Improving Forecasting with Language Models

How search-and-learn cut forecasting error on the biggest revenue surprises by about a third

Alistair Smallwood, Head of Applied AI at Primer

20 November 2026

ABSTRACT

A revenue forecast is most valuable, and hardest to make, when a company's quarter lands far from what analysts expected. Language models that read company documents tend to lean the right way on these big surprises but do not move their forecasts far enough. We set out to find out why, and to reduce the error.

We used a search-and-learn loop, where the model forecasts from company documents, inspects each miss against what the company then reported, and proposes changes that are kept only if they help on fresh companies. The loop showed that the model was not sizing evidence too cautiously: in most misses it had not found, or had not weighed, the evidence that mattered. The model builds its forecast segment by segment, so each piece of evidence can be tied to the part of the business it affects and sized against that segment's revenue. The change that worked adds a second step: after making its forecast, the model goes back through the documents looking for evidence the forecast missed, checks each piece against the figures for the segment it affects, and searches for longer before it stops.

We tested the result on 913 company-quarters from 434 companies reported in 2026, all after the training cutoff of the forecasting model, GPT-5.6, which read only public company documents and never saw consensus. On beats of 10% or more against consensus, average error was 34% lower than consensus adjusted for its usual bias; on misses of 10% or more, 36% lower. When the forecast sat 5% or more above consensus, 64% of those quarters became beats of more than 5%, against 18% of all quarters. We also tested the system on five model generations, from Opus 4.5 to Fable 5.2 and GPT-5.6, on quarters reported after each model's training cutoff. Newer models were generally better, but every one, left to itself, still fell well short on big beats and misses, and the search step helped every one. When the forecast does make a big call, it sizes it about right; what remains is the big surprises it does not call.

1 Introduction

Most quarters are close to what analysts expect. Forecasts matter most in the quarters that are not: a revenue beat or miss of 5% or more matters much more than an in-line print. These are also the quarters where analysts' consensus is least useful, because by definition it was wrong.

Language models can now read a company's filings, earnings call transcripts and presentations and produce a forecast from them. We wanted to know whether such a model can tell, before results, which companies will surprise by a lot, and whether it can say by how much. Our earlier

work suggested a specific weakness: models often lean the right way on a big surprise but do not move their forecasts far enough. We had called this a problem of magnitude.

This paper reports a larger test and a different explanation. The model does lean the right way on big surprises, and its error on them is well below consensus. But when we looked at the model's large calls, rather than at the large surprises, the calls were not too small. The shortfall comes from the many big surprises the model never called, and from evidence it did not find or did not weigh enough. Once it commits to a big call, it sizes it about right. Directing the model to search for what it missed helped every model we tried. For Fable 5.2 it raised the share of a big surprise captured from 15% to 39%, a larger step than the one from Opus 4.5 to Fable 5.2 with each model left to itself.

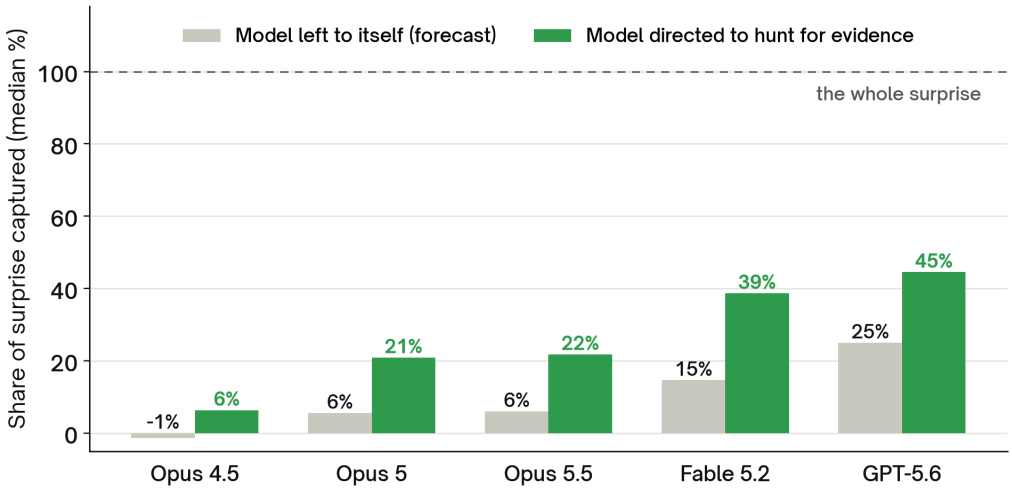
Section 2 shows this across five models. Section 3 describes the setting and data. Sections 4 to 6 report the main results: identifying big surprises, how much of each one the model captures, and whether its reasons were right. Section 7 describes how a search-and-learn loop led us to this explanation. Section 8 describes a data problem that inflates big surprises: stale consensus. Sections 9 and 10 discuss what this means and its limits.

