Assessing the Benefits of Optimized Agentic AI Systems for Asset Pricing*

Ralph S.J. Koijen Bradford Levy

March 26, 2026

Preliminary and incomplete

Abstract

Evaluating optimized AI systems for asset pricing is fundamentally difficult for two reasons. First, models are trained on all data, implying that any backtest or analysis using historical data suffers from look-ahead bias. In addition, markets are reflexive — as investors adopt AI, prices adjust — which may erode the very patterns the AI system was trained to exploit. We introduce a real-time, out-of-sample benchmark designed to sidestep both problems. The benchmark measures how well AI systems can explain contemporaneous stock returns around earnings announcements using only information available at announcement time, including the announcement text itself. Applying this benchmark to a range of agentic AI systems — which extract structured signals from earnings call transcripts and optimize over those signals — we find that the best-optimized systems more than double the explained variation in returns relative to standard benchmarks (R^2 increasing from 8% to close to 20%). We also discuss the efficiency gains of AI-based optimization compared to traditional machine learning methods. Beyond predictive performance, these systems generate narrative explanations for price movements that yield testable implications for asset pricing theory. We release an SDK and launch an open competition inviting researchers to improve on our results. Saturating this benchmark would represent fundamental progress in understanding how capital markets process firm-level information.

*Koijen: University of Chicago Booth School of Business, NBER, and CEPR. Email: ralph.koijen@chicagobooth.edu. Levy: University of Chicago Booth School of Business. Email: bradford.levy@ChicagoBooth.edu. We thank participants at The Future of AI in Asset Management Conference at Chicago Booth and the University of Utah for helpful feedback and suggestions. For financial support, Koijen thanks the Center for Research in Security Prices at the University of Chicago and the Fama Research Fund at the University of Chicago Booth School of Business.

1 Introduction

Modern AI systems have quickly become more capable across a wide range of domains. This naturally raises the question whether these models can also be used to analyze financial markets and explain asset prices. The apparent random movements in asset prices has long puzzled financial economists. The challenge is perhaps most salient around earnings releases. During a short window, stock prices display large moves which have proved difficult to explain, e.g., earnings surprises (measured in various ways) explain up to about 5% of the variation in returns (see, e.g., Chiang et al., 2019; Hand et al., 2022). It is therefore natural to ask how well modern AI models can explain this variation and reduce the "dark matter" of stock price fluctuations.

Using AI systems to explain stock price movements faces at least two challenges. First, it is well known that large language models (LLMs) suffer from look-ahead bias as they have been trained on vast amounts of data, including data on returns. While attempts have been made to mitigate the impact of look-ahead bias (e.g., Glasserman and Lin, 2023; Engelberg et al., 2025; Merchant and Levy, 2025), this remains a challenging task. Look-ahead bias is unlikely to be fully addressed unless the models are trained with hard knowledge cutoffs. Moreover, a growing literature highlights the benefits of *optimizing* AI systems, i.e., by optimizing prompts, context, or updating the model's parameters via supervised finetuning (SFT) or reinforcement learning (RL). If the AI model has knowledge of the announcement return, optimization is more likely to recover the information from the model and increase look-ahead bias (Hu et al., 2024). While training generative models with knowledge cutoffs is feasible (Drinkall et al., 2024; Levy, 2024; He et al., 2025), those models significantly underperform frontier AI models and can at best provide a lower bound on the ability of optimized AI systems to explain asset prices.

Second, while natural language processing has long been used by investors, modern AI models have significantly increased the set of tools available to market participants. As a result, in equilibrium, it is plausible that the relationship between text and returns evolves as the model capabilities expand and investors find new ways to deploy AI. As we are primarily interested in how well optimized AI systems explain current prices, conclusions based on historical data may not carry over to the current market environment.

To address these challenges, we make three contributions. First, we introduce a new real-time benchmark that evaluates optimized AI systems out-of-sample (OOS) in real time to explain stock price movements. The inputs to the model, which include the context and prompts that the AI model uses to make its forecasts (as well as the model parameters), are determined prior to the earnings announcement. We then feed the model the earnings call

information (i.e., the earnings numbers as well as the earnings call transcript) and let the model predict the announcement return.

Our second contribution is to consider a broad range of methods and models to assess their performance during the last quarter of 2025 on our new benchmark. Future versions of this paper will extend these results to subsequent earnings quarters. We provide three main empirical findings. First, frontier large language models (LLMs) can explain up to 15% of the variation in returns based on the earnings call transcript and earnings surprise information alone, which is significantly higher than the prior literature that explains up to 8% of the return variation (Hand et al., 2022). Second, this average number masks a significant amount of inter-industry variation. While there is little heterogeneity in the ability of earnings surprises to explain returns across industries, optimized AI systems perform much better in some industries than for others. We also find that no one model dominates; one model may outperform for financial firms, while another model outperforms in utilities. Third, there are non-trivial benefits from optimizing model prompts and context. We focus our attention on exploring two state-of-the-art, gradient-free methods of prompt optimization: Genetic-Pareto (GEPA) search (Agrawal et al., 2026) and Agentic Context Engineering (ACE) (Zhang et al., 2026). Both methods follow an iterative process on a training sample to develop a prompt or additional context that guides the model's reasoning to predict the contemporaneous return. We then evaluate the optimized prompts and context out-of-sample and find that it raises the fraction of explained variation to close to 20%.

Our third contribution is to show how these systems generate narrative explanations for price movements that yield testable implications for asset pricing theory beyond explaining return variation. Put simply, leading methods engineer the context, which results in a rule book that the model follows in making predictions. We show how to map the rules to factors that need to be incorporated in asset pricing theories. Our new benchmark provides a natural testing ground for rational and behavioral theories alike, and for how information ultimately gets incorporated into asset prices.

These results lead to three important insights. First, optimized AI systems have an important role to play in guiding future empirical and theoretical research in asset pricing by generating interpretable explanations of asset price movements—notably these explanations hold out-of-sample. Second, while we apply recently-developed methods to frontier LLMs and find significant gains already, we think that we just scratched the surface and that further improvements are within reach. For instance, we have not used information (i) from earlier earnings calls (e.g., some CEOs may generally be more optimistic than others), (ii) from peers' earnings calls or (iii) additional information that is typically available right before a call to pinpoint what specific items analysts are looking for in the upcoming call. Moreover, model

capabilities are expanding rapidly as are the prompt and context optimization methods. This makes for a clear and exciting journey to discovering what moves asset prices. Third, the benchmark naturally refreshes itself as the topics and themes that drive markets (e.g., from the great financial crisis to the European sovereign debt crisis or the COVID-19 pandemic) change and may change the connection between text and returns. In the ultimate case where the benchmark gets saturated, as has been the case with many other AI benchmarks, we have made great strides in understanding firm-level asset prices, at least around key event dates.

To simplify and coordinate the effort to discover how optimized AI system can be used to explain asset prices, we release an SDK and launch an open competition inviting researchers to improve on our results.

2 Benchmarking Agentic AI Systems

Evaluating the extent to which AI systems can explain asset prices requires a benchmark that is both methodologically rigorous and economically meaningful, e.g., free of look-ahead bias and focused on a task that is relevant. In this section, we outline the desirable features of such a benchmark, formally define its structure, and explain our approach to fully out-of-sample evaluation.

2.1 Desirable Features of a Benchmark

The application of LLMs, agentic tooling, and more broadly AI models to a variety of domains has shown rapid progress (see, e.g., Jumper et al., 2021). This progress is increasingly stark when optimization methods are applied (Ouyang et al., 2022; Khattab et al., 2024; Agrawal et al., 2026). Although a variety of domains have seen rapid progress, it is unclear whether we should expect similar performance when applying these methods to tasks involving markets.

For example, the fact that long time-series of content are used to train these models implies that the models will exhibit look-ahead bias for many in-sample tasks. Layering optimization on top of such models/systems is likely to exacerbate the bias. Even if one ignores look-ahead bias, the reflexivity of markets suggests that the benefits of the technology may not hold out of sample, i.e., in the context of markets the value of information or technology today generally differs from its value tomorrow. While these are just two examples of why utility on market-based tasks may be limited, they suggest that empirical research involving AI agents needs to adapt.

Acknowledging the differences and challenges associated with empirical research involing

AI, we first outline some desirable features of *any* benchmark:

1. Construct Validity - the benchmark measures what it claims to measure
2. Contamination Resistance - "answers" should not be readily available due to leakage at training or inference time
3. Ecological Validity - alignment with real-world task performance
4. Discriminative Power - systems with different capabilities achieve meaningfully different scores
5. Reproducibility & Transparency - benchmark is fully and openly specified

Guided by these features, we design a benchmark that examines the extent to which systems can explain the consequences of information releases on asset prices. As we will show in the next section, when such a benchmark is evaluated in real-time it naturally satisfies many—if not all—of these features.

