32%
- Purpose: Establish baseline performance within the potential training window of Llama 3.1 (8B, 70B) and DeepSeek 3.2.

**Period P2 (Out-of-Sample):** July 1, 2024 – December 31, 2024

- Duration: 6 months
- Market Regime: AI-driven rally with mixed sentiment.

- Buy-and-Hold Return: +24.75% (similar to P1).
- Purpose: Test generalization to post-cutoff periods for Llama 3.1 (8B, 70B) and DeepSeek 3.2.

### 3.2. Portfolio and Trading Universe

Our benchmark employs a focused trading universe consisting of five large-cap technology stocks: AAPL (Apple Inc.), MSFT (Microsoft Corporation), GOOGL (Alphabet Inc.), NVDA (NVIDIA Corporation), and TSLA (Tesla Inc.).

### 3.3. Evaluation Metrics

#### 3.3.1. ALPHA CALCULATION

For each model and period, we calculate alpha as:

$$\alpha = R_{LLM} - R_{Buy\&Hold} \quad (1)$$

This measures the excess return generated by the LLM strategy relative to a passive buy-and-hold benchmark.

#### 3.3.2. ALPHA DECAY ANALYSIS

The core bias indicator is alpha decay between periods:

$$\alpha_{Decay} = \alpha_{P2} - \alpha_{P1} \quad (2)$$

Negative alpha decay indicates that the model's relative performance deteriorated in the out-of-sample period, suggesting potential lookahead bias in the in-sample results.

### 3.4. Quantitative Baselines

We benchmark the AI agents against a suite of traditional quantitative strategies. These strategies span passive, systematic, trend-following, and contrarian approaches, providing a comprehensive reference for evaluating relative performance and robustness across market regimes. All strategies are executed with the same initial capital, fractional shares allowed, and monthly rebalancing (except where noted).

#### 3.4.1. BUY & HOLD (PASSIVE)

The simplest baseline allocates capital equally across all tickers at the start of the period and holds the positions without any rebalancing. This represents a passive, long-only exposure to the selected universe and serves as the reference for computing alpha (excess return over this benchmark).

#### 3.4.2. EQUAL-WEIGHT MONTHLY REBALANCE (SYSTEMATIC)

This strategy maintains equal dollar allocation to each ticker by rebalancing at the start of each month. It captures