2 Newer models are better, but still fall short on big surprises

We first ran the same forecasting system with five models on the 45 company-quarters of 2026 Q2 that were reported after every model's training cutoff. Each model made a forecast from the company's documents (left to itself) and then ran a second step, the “hunt”, which tells it to search for evidence its forecast missed and revise (directed). Ten of the 45 quarters were big surprises. Opus 5.5 and Fable 5.2, the newest models, have training data ending in June 2026; every Q2 result in this test was published in July 2026 or later, after every model's cutoff. Every model ran at its default reasoning effort, and made one run per case.

Two things are clear. Newer models see more: left to itself, the correlation between the forecast's departure from consensus and the actual surprise rises from 0.21 for Opus 4.5 to 0.62 for Fable 5.2. But left to itself, even Fable 5.2 captured a median of only 15% of each big surprise. Directed to hunt, every model did better: Fable 5.2 rose to 39% and GPT-5.6 to 45%, and the average miss on big surprises fell for all five models. Rerunning Opus 5.5 and Fable 5.2 at high reasoning effort did not help the “hunt”: average error changed by +0.07 points for Opus 5.5 (interval −0.23 to +0.43) and −0.02 for Fable 5.2 (interval −0.31 to +0.29). A more capable model, and directing it where to look, both mattered. More thinking time did not.

Figure 1. Newer models capture more; directing them adds more



This sample is small: ten big surprises and one run per model. The rest of the paper tests the same system at scale with one model.

3 Setting and data

Why GPT-5.6

A fair test needs quarters the model cannot have seen in training. GPT-5.6's training data ends on 16 February 2026, earlier than the other current models we use (Opus 5.5 and Fable 5.2 end in June 2026). Every result for 2026 Q1, Q2 and Q3 was published after that date, which gives three clean quarters and 913 company-quarters. We applied the same rule to every model used in forecasting, including the embedding model used for case search (released in January 2024).

What the model could read

The model used only company documents. It read public filings, results releases, earnings call transcripts and investor presentations, from the company itself and from other listed companies, and nothing else: no web search, no news, no share or commodity prices and no analyst research. Of the 82,126 search results served in the main run, 57% were filings, 30% call transcripts, 12% presentations and 0.3% other company documents such as prepared remarks.

The model read these documents, and each company's reported segment history, through Primer's document tools. These give agents fast, low-latency search over primary company filings, and serve only what was published before the case's cut-off time, which is the start of the day of the earnings announcement. The forecast is therefore a nowcast: the quarter has ended but its results are not yet out, and consensus is taken at the same moment. Documents with only a publication date count from the end of that day. A look-ahead audit before the main run checked five routes by which later information could reach a forecast: documents, timing and scoring, segment history, prompts and learned rules, and the model and its runtime. It found one misdated slide deck (in the model ladder, since excluded) and closed several weak points before the run.

Consensus

No model was given consensus at any point: not in prompts, tools or the segment history. The only analysts' expectations a model could meet were those quoted in public company documents, about quarters already reported. Consensus is used only to score. We take the consensus for company total revenue on the evening before results, with its broker count. Because companies have tended to beat consensus, we also use an adjusted consensus: consensus raised by the median beat across all companies over the 365 days before each case. This adjustment ranged from 1.15% to 1.45%, and uses nothing published after the case. In the test quarters 71% of company-quarters beat consensus, with a median beat of 1.42%, so the adjustment was well judged. Adjusted consensus is the harder benchmark.