2.2 Benchmark Definition

We formalize the benchmark within the context of a standard event study framework. Consider a stylized factor model where the returns of asset i around time t over a horizon h are given by:

$$r_{i,[t,t+h]} = \boldsymbol{\beta}_{i,t}^{\top} \mathbf{F}_{[t,t+h]} + \epsilon_{i,[t,t+h]}. \tag{1}$$

The benchmark isolates cases where content $\omega_{i,t}$ relevant to asset i is released at time t. Using an earnings announcement as an example, asset i represents a firm's equity, and $\omega_{i,t}$ represents the corpus of text and audio released during the event. The idiosyncratic return component can thus be decomposed as:

$$\epsilon_{i,[t,t+h]} = g(\omega_{i,t}, \Omega_{t-}) + \nu_{i,[t,t+h]} \tag{2}$$

where Ω_{t-} is the observable information set strictly prior to the release, $g(\cdot)$ is the unobservable, market-wide information pricing function, and $\nu_{i,[t,t+h]}$ is orthogonal noise.

Given this setup, the objective is to measure how well an AI system can approximate the true pricing function $g(\cdot)$. Formally, let $\mathcal{E} = \{(i_k, t_k)\}_{k=1}^{N}$ denote the benchmark evaluation sample of N discrete information events. A valid submission to the benchmark consists of a set of scalar predictions, $\{\hat{g}(\omega_{i,t}, \Omega_{t-})\}_{(i,t)\in\mathcal{E}}$, generated by the AI system.

Because both $g(\cdot)$ and ν are unobservable, we evaluate the performance of a submission based on its explanatory power in a cross-sectional regression:

$$\epsilon_{i,[t,t+h]} = \alpha + \gamma \hat{g}(\omega_{i,t}, \Omega_{t-}) + \boldsymbol{\theta}^\top \mathbf{X}_{i,t} + u_{i,[t,t+h]}, \tag{3}$$

where $\epsilon_{i,[t,t+h]}$ is the observed residual from the factor model, $\mathbf{X}_{i,t}$ is a vector of known, quantitative determinants of the idiosyncratic market reaction (such as the standardized unexpected earnings, or SUE), and $u_{i,[t,t+h]}$ is the regression residual.

The primary performance statistic is the incremental explanatory power, or ΔR^2, generated by the AI's predictions over a baseline model, e.g., one that includes only the quantitative controls $\mathbf{X}_{i,t}$ or the current "best" system. This relates the AI's edge directly to the inherent noise in the system. Assuming g and ν are orthogonal, the theoretical maximum R^2 a perfectly omniscient system could achieve is bounded by:

$$\overline{R}^2 = \frac{Var(g)}{Var(g) + Var(\nu)}. \tag{4}$$

This benchmark design is general enough to be of broad interest to both academics and quantitative practitioners. Because it closely mirrors research designs used to study macroeconomic announcements and corporate events, it is highly extensible. While our initial implementation focuses on corporate earnings announcements, we plan to expand to additional event types to assess AI capabilities across a wider variety of economic contexts.

2.3 A First Implementation: Earnings Announcements

The benchmark's initial iteration focuses on corporate earnings announcements (EAs) for three primary reasons. First, a rich literature spanning more than five decades has sought to explain the returns surrounding earnings announcements. Progress from the earliest iterations to today has been an increase in explanatory power from roughly 2% to 8%, with much of this coming from more robust measurement (Chiang et al., 2019) or the incorporation of numerous surprises along dimensions other than just earnings (Hand et al., 2022). This baseline suggests that the benchmark will either demonstrate significant discriminative power across AI models, or reveal that asset prices around these events are overwhelmingly driven by unobservable noise.

Second, EAs are information-rich events involving content across a variety of modalities: text, audio, and sometimes visual presentations. As such, the market reaction is likely a function of both quantitative and qualitative information. This multimodal synthesis is precisely where LLMs and AI agents are expected to add significant value beyond what historical,

lexicon-based methods have achieved. Third, there are more than 1,000 EAs every quarter. Even assuming moderate effect sizes, this yields a sufficiently large sample to provide statistically powerful, real-time tests of performance differences across systems. Consequently, we can be highly confident that the benchmark is robust against data contamination.

2.3.1 Event Date and Return Measurement

The core of the benchmark relies on isolating the market reaction to the specific content released during the EA. We measure the abnormal return in a narrow one-day window strictly aligned with when the information becomes actionable for market participants. For an earnings event occurring at time t_c on calendar date d, the effective trading event date t is defined as:

$$t = \begin{cases} d & \text{if } t_c \leq 16:00 \text{ ET and } d \text{ is a trading day} \\ \tau(d) & \text{if } t_c > 16:00 \text{ ET or } d \text{ is a non-trading day} \end{cases} \tag{5}$$

where $\tau(d)$ denotes the next available trading day following calendar date d.

Abnormal returns are measured using a standard market-adjusted return. For stock i on trading day t, we define the abnormal return as:

$$r_{i,t}^{abn} = r_{i,t} - r_{m,t} \tag{6}$$

where $r_{i,t}$ is the total stock return (inclusive of dividends) and $r_{m,t}$ is the return on the broader market. This market-adjusted return serves as our empirical proxy for the idiosyncratic residual $\epsilon_{i,[t,t+1]}$ defined in Equation (1). To enable real-time, out-of-sample measurement, we proxy the market return using the Vanguard Total Stock Market ETF (VTI). Both returns are calculated using close-to-close prices from $t-1$ to t.

2.3.2 Earnings Surprise Baseline

A critical question is how well existing approaches explain announcement returns. The standard measure in the literature is the earnings surprise: the difference between reported earnings and analyst consensus expectations formed strictly prior to the announcement. Recalling our general control vector $\mathbf{X}_{i,t}$, we define its primary component, the standardized earnings surprise $s_{i,t}$, for an announcement at time t as:

$$s_{i,t} = \frac{e_{i,t} - \mathbb{E}[e_{i,t}|\Omega_{t-}]}{p_{i,t-}} \tag{7}$$

where $e_{i,t}$ is the reported earnings per share, $\mathbb{E}[e_{i,t}|\Omega_{t-}]$ is the consensus (mean) analyst forecast formed prior to the release, and $p_{i,t-}$ is the pre-announcement stock price.

2.3.3 Cross-Sectional Percentile Rank

To ensure comparability across different time periods and varying market regimes, we convert both the abnormal return and earnings surprise into cross-sectional percentile ranks. Within each distinct evaluation period q (e.g., a calendar quarter), we compute the percentile rank as:

$$PRank(x_{i,t}) = \frac{\text{rank}(x_{i,t})}{N_q} \tag{8}$$

where $\text{rank}(x_{i,t})$ assigns a value of 1 to the most negative value of x and N_q to the most positive, and N_q is the total number of valid observations in evaluation period q. Henceforth, when we say abnormal return or earnings surprise, we are referring to the percentile ranked version of the variable, i.e., $PRank(r_{i,t}^{abn})$ and $PRank(s_{i,t+1})$, respectively, rather than the raw value.

This transformation addresses several empirical considerations. First, the cross-sectional distribution of returns and earnings surprises varies across market environments, e.g., returns are heavily dispersed during volatile regimes and compressed during calm periods. Second, percentile ranks are robust to extreme outliers. Third, this rank-based metric aligns with how investors evaluate relative portfolio performance.

2.3.4 Real-Time Out-of-Sample Evaluation

Benchmark submissions and evaluation follow a strict temporal protocal to ensure that they are truly out-of-sample and free from look-ahead bias. Specifically, a calendar of expected earnings announcements (EAs) is maintained to facilitate the collection of a fair ex ante infromation set, i.e., Ω_{t-}, prior to an event. Once an EA occurs, we collect the transcript from the event and provide it to candidate systems–either in its complete form or summarized.

Having passed the transcript to candidate systems, we then accept only *timely* submissions for evaluation. This means that if preparing a submission takes more than a pre-specified amount of time, the submission is rejected due to the possibility that it is contaminated. Finally, only systems that provide valid submissions for more than 90% of events will be considered for evaluation.

We want to highlight that this strict temporal protocol applies at the observation level. Thus, information learned earlier in a given evalutation period can—*and likely should*—be used when making predictions later in the period. Along these lines, our initial evaluations

use the first half of a quarter to optimize Agentic AI which are then applied to the second half of the quarter. We find that this leads to a significant increase in performance. Given these results, while we choose to split the quarter evenly, there is no reason that this optimization and learning cannot be applied online in real-time.

3 Data and Results

3.1 Data Sources and Sample

Our initial implementation of the benchmark and results cover the fourth calendar quarter of 2025, i.e., earnings announcements from October through December. We choose to present results for this quarter because it is the first full quarter for which our production version of the benchmark was "frozen" and operational. This enables us to collect data, generate predictions, and evaluate results strictly out of sample.

The data for the benchmark is sourced from three real-time providers. First, we collect expected earnings events from Benzinga, which identifies upcoming announcement dates, analyst consensus estimates $\mathbb{E}[e_{i,t}|\Omega_{t-}]$, and reported actuals $e_{i,t}$. Conference call transcripts $\omega_{i,t}$—the primary textual input to the AI systems—come from S&P Capital IQ. Pricing data, including close-to-close returns for individual stocks and for the VTI market proxy, are obtained from Massive (formerly Polygon.io).

Outputs from AI systems are collected as soon as conference call transcripts become available. Specifically, following Section 2, we pass each transcript to the candidate AI systems and require predictions be returned within a fixed time window. Because the benchmark is evaluated in real time, there is effectively no possibility that announcement returns or post-event commentary contaminate the information set provided to any model.[1]

The final sample comprises 1,849 earnings events spanning all 12 Fama–French industries and six reporting weeks within the quarter. Table 1 provides summary statistics for the sample. The median firm in our sample has a stock price of $29.66 and is covered by six analysts, while the mean coverage is 8.30 analysts. In robustness tests reported in the appendix, we verify that the choice of real-time data provider does not materially affect our results: substituting I/B/E/S for analyst estimates and actuals and Capital IQ for pricing data yields coefficient estimates and R^2 values that are not significantly different at the 1%

[1]For Opus 4.5, we do process transcripts with a short delay relative to the other models. While this lag could in principle introduce look-ahead bias if the model's training corpus contained post-announcement information, Anthropic has confirmed that the training data cutoff for Opus 4.5 is August 2025, well before our sample period begins. Further, we do not allow Opus to search the web or access external sources of information. Therefore the risk of leakage is quite small.

level—although the sample size changes slightly due to differences in coverage.

3.2 Sample Characteristics

As a first step, we analyze the composition of our sample across industry and reporting time within the quarter (see Panel A of Table 1). Three industries account for more than half of the sample: Business Equipment (361 observations), Health (319), and Finance (300). Panel A also reveals substantial heterogeneity in the timing of announcements across industries. Financial firms report markedly earlier than the rest of the market, contributing 36 of the 49 observations in week 2 and 89 of the 222 in week 3. This early clustering implies that any model seeking to explain returns in the first few weeks of earnings season must primarily understand the financial sector. By contrast, Health firms are concentrated later in the quarter: 254 of their 319 announcements (80%) fall in weeks 5 and 6. Business Equipment firms follow a similar late-reporting pattern, with 198 of 361 events occurring in week 5 alone. The busiest week overall is week 5, which contains 850 observations—46% of the entire sample—reflecting the well-documented tendency for the majority of firms to cluster their announcements in the middle of the reporting window.

Panel B reports week-level summary statistics for abnormal returns ($r_{i,t}^{abn}$), earnings surprises ($s_{i,t}$), and the R^2 from regressing percentile-ranked abnormal returns on percentile-ranked surprises. Two patterns stand out. First, the earliest announcers deliver the most positive news. Week 1 firms have a mean abnormal return of 6% and a mean surprise of 0.18% of price, both substantially above the full-sample averages of -1% and -0.05%, respectively. In contrast, the latest reporters deliver the most negative surprises: the mean surprise in week 6 is -0.58% of price. Collectively, these patterns are consistent with the voluntary disclosure literature which predicts that managers with favorable news accelerate their announcements, while those with unfavorable results delay.

Second, the cross-sectional dispersion in earnings surprises increases dramatically across the quarter, rising from a standard deviation of 0.11 in week 1 to 6.59 in week 6. The R^2 from the surprise-return regression peaks at 0.16 in week 3—the week dominated by financial firms—and settles to 0.05 for the full sample. Given the heterogeneity across weeks, we do not place strong weight on the week-by-week variation in R^2; instead, the full-sample R^2 of 0.05 serves as our primary benchmark for evaluating AI models.

Panel C reports cross-sectional descriptive statistics for key firm characteristics. The typical firm in our sample is covered by six analysts (median), though coverage varies widely, with a standard deviation of 6.20 and an interquartile range of 3 to 12 analysts. Dispersion in analyst forecasts—a proxy for ex ante uncertainty about earnings—has a median of 0.04

and a mean of 0.09, indicating that a small number of firms with highly uncertain earnings pull the mean well above the median. Earnings surprises exhibit a similar pattern: the median surprise is positive at 0.12% of price, suggesting that the typical firm modestly beats consensus expectations, while the mean is −0.05%, dragged down by a left tail of large negative surprises concentrated among late-reporting firms.

3.3 Initial Results

Our results from the first iteration of the benchmark are focused on evaluating three frontier LLMs: Gemini 2.5 Flash-Lite, GPT-5 nano, and Opus 4.5. We begin by assessing their baseline ability to explain contemporaneous stock returns around earnings announcements, i.e., how good are the models when given comparatively simple prompts. We then examine the effects of transcript summarization, event timing, and industry heterogeneity. Throughout, we report R^2 from regressions of abnormal returns on model predictions and the earnings surprise, with 99% bootstrap confidence intervals in brackets.

3.3.1 LLM Predictions Using Full Transcripts

Table 2 presents our baseline results using full earnings call transcripts. Column (1) establishes the benchmark: regressing percentile-ranked abnormal returns on the percentile-ranked earnings surprise alone yields an R^2 of 5.0%, consistent with the upper end of estimates in the existing literature. The remaining columns evaluate the three LLMs, first in isolation and then jointly with the surprise.

Even without any optimization, LLM predictions substantially increase explanatory power. Opus 4.5 achieves the highest standalone R^2 of 10.9% [0.08, 0.15], more than doubling the surprise-only baseline. Gemini 2.5 Flash-Lite and GPT-5 nano perform similarly to each other, with standalone R^2 values of 7.5% and 7.1%, respectively. Notably, GPT-5 nano achieves its R^2 with a much larger coefficient on its prediction (0.726 versus 0.374 for Gemini), suggesting that the two models use different scales but extract comparable amounts of information from the transcript.

A key finding is that LLM predictions and the earnings surprise are complementary. When we include both the LLM prediction and the surprise in the regression (columns 5–7), the coefficient on the surprise remains large and highly significant for all three models, declining only modestly from 0.219 to approximately 0.18–0.19. This indicates that the models capture information in the transcript that goes beyond simple earnings news—likely including guidance, management tone, and the question-and-answer discussion—while the quantitative surprise retains independent explanatory power. The combined specification

using Opus 4.5 yields an R^2 of 14.3% [0.10, 0.18], nearly tripling the surprise-only benchmark.

Figure 1 places these results in historical context. The literature has progressed from an R^2 of roughly 2% in Lev (1989) to approximately 4% in Chiang et al. (2019) and 7.5% in Hand et al. (2022), where the latter result required incorporating surprises along 33 dimensions beyond earnings. Our best unoptimized model achieves 14.3% using only the earnings surprise and a single LLM prediction derived from the transcript. This represents a significant advance in the ability to explain the "dark matter" of stock price movements around earnings announcements.

3.3.2 Summarized Transcripts

Table 3 repeats the analysis using transcript summaries generated by Opus 4.5 in place of full transcripts. These summaries are approximately 28 times shorter than the original documents, which has important implications for both cost and the feasibility of prompt optimization, as optimization methods require many evaluations of the objective function.

The results show that summaries preserve most of the information relevant for explaining returns. Opus 4.5 applied to its own summaries achieves a standalone R^2 of 9.3% [0.06, 0.13], compared to 10.9% with the full transcript. When combined with the surprise, the summary-based specification yields an R^2 of 13.0% [0.09, 0.18] versus 14.3% with the full transcript—a decline of only 1.3 percentage points, or roughly 9% in relative terms. The bootstrap confidence intervals for the two specifications overlap substantially, suggesting that the difference is not statistically significant at the 1% level.

The loss from summarization is more pronounced for the other two models. GPT-5 nano's standalone R^2 drops from 7.1% to 3.0% when using summaries, indicating that this model is more reliant on details present in the full transcript that are lost during summarization. Gemini 2.5 Flash-Lite exhibits a more moderate decline, from 7.5% to 6.5%. These differences across models suggest that the quality of the summarization step interacts with the downstream prediction model, a question we leave for future investigation. For our optimization experiments below, we proceed with summaries given their substantially lower cost and the modest loss in explanatory power for the best-performing model.

3.3.3 Early versus Late Announcements

Our optimization methods require a training set to learn from and a separate evaluation set to test on. We implement this by splitting the quarter at the median announcement date, yielding an "Early" subsample of 1,032 observations and a "Late" subsample of 817 observations. The early period is used for training, and the late period serves as the out-of-

sample evaluation set.

Table 4 reports results from this split. A natural concern is that model performance might differ systematically across the two subperiods, which would complicate interpretation of the optimization results. Reassuringly, we find that performance is similar or modestly better in the late period. For Opus 4.5 with summaries and surprise, the R^2 is 11.8% in the early period and 14.1% in the late period. Gemini 2.5 Flash-Lite shows a comparable pattern (9.3% early, 10.5% late), as does GPT-5 nano (6.6% early, 8.5% late). The slightly higher R^2 in the late period may reflect the compositional shift toward industries where LLMs perform particularly well, or simply sampling variation. The key takeaway is that there is no evidence that the evaluation period is systematically harder than the training period, which would bias against finding optimization gains.