The companies

We left out quarters with no consensus or broker count, quarters where a change in the company's structure or reporting perimeter broke its history, and a small number that failed our checks for leaked figures. Twenty company-quarters whose consensus had not been revised since the company's previous results, and one with a consensus data error, are set aside (section 8). That leaves 913 company-quarters from 434 companies: 347 in Q1, 302 in Q2 and 264 in Q3. Of these, 208 are big surprises: 163 beats and 45 misses of more than 5% at the total revenue level.

Country of listing	Companies	Company-quarters	Big surprises
United States	394	835	203
Canada	4	8	0
Europe	31	58	4
Middle East (Qatar, United Arab Emirates)	2	6	1
Other (Cayman Islands, China)	3	6	0
All	**434**	**913**	**208**

Table 1. Where the companies are listed. Europe: Sweden 18, Norway 2, Netherlands 2, Ireland 2, Germany 1, Switzerland 1, Finland 1, United Kingdom 1, Luxembourg 1, Belgium 1, Denmark 1.

Market value	Companies	Company-quarters	Big surprises
Large (over $10bn)	194	394	86
Mid ($2bn to $10bn)	143	325	61
Small (under $2bn)	95	191	59
Not recorded	2	3	2

Table 2. Companies by market value, as recorded in our company table in November 2026.

Sector (GICS)	Companies	Company-quarters	Big surprises
Industrials	114	229	46
Consumer Discretionary	75	172	24
Information Technology	61	133	25
Financials	30	69	21
Energy	24	63	31
Materials	30	62	18
Consumer Staples	25	55	3
Utilities	19	44	21
Communication Services	14	35	7
Health Care	29	29	6
Real Estate	12	20	4
Not classified	1	2	2

Table 3. Companies by sector.

Most companies are listed in the United States. Large, mid and small companies are all well represented, and big surprises are more common among smaller companies: small companies make up 21% of all company-quarters but 28% of big surprises.

The forecasting system

For each company-quarter the model builds its revenue forecast segment by segment from the company's documents, using the reported history of each segment, and adds the segments up to a total. Building by segment matters for big surprises: a reason for a beat or a miss, such as a new contract, a price rise or a weak end market, usually belongs to one part of the business, and tying it to that segment gives the model a base against which to size it. A second call, the “hunt”, receives that forecast and its stated assumptions and is told to search for evidence the forecast missed, to check each piece of evidence against a stated figure rather than a phrase, and to test the company's habit against its own guidance. “Our forecast” in what follows is the forecast after the “hunt”. We score company total revenue against the first figure the company reported, and measure error as the absolute difference divided by reported revenue. Intervals come from resampling companies, 2,000 times, so that several quarters from one company are not counted as independent evidence.

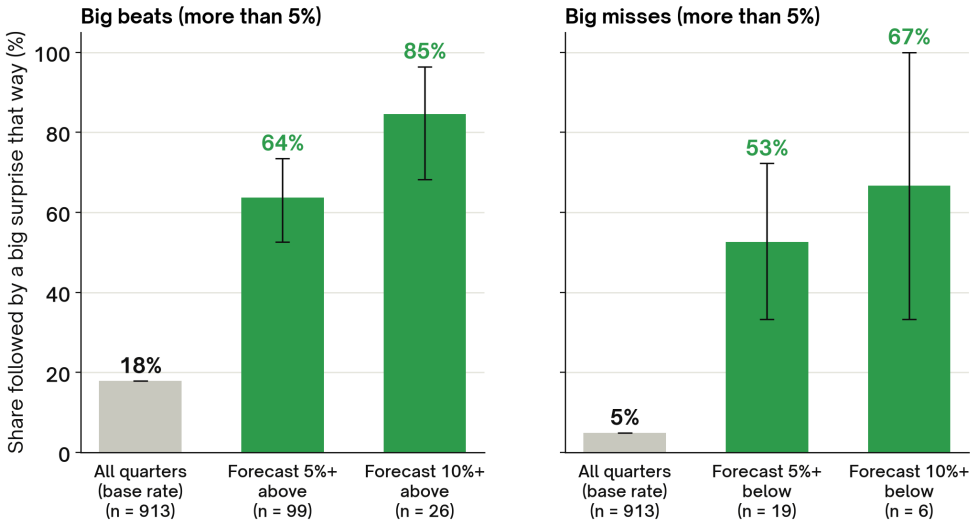
4 Can the model tell which companies will surprise?