3.4 Optimization

The results presented thus far are based on comparatively simple and basic prompts, e.g., "Your task is X, be mindful of Y and Z, the typical steps to performing X are 1, 2, and 3. Now, go do it." Recent work in the CS literature has shown that significant increases in performance can be achieved through minor changes to the prompts, e.g., encouraging the model to think before answering (Wei et al., 2022), or gradient-free optimization of prompts, e.g., Genetic-Pareto Prompt Optimization (Agrawal et al., 2026) and Agentic Context Engineering (ACE) (Zhang et al., 2026). In this section, we explore the value of model context optimization within the setting of the benchmark.

3.4.1 Setup

Given that performance on the benchmark is measured by the R^2 of the regression defined in Eq. (3), our primary objective is to maximize this out-of-sample explanatory power by varying the context prompt p provided to the LLM. To ensure rigorous evaluation and prevent overfitting, we partition our sample of discrete information events into a training set, $\mathcal{E}_{train}$, and a validation set, $\mathcal{E}_{val}$. We index individual events (unique firm-time observations) by k.

To make the prompt optimization computationally tractable, we adopt a two-stage approach on the training data. First, we estimate the baseline regression model strictly over the training set:

$$\epsilon_k = \alpha + \boldsymbol{\theta}^\top \mathbf{X}_k + u_k, \quad \forall k \in \mathcal{E}_{train}. \tag{9}$$

We then extract the residuals, u_k, which represent the portion of the abnormal returns

unexplained by the controls. In the second stage, we optimize the prompt p by minimizing the mean squared error (MSE) between the AI system's predictions and these residuals. Formally, we search for the optimal prompt p^*:

$$p^* = \arg\min_{p \in \mathcal{P}} \frac{1}{|\mathcal{E}_{train}|} \sum_{k \in \mathcal{E}_{train}} (u_k - \hat{g}_p(\omega_k, \Omega_k))^2, \tag{10}$$

where $\mathcal{P}$ is the set of all possible prompts, and $\hat{g}_p(\cdot)$ is the scalar prediction from the LLM given prompt p.

After identifying p^*, we generate out-of-sample predictions, $\hat{g}_{p^*}(\omega_k, \Omega_k)$, for all events in the validation set. Final benchmark performance is then evaluated by estimating the full cross-sectional regression on $\mathcal{E}_{val}$:

$$\epsilon_k = \alpha_{val} + \gamma \hat{g}_{p^*}(\omega_k, \Omega_k) + \boldsymbol{\theta}_{val}^\top \mathbf{X}_k + \nu_k, \quad \forall k \in \mathcal{E}_{val}, \tag{11}$$

and calculating the R^2. We split the data from calendar quarter four of 2025 into training and validation sets, i.e., $\mathcal{E}_{train}$ and $\mathcal{E}_{val}$, based on the median date following Section 3.3.3.

While conceptually simple, there are several practical challenges to implementing solutions to Eq. (10). One of the most notable is the lack of continuous gradient information to steer changes to the prompt. As such, we focus our attention on exploring two state-of-the-art, gradient-free methods of prompt optimization: Genetic-Pareto (GEPA) search (Agrawal et al., 2026) and Agentic Context Engineering (ACE) (Zhang et al., 2026).

3.4.2 Genetic-Pareto Prompt Optimization

GEPA (Agrawal et al., 2026) is a prompt optimization algorithm that iteratively improves the instructions given to an LLM, i.e. the prompt or context p above. The method maintains a population of candidate prompts and evolves them through repeated selection, mutation, and evaluation. At each iteration, GEPA selects a candidate from the Pareto front—the subset of prompts that each achieve the highest score on at least one training instance—sampling in proportion to the number of instances on which each candidate leads. This selection mechanism encourages diversity across the candidate pool.

Once a prompt has been selected, it is applied to a mini-batch of training examples, and a scoring function identifies where the model succeeds and fails, returning textual diagnostics describing failure patterns. An LLM then reads the current prompt alongside these diagnostics and proposes a targeted revision, i.e., "reflective mutation." This reflection step involves the model itself diagnosing shortcomings in natural language rather than relying on random perturbations or gradient signals. If the revised prompt improves performance on

the minibatch it enters the candidate pool; otherwise it is discarded. The algorithm terminates when a computational budget is exhausted and returns the candidate with the highest average score on a held-out evaluation set.

Table 5 reports results from applying Genetic-Pareto (GEPA) prompt optimization to the training dataset $\mathcal{E}_{train}$ and evaluating Gemini 2.5 Flash-Lite with the optimized prompt out-of-sample on $\mathcal{E}_{val}$. During training, GEPA achieves a roughly 30% reduction in the MSE objective, indicating that the optimization procedure does meaningfully alter the model's predictions. However, the R^2 improvement in the evaluation set is modest. The baseline specification (Gemini with the default prompt, plus surprise) yields an R^2 of 10.5%, while the GEPA-optimized specifications achieve 12.8% under both loss functions. The 99% bootstrap confidence intervals for the baseline [0.06, 0.16] and optimized [0.08, 0.19] specifications overlap, though the point estimates suggest a gain of approximately 2.3 percentage points.

The reason for the gap between training improvement and evaluation R^2 becomes apparent upon decomposing the MSE objective and examining the optimized prompt. Beginning with the former, the MSE objective can be written as:

$$MSE(u, \hat{g}) = Var(u) + Var(\hat{g}) + 2Cov(u, \hat{g}) + \mathbb{E}[\hat{g}^2]. \tag{12}$$

As such, meaningful decreases in the objective can come from improving the covariance between the predictions and the residual—which would improve the R^2—or through other channels such as de-biasing the model, i.e., making $\mathbb{E}[\hat{g}] = 0$. Our empirical results suggest that de-biasing is the "path of least resistance" in our setting.

We can also examine the optimized prompt (see Figure 2) to explore whether the primary effect of GEPA is to de-bias the model's predictions. Doing so reveals the core directive which emerges from the optimization: "good is priced in." Effectively, the model learns that its default predictions systematically overestimate the market's response to strong operational results.

Specifically, the optimized prompt instructs the model to treat companies that report record profits and 20% growth as delivering a neutral outcome, not a positive one, because such performance is already reflected in the stock price. The prompt further develops sector-specific heuristics: for telecom and utility stocks, it caps predictions at 0.50 when subscriber churn rises, even if net additions are strong; for high-growth technology firms, it discounts long-term product roadmap promises in favor of immediate financial delivery; and for financials, it treats share buybacks as standard capital maintenance rather than positive catalysts.

Given the observed tendency of the model to focus on de-biasing, we additionally explore an optimization objective where we subtract the covariance between the residual and the

model's predictions. This has the effect of increasing the weight placed on the $Cov(u, \hat{g})$ term in Eq. (12). We do not find a significant increase in the R^2 under this objective, suggesting that directly optimizing for increased R^2 may be necessary—although debiasing alone is a valuable result.

These findings illustrate both a strength and a limitation of prompt optimization. The strength is interpretability: the optimized prompt reveals systematic biases in the model's default reasoning about markets. The limitation is that correcting a mean bias, while valuable, contributes less to R^2 than improving the model's ability to rank outcomes correctly. De-biasing primarily shifts the intercept of the prediction, whereas R^2 is driven by the correlation between predicted and actual returns.

3.4.3 Agentic Context Engineering (ACE)

ACE, introduced by Zhang et al. (2026), is a framework for optimizing the context provided to an LLM by building a structured collection of rules—termed a "playbook"—through an iterative generate-reflect-curate loop. In each iteration, a Generator module receives the current playbook and a task instance, attempts the task while recording its reasoning, and flags which existing rules proved useful or misleading. A Reflector module then analyzes this record, identifies patterns in errors and successes, and proposes targeted additions or modifications—small "delta" updates rather than a wholesale rewrite of the context. Finally, a Curator module integrates these deltas into the playbook: new rules are appended, existing rules are updated based on whether they improved or degraded performance, and duplicate or overlapping rules are removed.

This incremental architecture addresses a key failure mode of monolithic prompt optimization: because each iteration modifies only a small portion of the context, the method avoids "context collapse," in which successive LLM rewrites progressively compress and discard accumulated knowledge. The result is a modular, interpretable set of rules that grows and refines over time.

In Table 6, we find that ACE delivers a material improvement in explanatory power. Without ACE, Opus 4.5 with the surprise achieves an R^2 of 14.1% [0.09, 0.20] on the late subsample. With the ACE rulebook, the R^2 rises to 17.1% [0.12, 0.24], a gain of 3.0 percentage points. The coefficient on the LLM prediction is essentially unchanged (0.481 versus 0.484), while the coefficient on the surprise declines slightly from 0.175 to 0.166, suggesting that ACE enables the model to capture some information that was previously reflected only in the quantitative surprise. The R^2 of 17.1% represents more than a threefold improvement over the surprise-only baseline and, to our knowledge, the highest explanatory power reported in the earnings announcement return literature.

The ACE rulebook, like the GEPA prompt, is interpretable. It contains specific, actionable heuristics that reflect how markets actually respond to different types of earnings news (see Figure 3). For instance, the rulebook instructs the model that when a pre-profit company reports revenue growth exceeding 30% year-over-year but operating losses remain flat, the prediction should fall in the 0.10–0.20 range, because markets demand that high growth translate into financial improvement. Similarly, when management uses cautionary forward-looking language—phrases such as "aggressive consensus" or "gradual recovery"—the rulebook directs predictions to the 0.35–0.50 range. The rulebook also identifies interaction effects: when record margins are attributed to non-recurring benefits (such as precious metals revaluations) and the same quarter features an operational execution failure alongside a major geographic segment declining by more than 15%, the prediction should be 0.10–0.20 regardless of positive demand narratives elsewhere in the transcript. These rules echo well-established findings in the accounting and finance literature regarding the market's differential response to transitory versus permanent earnings components, but they are discovered entirely from data by the agentic system.

3.4.4 Model Size and Reasoning Effort

A natural question is whether the gains from ACE depend on the size and capability of the model that applies the rulebook. Table 7 addresses this by evaluating the ACE rulebook—developed using Opus 4.5 at high reasoning effort—with smaller models (Haiku 4.5 and Sonnet 4.5) and with Opus 4.5 at reduced reasoning effort.

The results reveal an important pattern: the ACE rulebook effectively transfers knowledge from a strong model to weaker ones. Without ACE, Opus 4.5 at high reasoning effort achieves an R^2 of 14.1%. With the ACE rulebook, Haiku 4.5—a substantially smaller and cheaper model—achieves 11.7%, Sonnet 4.5 achieves 12.1%, and Opus 4.5 at low reasoning effort achieves 13.9%. The last result is particularly striking: Opus 4.5 at low effort with ACE (13.9%) nearly matches Opus 4.5 at high effort without ACE (14.1%), despite requiring far less computation at inference time. This suggests that much of the value added by the most capable model lies in the structured reasoning that the ACE rulebook captures explicitly, rather than in the raw scale of the model's parameters or the depth of its inference-time computation.

These findings have practical implications for deploying AI systems to analyze earnings announcements at scale. The ACE rulebook can be developed once per quarter using the most capable model during the training period, and then applied to a cheaper, faster model for the remainder of the quarter. This "distillation through context" approach offers a favorable cost-performance tradeoff: the smaller models with ACE achieve R^2 values of

11.7–13.9%, compared to 14.1% for the full-scale model without ACE, at a fraction of the inference cost.

4 Implications for Asset Pricing Models

The interpretability of the context optimization methods we explore provides at least two important benefits over traditional machine learning (ML) methods. The first benefit relates to efficiency. Both methods we explore leverage contextualized feedback during optimization, i.e., after a model reasons about a specific instance of a forecasting problem and makes a prediction, its prediction is compared to the realized value and feedback is given on why the prediction was good or bad. This feedback—contextualized by the specific instance at hand—allows an LLM to more readily learn from even a single observation.[2]

To see this, consider a binary classification task where traditional ML methods learn by minimizing some loss function over predictions and targets: 0's and 1's (e.g., successful and unsuccessful predictions). The only feedback in this setting is the gradient of the loss, which for any single observation may be incredibly noisy. In contrast, ACE and GEPA explicitly reflect on *why* the prediction was (un)successful, a process which leverages the existing knowledge in the LLM and any information encoded in the current version of the rulebook or prompt.

To provide a simple analogy, a child learning math who receives only "correct" or "incorrect" as feedback, may take a long time to master the subject. If, instead, the child receives additional feedback *why* a particular answer is correct or not, she will learn more efficiently. In our context, an LLM reflecting on why a particular forecast is too optimistic or pessimistic may lead to rules that generalize to other earnings calls.

The second key benefit is that the optimized prompts and context can guide the design of better and more realistic asset pricing models. Specifically, the rulebooks learned by ACE or the prompts learned by GEPA provide a list of factors that matter to investors in the sense that focusing on these aspects of the call improves out-of-sample predictions. These factors should naturally feature in asset pricing models. As an example, if changes in guidance are of first-order importance to explain announcement returns, asset pricing models ideally account for guidance provided by firms to investors. Leading asset pricing models often include factors based on realized accounting fundamentals, prices, and returns, but guidance (or changes therein) does not feature prominently. Further, the optimized context often provides rules that are specific to a certain industry, which may be helpful to develop

[2] Agrawal et al. (2026) illustrate the efficiency gains of GEPA compared to traditional reinforcement learning methods.

industry-specific models.

In addition to explaining announcement returns, we can connect these same factors, i.e., those identified by ACE or GEPA, to future fundamentals. These connections then provide calibration targets for behavioral asset pricing models that often feature under- and overreaction to certain types of information. As emphasized throughout the paper, progress on our new benchmark naturally guides the development of new asset pricing models, be it rational, behavioral, or information-based models.

5 A New Competition to Explain Asset Prices

Our results thus far suggest that AI systems can materially increase our understanding of asset prices around earnings announcements. While these results are a contribution on their own, we do not intend for them to be the "definitive" results. For example, we anticipate further progress from including historical events, incorporating peers, or improved finance-specific harnesses for agents. Rather than working on these in a silo, we are introducing a competition with the explicit goal of improving our understanding of asset prices around information releases—related, Hellum et al. (2025) introduce a broad competition to identify the pricing kernel, in contrast, our aim is focused on understanding how information is priced. This follows a broader tradition in asset pricing to benchmark asset pricing models by their ability to explain differences in average returns. However, given the volatility of returns, it is challenging to discriminate between models during an out-of-sample period. Our benchmark instead can be evaluated during a single earnings season, which allows for the rapid evolution of optimized AI systems.[3]

In line with the principles in Section 2, the competition is open to anyone—the cost of running Gemini in our experiments was much less than $10 for an entire earnings season—and evaluated transparently. Participants need only register their system with the competition to receive programmatic access to (i) a common EA calendar and (ii) an opportunity to log a prediction in real-time when an EA occurs. Evaluation is then performed at varying time-scales, e.g., best system over the current earnings cycle to date, best over the last week, best today, and presented on a public leaderboard.

One challenge with evaluation is that EA transcripts are produced with a lag relative to the event, e.g., hours following the conference call. As a result, by the time that we have the transcript and request a participant's prediction, it is plausible that the participant could observe the market reaction which has already occurred and use that to guide their

[3]See also Gabaix et al. (2025) for cross-sectional benchmarks of models of asset and investor embeddings, which are methods to represent assets and investors beyond traditional characteristics.

prediction. In untabulated empirical analyses, we find that much of the reaction occurs within the time window between the call starting and the transcript being produced. To ensure fair play, we require all participants agree to an AI audit of their system if they are in the top 10 of the leaderboard to ensure that no information (other than the EA) is used in forming a prediction.

To incentivize participation and provide support for long-running systems, the competition includes prizes for the top three systems each quarter as well as compute/token grants for the top 10 systems that enable them to continue running until they fall out of the top 10. By defining a common set of events, evaluation methodology, and introducing a competition, we aim to spur progress on our understanding of asset prices similarly to how the introduction of the ImageNet Large Scale Visual Recognition Challenge (Deng et al., 2009) (i) led to dramatic decreases in the error rates of supervised models over only a few years (Russakovsky et al., 2015), and (ii) supported subsequent development of scalable self-supervised learning methods within computer vision (see, e.g., Chen et al., 2020; Caron et al., 2021).

6 Conclusions

Evaluating optimized AI systems for asset pricing is fundamentally difficult for at least two reasons. First, AI models are trained on all data, implying that any backtest or analysis using historical data suffers from look-ahead bias. In addition, markets are reflexive — as investors adopt AI, prices adjust — which may erode the very patterns the AI system was trained to exploit. We introduce a real-time, out-of-sample benchmark designed to sidestep both problems. The benchmark measures how well AI systems can explain contemporaneous stock returns around earnings announcements using only information available at announcement time, including the announcement text itself.

Applying this benchmark to a range of agentic AI systems — which extract structured signals from earnings call transcripts and optimize over those signals — we find that the best-optimized systems more than double the explained variation in returns relative to standard benchmarks (R^2 rising from 8% to close to 20%). Beyond predictive performance, these systems generate narrative explanations for price movements that yield testable implications for asset pricing theory. We release an SDK and launch an open competition inviting researchers to improve on our results. Saturating this benchmark would represent fundamental progress in understanding how capital markets process firm-level information and inform the development of future asset pricing theory.