The fair way to ask this uses only what was known before results: group the quarters by how far our forecast sat from consensus, then look at what happened.

Figure 2 shows this in two panels, one for beats and one for misses. A big beat is a quarter where reported revenue came in more than 5% above consensus; a big miss, more than 5% below. The grey bar in each panel is the base rate: of all 913 quarters in the sample, 18% were big beats and 5% were big misses. The green bars take only the quarters where our forecast, made before results, sat well away from consensus: 5% or more above it (or below it, for misses), and 10% or more. Each green bar shows the share of those quarters that then turned into a big beat (or a big miss). If the forecast carried no information, the green bars would be the same height as the grey one. The black lines on each bar show the range within which the true share is likely to lie (95% intervals, from resampling companies).

What the chart shows. When our forecast was 5% or more above consensus, it correctly predicted a big beat 64% of the time, against 18% for quarters in general. When it was 10% or more above consensus, the quarter was a big beat 85% of the time. When our forecast was 5% or more below consensus, it correctly predicted a big miss 53% of the time, against 5% in general; at 10% or more below, 67% of the time. There were few such downward forecasts (19 and 6), so those figures are less certain.

Figure 2. A strong call before results is often followed by a big surprise

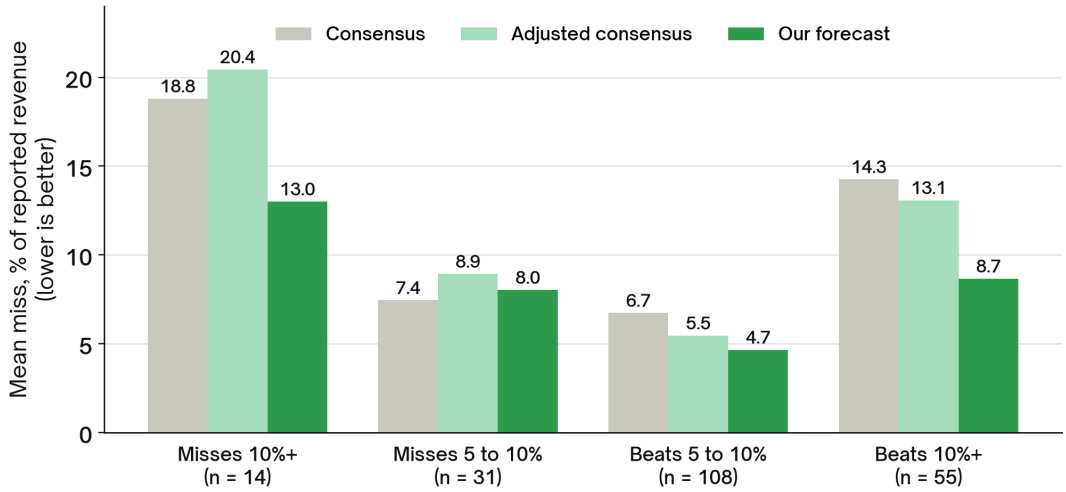


Smaller downward calls did not work: forecasts 2% to 5% below consensus were followed by results slightly above it on average.

5 How much of each big surprise does the model capture?

Section 4 asked whether the forecast points the right way. This section asks how close it comes. For each big surprise we measure the average error of our forecast, as a share of reported revenue, and compare it with the error of consensus and of adjusted consensus. In all the charts in this section, lower is better. Figure 3 groups the big surprises by how far reported revenue landed from consensus: misses and beats of 5% to 10%, and of 10% or more. Figure 4 shows every range, including the quarters close to consensus.