References

Agrawal, Lakshya A, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab, "GEPA: Reflective Prompt Evolution Can Outperform Reinforcement Learning," in "The Fourteenth International Conference on Learning Representations" 2026.

Caron, Mathilde, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin, "Emerging Properties in Self-Supervised Vision Transformers," in "2021 IEEE/CVF International Conference on Computer Vision (ICCV)" 2021, pp. 9630–9640.

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in "Proceedings of the 37th International Conference on Machine Learning" ICML'20 JMLR.org 2020.

Chiang, Chin-Han, Wei Dai, Jianqing Fan, Harrison Hong, and Jun Tu, "Robust Measures of Earnings Surprises," *The Journal of Finance*, 2019, *74* (2), 943–983.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in "2009 IEEE Conference on Computer Vision and Pattern Recognition" 2009, pp. 248–255.

Drinkall, Felix, Eghbal Rahimikia, Janet Pierrehumbert, and Stefan Zohren, "Time Machine GPT," in Kevin Duh, Helena Gomez, and Steven Bethard, eds., *Findings of the Association for Computational Linguistics: NAACL 2024*, Association for Computational Linguistics Mexico City, Mexico June 2024, pp. 3281–3292.

Engelberg, Joseph, Asaf Manela, William Mullins, and Luka Vulicevic, "Entity Neutering," 2025.

Gabaix, Xavier, Ralph S.J. Koijen, Robert J. Richmond, and Motohiro Yogo, "Asset Embeddings," 2025. Working paper, Harvard University.

Glasserman, Paul and Caden Lin, "Assessing Look-Ahead Bias in Stock Return Predictions Generated By GPT Sentiment Analysis," 2023.

Hand, John R. M., Henry Laurion, Alastair Lawrence, and Nicholas Martin, "Explaining firms' earnings announcement stock returns using FactSet and I/B/E/S data feeds," *Review of Accounting Studies*, Dec 2022, *27* (4), 1389–1420.

He, Songrun, Linying Lv, Asaf Manela, and Jimmy Wu, "Chronologically Consistent Large Language Models," 2025.

Hellum, Oliver, Theis Jensen, Bryan Kelly, and Lasse Pedersen, "The Power of the Common Task Framework," 2025. SSRN Electronic Journal.

Hu, Shengyuan, Yiwei Fu, Steven Wu, and Virginia Smith, "Jogging the Memory of Unlearned Models Through Targeted Relearning Attacks," in "ICML 2024 Workshop on Foundation Models in the Wild" 2024.

Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis, "Highly accurate protein structure prediction with AlphaFold," *Nature*, Aug 2021, *596* (7873), 583–589.

Khattab, Omar, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts, "DSPy: Compiling Declarative Language Model Calls into State-of-the-Art Pipelines," in "The Twelfth International Conference on Learning Representations" 2024.

Lev, Baruch, "On the Usefulness of Earnings and Earnings Research: Lessons and Directions from Two Decades of Empirical Research," *Journal of Accounting Research*, 1989, *27*, 153–192.

Levy, Bradford, "Caution Ahead: Numerical Reasoning and Look-ahead Bias in AI Models," 2024. Working paper, Chicago Booth School of Business.

Merchant, Humzah and Bradford Levy, "A Fast and Effective Solution to the Problem of Look-ahead Bias in LLMs," in "NeurIPS 2025 Workshop GenAI in Finance" 2025.

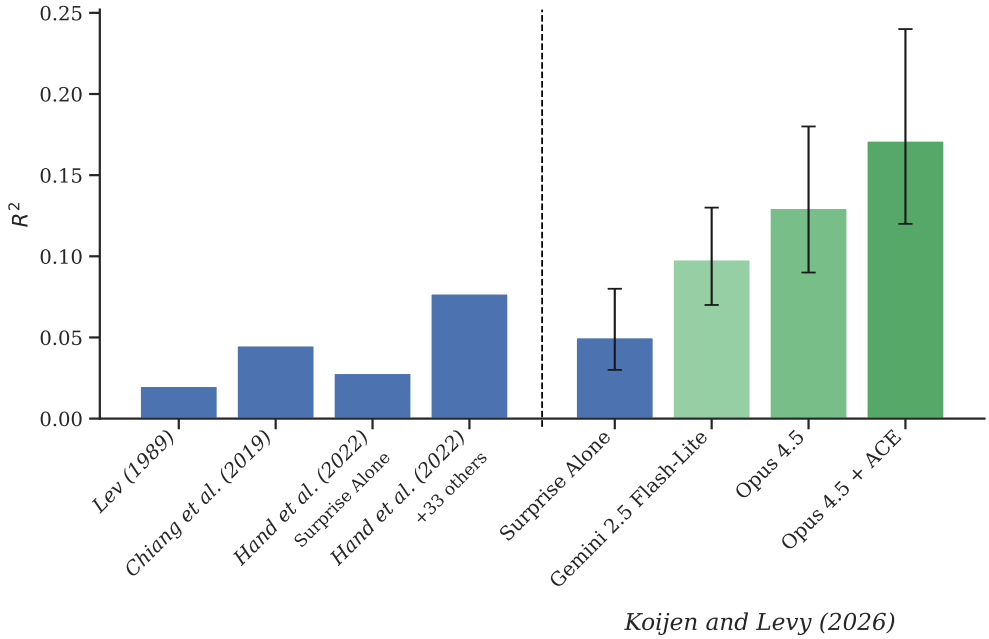
Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe, "Training language models to follow instructions with human feedback," in S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds., *Advances in Neural Information Processing Systems*, Vol. 35 Curran Associates, Inc. 2022, pp. 27730–27744.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, Dec 2015, *115* (3), 211–252.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in "Proceedings of the 36th International Conference on Neural Information Processing Systems" NIPS '22 Curran Associates Inc. Red Hook, NY, USA 2022.

Zhang, Qizheng, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun, "Agentic Context Engineering: Evolving Contexts for Self-Improving Language Models," 2026.

Figure 1: Progress on Explaining Asset Prices Around Earnings Announcements



Notes: For each paper, this figure plots the R^2 from a regression of abnormal returns over a narrow window around the earnings announcement on measures found to explain the returns.

Figure 2: Excerpt from the GEPA-Optimized Prompt

This figure presents excerpts from the prompt produced by Genetic-Pareto (GEPA) optimization, trained on $\mathcal{E}_{train}$ using Gemini 2.5 Flash-Lite as the target model. Panel A shows the core directive discovered by the optimization. Panel B presents sector-specific heuristics derived from past prediction errors. See Table 5 for evaluation results. Carriage returns indicated by "↪" are solely to indicate line wrapping in this presentation and are not part of the prompt.

Panel A: Core directive

The Core Directive: Assume "Good" is "Priced In"

Your previous predictions have consistently failed by overestimating the market's excitement for strong operational results.

* The Baseline is Success: Public companies are expected to grow, beat estimates, and improve margins. A company reporting record profits and 20% growth is usually just meeting the high expectations already built into the stock price. This is a Neutral (0.40 – 0.60) result, not a positive one.
* Neutral = The Graveyard of Good News: If a company reports a "beat and raise" that is merely incremental, or if they report strong earnings but flat guidance, the stock will likely not move or may even drift lower as traders "sell the news."
* The Bar for "Up" is Extremely High: To justify a prediction > 0.75, the news must be shocking (e.g., doubling guidance, a massive surprise buyback >10% of market cap, or a breakthrough product launch today).

Panel B: Sector-specific heuristics

Sector–Specific Heuristics (based on past errors):

1. Telecom/Utilities (e.g., AT&T): These are defensive yield stocks. High subscriber growth (net adds) is often ignored if churn ticks up even slightly or if free cash flow guidance isn't raised. The market fears competition more than it rewards growth here. Heuristic: If churn rises, cap prediction at 0.50.
2. High–Growth/Tech (e.g., Tesla): These stocks have massive premiums. Operational improvements (cash flow, margins) are expected. "Roadmap" items (future products years away) are often discounted as "vaporware" in the short term. Heuristic: Discount long–term promises. Only value immediate financial delivery.
3. Financials/Insurance (e.g., Travelers): Buybacks are standard operating procedure, not surprises. A $3B buyback is often just capital maintenance. "Underwriting discipline" (shrinking volume to maintain price) is often viewed as a growth headwind. Heuristic: Without a massive dividend hike or distinct catalyst, stick to 0.40–0.50.

Figure 3: Example Rules from the ACE Rulebook

This figure presents three representative rules from the rulebook produced by Agentic Context Engineering (ACE), trained on $\mathcal{E}_{train}$ using Opus 4.5 as the target model. Each rule specifies conditions observed in earnings call transcripts and the prediction range the model should assign. See Table 6 for evaluation results. Carriage returns indicated by "↪" are solely to indicate line wrapping in this presentation and are not part of the rulebook.

* When a pre−profit company reports strong revenue growth (>30% YoY) BUT operating
  ↪ losses remain flat or worsen (EBITDA/operating loss not narrowing proportionally),
  ↪ apply 0.10−0.20 range. Markets demand that high growth translate to financial
  ↪ improvement. Amplified when: (1) forward guidance suggests growth momentum
  ↪ peaking (flat sequential); (2) in hot theme sector (LiDAR, autonomy, quantum, etc.);
  ↪ (3) customer conversion rates below 15%.

* When management uses cautionary forward language ("aggressive consensus," "gradual
  ↪ recovery," guiding to low end), apply 0.35−0.50. When forward quarter guidance
  ↪ represents >50% deceleration from recent organic growth, apply 0.10−0.25. When
  ↪ one−time items inflate headline EPS with no forward acceleration, apply 0.10−0.20.
  ↪ CRITICAL: premium brand guidance below framework + DTC decline = 0.05−0.15.

* When record or near−record margins include acknowledged non−recurring benefits (precious
  ↪ metals, inventory revaluations, one−time mix) AND there is a concrete operational
  ↪ execution failure in the same quarter (equipment downtime, supply disruption, facility
  ↪ issue) AND a major geographic segment is declining >15% from external headwinds,
  ↪ apply 0.10−0.20 range regardless of multi−sector demand narratives.

Table 1: Summary Statistics for Q4 2025 Earnings Announcements. Panel A reports the number of unique earnings events by reporting week and Fama-French 12 industry classification. Weeks are indexed starting from the first week with observations in the quarter. Panel B reports the weekly mean, median, and standard deviation of one-day cumulative abnormal returns ($r_{i,t}^{\text{abn}}$) and the earnings surprise expressed as a percentage of price ($s_{i,t}$), along with the R^2 from a regression of percentile-ranked abnormal returns on percentile-ranked surprise. Panel C reports cross-sectional descriptive statistics for key firm characteristics. All continuous variables are winsorized at the 1st and 99th percentiles.

Panel A: Number of Observations by Week and Industry

Week	NoDur	Durbl	Manuf	Enrgy	Chems	BusEq	Telcm	Utils	Shops	Hlth	Fin	Other	Total
1	2	0	2	0	0	0	0	0	0	1	0	1	6
2	0	1	4	1	1	1	0	0	1	1	36	3	49
3	8	6	26	9	6	23	3	4	10	10	89	28	222
4	16	16	62	18	12	80	7	19	39	53	77	59	458
5	35	13	57	30	30	198	17	32	57	169	82	130	850
6	6	7	21	7	3	59	6	2	16	85	16	36	264
Total	67	43	172	65	52	361	33	57	123	319	300	257	1849

Panel B: Weekly Statistics

		$r_{i,t}^{\text{abn}}$			$s_{i,t}$ (%)			
Week	N	Mean	Med.	SD	Mean	Med.	SD	R^2
1	6	0.06	0.05	0.12	0.18	0.17	0.11	
2	49	-0.00	-0.00	0.07	0.13	0.12	0.16	0.00
3	222	0.00	-0.01	0.08	0.13	0.10	1.26	0.16
4	458	-0.01	-0.01	0.10	0.11	0.10	1.31	0.05
5	850	-0.01	-0.01	0.12	-0.02	0.14	2.93	0.05
6	264	-0.01	-0.02	0.12	-0.58	0.15	6.59	0.03
Total	1,849	-0.01	-0.01	0.11	-0.05	0.12	3.28	0.05

Panel C: Descriptive Statistics

Variable	N	Mean	SD	P25	Median	P75
Price	1,849	71.41	112.68	8.67	29.66	79.98
$r_{i,t}^{\text{abn}}$	1,849	-0.01	0.11	-0.06	-0.01	0.05
Coverage	1,796	8.30	6.20	3.00	6.00	12.00
Dispersion	1,694	0.09	0.16	0.02	0.04	0.10
$s_{i,t}$	1,849	-0.05	3.28	-0.03	0.12	0.44

Table 2: Explaining Contemporaneous Stock Returns Using Full Transcripts

	Dependent variable: $r_{i,t}^{abn}$						
	Baseline	Gemini 2.5 Flash-Lite	GPT-5 nano	Opus 4.5	Gemini 2.5 Flash-Lite	GPT-5 nano	Opus 4.5
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
α	0.384***	0.206***	0.026	0.140***	0.135***	-0.023	0.075***
	(0.013)	(0.024)	(0.040)	(0.024)	(0.025)	(0.039)	(0.025)
$s_{i,t}$	0.219***				0.190***	0.187***	0.181***
	(0.022)				(0.022)	(0.022)	(0.021)
$\hat{g}(\omega_{i,t}, \Omega_{t-})$		0.374***	0.726***	0.608***	0.343***	0.656***	0.565***
		(0.031)	(0.061)	(0.040)	(0.030)	(0.060)	(0.040)
R^2	0.050	0.075	0.071	0.109	0.112	0.107	0.143
	[0.02, 0.08]	[0.05, 0.11]	[0.04, 0.10]	[0.08, 0.15]	[0.08, 0.15]	[0.08, 0.15]	[0.10, 0.18]
Adjusted R^2	0.050	0.074	0.071	0.109	0.111	0.106	0.142
	[0.02, 0.08]	[0.05, 0.11]	[0.04, 0.10]	[0.08, 0.15]	[0.08, 0.15]	[0.08, 0.15]	[0.10, 0.18]
Observations	1849	1849	1849	1849	1849	1849	1849
F Statistic	97.282***	149.676***	142.150***	226.852***	116.796***	111.038***	154.169***

Notes: This table shows the results of regressing abnormal returns ($r_{i,t}^{abn}$) on the earnings surprise ($s_{i,t}$) and various LLM-derived predictions ($\hat{g}(\omega_{i,t}, \Omega_{t-})$) of the $r_{i,t}^{abn}$ using the full earnings call transcript ($\omega_{i,t}$). Standard errors are presented in parentheses under coefficient estimates. 99% confidence intervals based on 1,000 bootstrap samples are presented in brackets under estimates.

Table 3: Transcript Summaries

	Dependent variable: $r_{i,t}^{abn}$						
	Baseline	Gemini 2.5 Flash-Lite	GPT-5 nano	Opus 4.5	Gemini 2.5 Flash-Lite	GPT-5 nano	Opus 4.5
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
α	0.384***	0.283***	0.271***	0.210***	0.220***	0.198***	0.134***
	(0.013)	(0.020)	(0.030)	(0.022)	(0.021)	(0.030)	(0.023)
$s_{i,t}$	0.219***				0.180***	0.203***	0.189***
	(0.022)				(0.022)	(0.022)	(0.021)
$\hat{g}(\omega_{i,t}, \Omega_{t-})$		0.289***	0.295***	0.474***	0.252***	0.257***	0.442***
		(0.025)	(0.039)	(0.034)	(0.025)	(0.038)	(0.034)
R^2	0.050	0.065	0.030	0.093	0.098	0.073	0.130
	[0.03, 0.08]	[0.04, 0.10]	[0.01, 0.05]	[0.06, 0.13]	[0.07, 0.13]	[0.04, 0.11]	[0.09, 0.18]
Adjusted R^2	0.050	0.064	0.029	0.092	0.097	0.072	0.129
	[0.03, 0.08]	[0.04, 0.10]	[0.01, 0.05]	[0.06, 0.13]	[0.06, 0.13]	[0.04, 0.10]	[0.09, 0.17]
Observations	1849	1849	1849	1849	1849	1849	1849
F Statistic	97.282***	128.398***	57.137***	188.621***	100.273***	72.244***	137.546***

Notes: This table shows the results of regressing abnormal returns ($r_{i,t}^{abn}$) on the earnings surprise ($s_{i,t}$) and various LLM-derived predictions ($\hat{g}(\hat{\omega}_{i,t}, \Omega_{t-})$) of the $r_{i,t}^{abn}$ using summaries of the earnings call transcript ($\hat{\omega}_{i,t}$). Summaries are created using Opus 4.5. Standard errors are presented in parentheses under coefficient estimates. 99% confidence intervals based on 1,000 bootstrap samples are presented in brackets under estimates.