Figure 3. Error on big surprises



On beats of 10% or more, our forecast's average error was 8.7%, against 14.3% for consensus and 13.1% for adjusted consensus: 39% and 34% lower. On misses of 10% or more it was 13.0%, against 18.8% and 20.4%: 31% and 36% lower. Adjusted consensus does worse than plain consensus on misses because the adjustment raises it, the wrong way when revenue falls short. On beats of 5% to 10% the gain is smaller, and on misses of 5% to 10% there is none: our forecast was no better than consensus. The gain holds in each quarter: across all big surprises, error was 21% lower than adjusted consensus in Q1, 29% in Q2 and 25% in Q3.

Figure 4. Error across every range of outcome

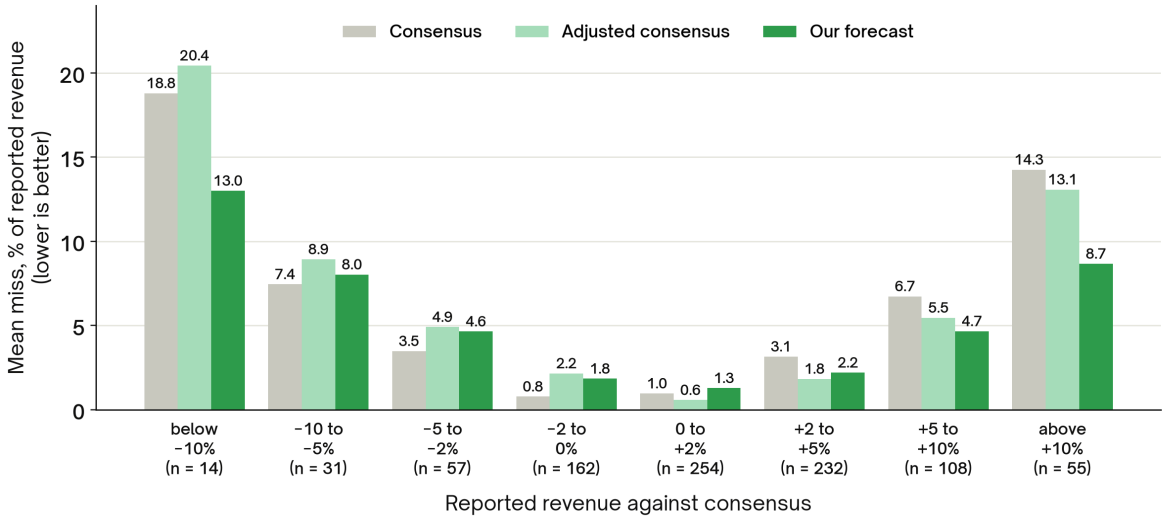


Figure 4 shows that the gain is concentrated in the outer ranges. In quarters that landed within 2% of consensus, consensus was more accurate, as it must be when grouping by its own error: these are the quarters where it was nearly right.

These groups are defined by the outcome, which flatters any forecast that differs from consensus, so they should be read alongside section 4, which does not depend on the outcome. They also show the limit. On beats of 10% or more, the typical forecast moved only about 41% of the way from consensus to the reported figure.