Table 4: Announcement Timing: Early versus Late

	Dependent variable: $r_{i,t}^{abn}$					
	Gemini 2.5 Flash-Lite		GPT-5 nano		Opus 4.5	
	(1) Early	(2) Late	(3) Early	(4) Late	(5) Early	(6) Late
α	0.230***	0.207***	0.251***	0.138***	0.150***	0.118***
	(0.029)	(0.030)	(0.041)	(0.046)	(0.032)	(0.033)
$\hat{g}(\omega_{i,t}, \Omega_{t-})$	0.222***	0.276***	0.170***	0.345***	0.398***	0.481***
	(0.034)	(0.038)	(0.050)	(0.059)	(0.047)	(0.051)
$s_{i,t}$	0.218***	0.156***	0.244***	0.176***	0.213***	0.175***
	(0.033)	(0.030)	(0.033)	(0.030)	(0.032)	(0.029)
R^2	0.093	0.105	0.066	0.085	0.118	0.141
	[0.06, 0.14]	[0.06, 0.16]	[0.04, 0.11]	[0.05, 0.14]	[0.07, 0.18]	[0.09, 0.20]
Adjusted R^2	0.091	0.103	0.064	0.083	0.116	0.139
	[0.05, 0.14]	[0.06, 0.16]	[0.03, 0.11]	[0.04, 0.14]	[0.07, 0.17]	[0.09, 0.20]
Observations	1032	817	1032	817	1032	817
F Statistic	52.506***	47.772***	36.262***	38.040***	68.869***	66.897***

Notes: This table shows the results of regressing abnormal returns ($r_{i,t}^{abn}$) on the earnings ($s_{i,t}$) and various LLM-derived predictions ($\hat{g}(\omega_{i,t}, \Omega_{t-})$) of the $r_{i,t}^{abn}$ using summaries of the earnings call transcript ($\hat{\omega}_{i,t}$). Summaries are created using Opus 4.5. The sample from Table 3 is split based on the median date into "Early" and "Late" sub-samples. Standard errors are presented in parentheses under coefficient estimates. 99% confidence intervals based on 1,000 bootstrap samples are presented in brackets under estimates.

Table 5: Genetic-Pareto Prompt Optimization

	Dependent variable: $r_{i,t}^{abn}$					
	Baseline		Loss = MSE		Loss = MSE - Cov	
	(1)	(2)	(3)	(4)	(5)	(6)
α	0.261***	0.207***	0.296***	0.217***	0.223***	0.151***
	(0.029)	(0.030)	(0.023)	(0.026)	(0.030)	(0.032)
$\hat{g}_{p^*}(\omega_{i,t}, \Omega_{t-})$	0.310***	0.276***	0.369***	0.352***	0.443***	0.419***
	(0.038)	(0.038)	(0.041)	(0.040)	(0.049)	(0.048)
$s_{i,t}$		0.156***		0.175***		0.172***
		(0.030)		(0.029)		(0.029)
R^2	0.075	0.105	0.090	0.128	0.091	0.128
	[0.04, 0.13]	[0.06, 0.16]	[0.05, 0.15]	[0.08, 0.19]	[0.05, 0.14]	[0.08, 0.18]
Adjusted R^2	0.074	0.103	0.089	0.126	0.090	0.126
	[0.04, 0.13]	[0.06, 0.16]	[0.05, 0.14]	[0.07, 0.19]	[0.05, 0.14]	[0.08, 0.18]
Observations	817	817	817	817	817	817
F Statistic	66.158***	47.772***	80.447***	59.945***	81.873***	59.875***

Notes: This table shows the results of using Genetic-Pareto (GEPA) prompt optimization to design the prompt used when making the LLM-derived predictions. Specifically, summaries are created using Opus 4.5 and then the sample from Table 3 is split based on the median date into "Early" and "Late" sub-samples. The Early sub-sample is used as a training set to which we apply GEPA to develop a prompt that minimizes one of two loss functions: (i) the mean squared error (MSE) or (ii) MSE minus the covariance between the actual and predicted $r_{i,t}^{abn}$. Using the Late sub-sample as an evaluation set, we present regressions of abnormal returns ($r_{i,t}^{abn}$) on the earnings surprise and the LLM-derived predictions ($\hat{g}_{p^*}(\omega_{i,t}, \Omega_{t-})$) under the optimized prompt p^*. Standard errors are presented in parentheses under coefficient estimates. 99% confidence intervals based on 1,000 bootstrap samples are presented in brackets under estimates.

Table 6: Agentic Context Engineering

	Dependent variable: $r_{i,t}^{abn}$			
	Baseline		With ACE Rulebook	
	(1)	(2)	(3)	(4)
α	0.195***	0.118***	0.199***	0.128***
	(0.031)	(0.033)	(0.027)	(0.029)
$\hat{g}_{p^*}(\omega_{i,t}, \Omega_{t-})$	0.502***	0.481***	0.506***	0.484***
	(0.052)	(0.051)	(0.045)	(0.044)
$s_{i,t}$		0.175***		0.166***
		(0.029)		(0.028)
R^2	0.103	0.141	0.137	0.171
	[0.06, 0.16]	[0.09, 0.20]	[0.08, 0.20]	[0.12, 0.24]
Adjusted R^2	0.102	0.139	0.136	0.169
	[0.06, 0.16]	[0.09, 0.20]	[0.08, 0.20]	[0.12, 0.24]
Observations	817	817	817	817
F Statistic	93.241***	66.897***	128.986***	84.077***

Notes: This table shows the results of using Agentic Context Engineering (ACE) to develop a "rulebook" used when making the LLM-derived predictions. Specifically, summaries are created using Opus 4.5 and then the sample from Table 3 is split based on the median date into "Early" and "Late" sub-samples. The Early sub-sample is used as a training set to which we apply ACE to learn a rulebook of general insights that improve predictions. Using the Late sub-sample as an evaluation set, we present regressions of abnormal returns ($r_{i,t}^{abn}$) on the earnings surprise and the LLM-derived predictions ($\hat{g}_{p^*}(\omega_{i,t}, \Omega_{t-})$) under the optimized context p^*. Standard errors are presented in parentheses under coefficient estimates. 99% confidence intervals based on 1,000 bootstrap samples are presented in brackets under estimates.

Table 7: Effect of Model Size and Reasoning Effort

	Dependent variable: $r_{i,t}^{abn}$			
		With ACE		
	(1) Opus 4.5 (High)	(2) Haiku 4.5	(3) Sonnet 4.5	(4) Opus 4.5 (Low)
α	0.118***	0.186***	0.238***	0.195***
	(0.033)	(0.030)	(0.025)	(0.027)
$\hat{g}_{p^*}(\omega_{i,t}, \Omega_{t-})$	0.481***	0.380***	0.323***	0.417***
	(0.051)	(0.047)	(0.039)	(0.045)
$s_{i,t}$	0.175***	0.172***	0.174***	0.164***
	(0.029)	(0.029)	(0.029)	(0.029)
R^2	0.141	0.117	0.121	0.139
	[0.09, 0.20]	[0.07, 0.18]	[0.07, 0.18]	[0.08, 0.21]
Adjusted R^2	0.139	0.115	0.119	0.137
	[0.09, 0.20]	[0.06, 0.18]	[0.07, 0.18]	[0.08, 0.20]
Observations	817	817	817	817
F Statistic	66.897***	54.119***	56.091***	65.714***

Notes: This table shows the interaction between using a rulebook developed using Agentic Context Engineering (ACE) and model size or reasoning effort. Specifically, summaries are created using Opus 4.5 and then the sample from Table 3 is split based on the median date into "Early" and "Late" sub-samples. The Early sub-sample is used as a training set to which we apply ACE to learn a rulebook of general insights that improve predictions. Using the Late sub-sample as an evaluation set, we present regressions of abnormal returns ($r_{i,t}^{abn}$) on the earnings surprise and the LLM-derived predictions ($\hat{g}(\omega_{i,t}, \Omega_{t-})$) under the optimized context p^*. Standard errors are presented in parentheses under coefficient estimates. 99% confidence intervals based on 1,000 bootstrap samples are presented in brackets under estimates.

A Variable Definitions

Table A1: Mathematical Notation and Variable Definitions

Variable	**Definition**
Time and Indexing	
i	Stock index
t	Time period (typically calendar quarter for evaluation periods)
t_c	Time of earnings conference call
d	Calendar date
t_0	Event date (first trading day when earnings information is available)
Returns and Performance	
$r_{i,t}$	Return for stock i on day t (including dividends)
$r_{m,t}$	Market return on day t (VTI)
$r_{i,t}^{abn}$	Abnormal return for stock i on day t ($r_{i,t} - r_{m,t}$)
$CAR_{3,i}$	Three-day cumulative abnormal return for stock i around event date
$p_{i,t+1}$	Percentile rank of stock i's return at announcement time $t+1$
$\hat{p}_{i,t+1}$	Predicted percentile rank for stock i at announcement time $t+1$
c_i	Discrete class for stock i (Down, Neutral, or Up)
Earnings and Fundamentals	
$e_{i,t+1}$	Reported earnings per share for stock i at time $t+1$
$\mathbb{E}_t[e_{i,t+1}]$	Consensus (mean) analyst forecast of EPS formed at time t
$\sigma_t(\mathbb{E}[e_{i,t+1}])$	Standard deviation of analyst estimates at time t
$s_{i,t+1}$	Standardized earnings surprise for stock i at time $t+1$
Sample Statistics	
N_t	Number of earnings announcements in evaluation period t
Regression Parameters	
α	Intercept coefficient
β	Slope coefficient (or β_1, β_2 for multiple regressors)
$\varepsilon_{i,t+1}$	Regression error term at announcement time $t+1$