Overall accuracy

Figure 5. Mean and median error

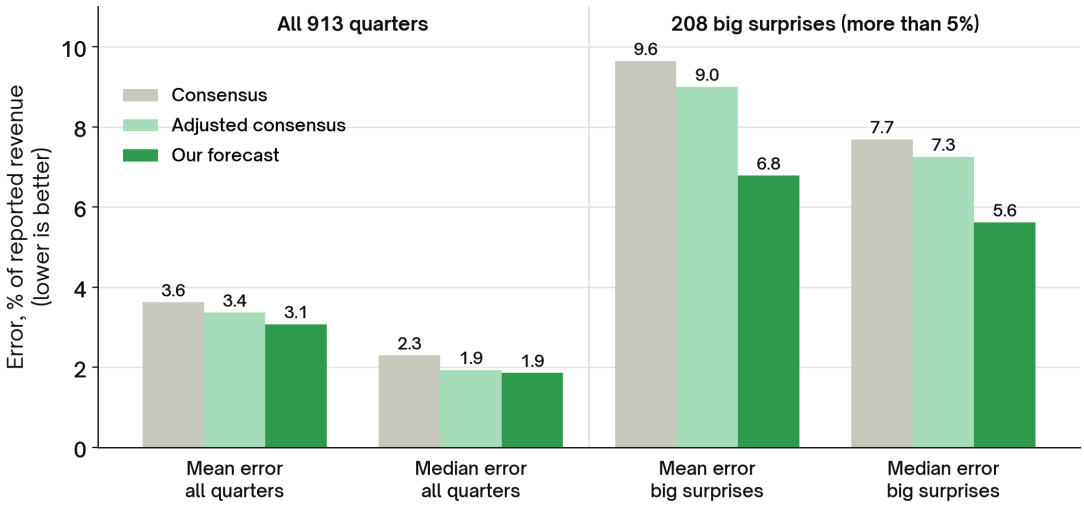


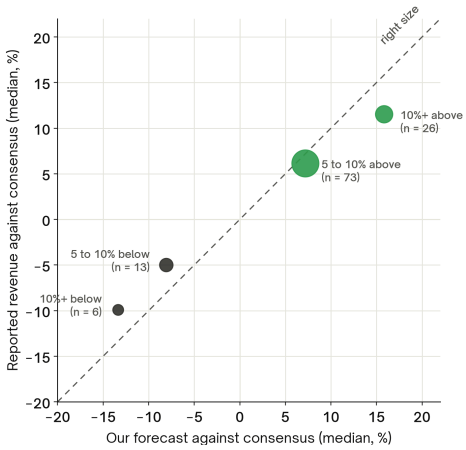
Figure 5 sets these results against the whole sample. Across all 913 quarters, our forecast's mean error was 3.1%, against 3.6% for consensus and 3.4% for adjusted consensus; its median error was 1.9%, against 2.3% and 1.9%. On the 208 big surprises the gap is much wider: a mean of 6.8% against 9.6% and 9.0%, and a median of 5.6% against 7.7% and 7.3%. In ordinary quarters the forecast is about as accurate as adjusted consensus; the gain comes from the big surprises.

Two views of sizing

Averages over the big surprises, as in Figures 3 and 5, make the forecast look too cautious: on beats of 10% or more it moved only about 41% of the way. But those averages mix two different kinds of quarter: big surprises the model did not see coming, where its forecast stayed close to consensus, and big surprises it did anticipate. When it did anticipate one, it was usually right (section 4).

Figure 6 shows whether those big calls were the right size. It takes every quarter where our forecast sat 5% or more from consensus, groups them by how far it sat (5% to 10%, or 10% or more, above or below), and plots the median result against the median forecast. The size of each bubble shows the number of quarters. Points on the dotted line were exactly the right size.

Figure 6. When the forecast makes a big call, the size is about right



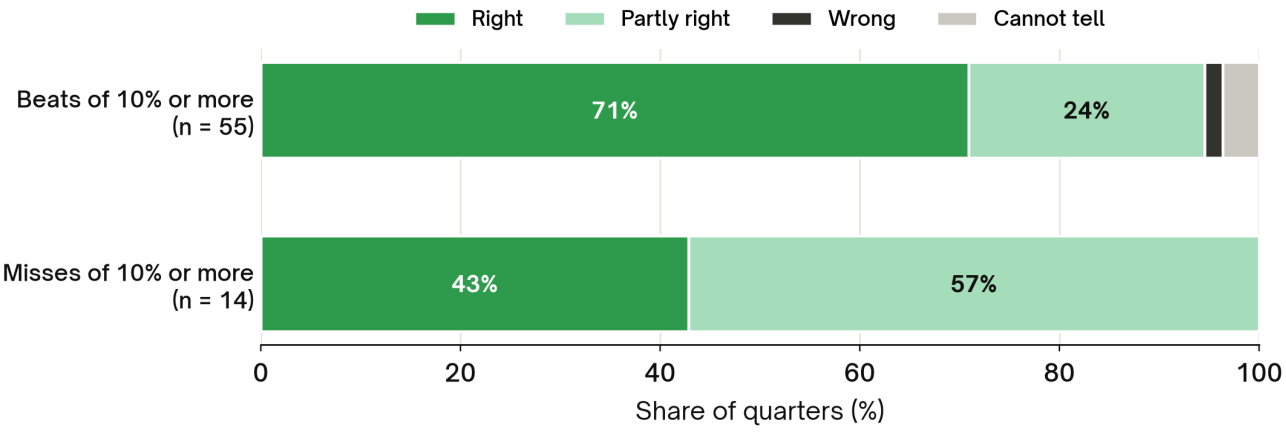
If anything, the big calls were a little too large: on calls of 10% or more above consensus, the median forecast was 15.8% and the median result 11.6%.

6 Were the reasons right?

An AI forecast is only useful if the reason behind it is right. A forecast that lands close for the wrong reason cannot be relied on next quarter, cannot be checked by an analyst, and gives an investor no basis for acting on it. So beyond accuracy, we asked whether the model's reasoning was right. For the 69 quarters where revenue landed 10% or more from consensus, we checked each forecast's own reasoning against the company's documents for the reported quarter: its results release and its earnings call transcript, both from Primer. The check was made after the results by Claude Opus 5.5, which saw the outcome.

We asked a simple question: did the forecast's reasoning name the cause of the surprise? We looked anywhere in its reasoning, and did not mind how much weight it gave the cause or how far it moved the forecast. “Right” means it named the main cause, pushing revenue the way it went; “partly right” means it named a real cause that mattered but not the main one, or named the main factor but expected it to push the other way; “wrong” means it named none of the real causes.

Figure 7. Did the forecast name the real cause?



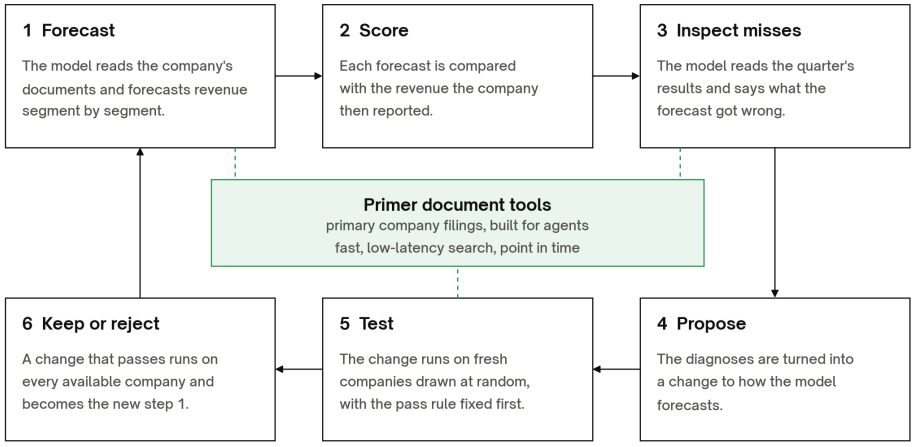
On big beats, the forecast named the main cause in 39 of 55 cases and part of it in 13 more; only 1 named none. A typical case is Micron: the forecast said memory prices would keep rising in a supply-short market, which is what the company reported.

On big misses, it named the main cause in 6 of 14 and part of it in the other 8. Houlihan Lokey is an example: the forecast flagged that deal timing could defer closings, which is what happened, though it gave this little weight and still forecast growth. Aldar Properties is a partial case: the forecast named weaker hotels, a real but small cause, and missed the deferred land sales that drove the quarter. On the downside the model usually saw at least part of what was coming, but gave it too little weight.

This is a generous test: a cause counts as named even if the forecast mentioned it only in passing.

7 How we found that the model was not finding the right evidence

Figure 8. The search-and-learn loop



We did not start with this explanation, and no person proposed the change that produced it. The “hunt” came out of our search-and-learn loop (Figure 8), which is built to take people out of the work of improving the forecast. A person designed the loop, set its goal and approved the budget for each test; AI agents did the rest. The model forecasts from company documents; each forecast is scored against the reported quarter; a model inspects the misses, reading that quarter's own results, and says what went wrong; and an AI research agent reads those diagnoses, proposes changes and tests them. When an earlier change, a second step that revised the forecast, only averaged in the extra information it was given, the agent proposed sending the model back into the documents to look for evidence its forecast had missed. That was the “hunt”. Every step reads documents through Primer's tools, which serve only what was public at each case's cut-off time.

Every proposal was first tried on a small set of cases, often chosen because the forecast had gone wrong. A proposal that helped there was then tested on fresh companies it had never seen, drawn at random before any forecast was made (within bands of surprise size, so that big surprises were represented), with the pass rule fixed in advance. Only a proposal that survived went on to the full run on every available company. The method was developed on cases from 2026 Q1 and Q2; no 2026 Q3 quarter was used in its development. On the 50 big surprises in Q3, error was 25% lower than adjusted consensus, in line with the other two quarters. Most proposals did not survive: company graphs and notebooks, each company's own beat history, similar situations at other companies found by vector search, query tools over a library of past cases, pointers to analyst coverage, methods learned from misses, a pre-mortem, a size rule and more reasoning effort.

These were the key findings flagged by the loop, which led it to propose a “hunt” stage and then to refine it:

- Reviewing 893 misses from an earlier baseline run after the results, a model judged that the forecast had captured the main driver of the quarter in only 46, about 5%.
- Across 190 “hunt” forecasts from four development stages, forecasts built on 10 or fewer tool calls moved the right way 50% of the time; those with 11 or more tool calls, 72%. This is an association: harder cases may also draw more searching.

The loop's proposals from these diagnoses became the “hunt”.

8 Stale consensus

Some big surprises are not really surprises. They come from a consensus figure that nobody had updated.

While checking the reasons in section 6, we found several quarters where the consensus figure looked out of date, so we checked all 934 quarters we had scored. We counted a consensus figure as stale if no analyst had revised it since the company last reported results. 20 quarters failed this test. Most of them (16 of 20) were a single analyst's estimate, typically about nine months old. Their surprises averaged 14.8%, against 3.8% for all other quarters, because the estimate had fallen behind the company.

We removed these 20 quarters, and one more where the consensus figure was plainly wrong: it sat 26% below the same quarter a year earlier, although consensus had tracked the company closely for the seven quarters before.

Removing them made our results weaker, not stronger: some of our apparent wins had been against estimates nobody had updated.

9 Discussion

Data quality comes first. Before asking whether a model can forecast, it is worth asking whether it is reading the right information and being scored against the right benchmark. Stale consensus (section 8) made the model look better than it was until we removed it. Working from primary company filings, served point in time through Primer's document tools, means every figure the model reads can be traced to the company's own words on a known date, with search fast enough for an agent to read widely before it forecasts.

This system uses close to the least information possible. It reads public company documents and nothing else. It sees no news, no share prices, no commodity prices and no proprietary data. Several of its biggest misses turned out to be driven by things it could not possibly have known from company documents alone: GeoPark and Valero on oil and fuel prices, and Targa on gas and natural gas liquids prices.

More information is not automatically better. In our tests, giving the model more to read without telling it what to look for did not help: libraries of similar cases, peers' results and the company's own record of beats all failed on fresh companies. Only directed search did. New sources need to be added in the same way: tested, tuned and kept only if they help on companies the system has not seen.

Where this leads. Added in that way, we expect proprietary data, news, and share and commodity prices to make this system substantially better than it already is, and to let it outperform analysts' consensus on big surprises across far more companies than any analyst team can cover. That is our expectation rather than a finding of this paper, and it should be tested on quarters that have not yet been reported.

10 Limits

- **Exploratory.** The main test is retrospective, on quarters after the model's training cutoff. It is not a forward test with a frozen forecast.
- **One model at scale.** The 913-case run uses GPT-5.6 only. The model ladder uses 45 cases and ten big surprises.
- **One period.** Three quarters of 2026, in which most companies beat consensus.
- **The reasons check** was made by a model that saw the outcome, with our own spot checks.
- **Revenue only.** We forecast and score company total revenue, not profit, and we do not test whether the forecasts would have made money.

We are extending this work to profit lines, to news and price data, and to quarters not yet reported. To discuss the results or run the system on your own coverage, contact alistair@primerapp.com.