Machine Learning as a Tool for Theory Induction

Steven Ma* Sendhil Mullainathan† Ashesh Rambachan‡

November 2026§

Abstract

Machine-learning algorithms often out-predict economic theories in a wide range of settings. Yet a learned algorithm is not portable: it is a function of one dataset's variables, and it makes no predictions anywhere else. Economic theories are the opposite: they can be carried to settings far from the ones in which they were written, and that portability is much of why economics values theories in the first place. We develop an end-to-end procedure that induces portable theories from a machine-learning algorithm. The first challenge to overcome is search: we need to look for functions written in the "language" that theories belong to. We solve this by using symbolic regression — a genetic algorithm that searches over some prespecified set of variables and operations. The second challenge is a less transparent one: how do we ensure that the functions we discover are portable? We argue that portability comes not from functions but from the words that describe how to map those functions into new contexts; we call this *binding*. We instantiate binding with large language models, which we embed in search to ensure that it only produces functions that port to a broad set of contexts. We illustrate the procedure on financial analysts' quarterly earnings forecasts, where a machine-learning algorithm out-predicts five known theories of expectations. The procedure induces ten new theories; the best-fitting one is a generalization of sticky expectations which says that forecasters keep responding to news long after it arrives and become less sticky when there is bad news. These theories close a third of the gap between the known theories and the machine-learning algorithm, and also predict well on annual forecasts. Even more than the specific theories we induce, we see our main contribution as a framework for specifying the desired properties of a theory, and a modular theory induction algorithm whose components can be independently improved and adapted. We hope that future work builds on our approach to develop increasingly capable induction algorithms and refine our understanding of what exactly we want from them.

1 Introduction

Economists have long created theories to capture regularities in the world. Now we have a different tool for discovering regularities, if not in the world, at least in data: machine learning. The differences between these two approaches are stark: one mines the data, and the other builds on human insight; one is often an uninterpretable black box, and the other is most prized when it is clear and compact. Recent empirical work heightens the tension between them. When we have rich datasets, algorithms are often able to find patterns that existing theories have missed. So much so that across a wide range of contexts, machine learning seems to predict better than our theories

*Department of Economics, Harvard University. Email: stevenma@g.harvard.edu.

†Department of Economics and Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology. Email: sendhil@mit.edu.

‡Department of Economics, Massachusetts Institute of Technology. Email: asheshr@mit.edu.

§All results are preliminary.

do, in economics (Peysakhovich and Naecker, 2017; Fudenberg and Liang, 2019; Fudenberg et al., 2022) and in other sciences, from protein structure (Jumper et al., 2021) to the weather (Lam et al., 2023). That all raises the question: can we not somehow integrate these two approaches? Why not extract the regularities implicit in machine-learning models and use them to create novel theories?

The desire to extract theories from machine learning is not simply a preference for the familiar. Theories are portable in a way machine-learning algorithms are not. The algorithm is an artifact with a single use — predicting outcomes in the setting it was trained on; there is little guarantee it will do well even on new distributions, much less in whole new settings (and it is often unclear how you would port a machine-learning model to radically new settings). Theories, on the other hand, are remarkably portable. They apply to a whole range of settings: a theory of rational expectations applies to macroeconomic forecasts, financial analysts' forecasts, and forecasts in the lab. How would we even take a machine-learning model trained on analysts' quarterly earnings forecasts and apply it to households' inflation expectations? Portability is especially valuable when we want to understand new contexts where there is not enough data to train machine-learning models (e.g. macroeconomics).

In this paper we develop an end-to-end *induction procedure* that produces novel theories from machine-learning algorithms. Doing so requires overcoming two challenges.

The first challenge is the most obvious. The economic theories we're most familiar with — expected utility theory, CAPM, Nash equilibrium — all look different from each other. Yet despite this variation, none of them look anything like a neural network, which has millions of parameters arranged across layers. Metaphorically, the functions we use in theories appear to be written in one "language" with a "grammar" in which neural networks do not belong. That then is the transparent challenge of induction: how can we extract familiar looking mathematical functions from complex networks while still preserving the predictive power? The challenge is one of search. Machine learning often uses neural networks to fit data because many optimization procedures exist that search their space efficiently. We need analogous procedures to search over the more rugged grammar of functional forms that typically constitute a theory.

We overcome this challenge by turning to a method from computer science: symbolic regression (Koza, 1990; Schmidt and Lipson, 2009; Udrescu and Tegmark, 2020; Cranmer et al., 2020). Symbolic regression searches over functions constructed from a prespecified set of operations and variables, subject to a complexity cap. As we describe later, using symbolic regression in economic contexts requires deciding on the grammar of theories (which now is no longer a metaphor but an actual choice: which variables and operations functions can be built from). Importantly, the specific implementation we choose in this paper can be easily adjusted or expanded in the future.

The second challenge is less transparent. It is one we did not even appreciate until we started building an induction procedure. If a procedure using earnings expectations data to induce theories of expectations produced

$$\text{forecast}_t[\text{earnings}_{t+1}] = \text{earnings}_t + \beta\,(\text{earnings}_t - \text{earnings}_{t-1})$$

at a time when the idea behind it (Metzler, 1941) was not known, this would be a success. Not because of the equation. But because of something we intuitively see in it. We read it not just as a statement about earnings but as a broader idea, one to which we attach the words "extrapolative expectations". Having read it as such, we can then reapply this equation to household inflation expectations, where we would estimate $\text{forecast}_t[\text{inflation}_{t+1}] = \text{inflation}_t + \beta\,(\text{inflation}_t - \text{inflation}_{t-1})$. Notice though we have now performed a map: earnings_t in the analyst context has been mapped to inflation_t in this context. As humans, we do this without thinking. What allows us to do that? The words. If we want an algorithm to find such theories, it must know that this equation has words

that are portable and can be applied elsewhere. Note that an equation without words is itself not particularly portable. If our procedure had instead just returned the equation $y = x_1 + \beta\,(x_1 - x_2)$, we wouldn't know how to port it: how would one know what x_1 and x_2 correspond to in a novel context?

What this and many other examples make clear is: having familiar looking functions does not by itself constitute a theory. The words are essential, and are also how we determine which contexts a theory applies to and which variables in a dataset are linked to which concepts in a theory. We call this *binding*. Binding is a central part of science, but an underappreciated one: we do it so often that we sometimes forget it exists. Crucially, notice the need for words is not arising because of human limitations. It is actually being revealed in the act of attempting to make induction computational. Words are the mechanism that provides theories their defining characteristic, portability.

This observation turns the second challenge into something concrete: how do we attach words to functions and what constraints are there on binding? Presumably not all words can attach to a given function. The challenge is that we must be able to computationally approximate valid bindings. This is especially important because we don't want to just do naive symbolic function search but also want to embed binding into our search process: we are looking for functions built from our variables and operations that have descriptions with sufficient portability.

In Section 3 we lay out a formalism of binding. A theory is a function class along with a verbal description of what it captures. A context is a setting with inputs and an outcome, along with a description of what each of them is. A theory then makes predictions when a researcher *binds* the inputs in the context to the variables in the theory. In Section 4 we then use binding to formalize three crucial properties we want induced theories to have: predictive accuracy, portability and atomicity. We note that we want portable theories not only because they apply more broadly but also because we often think they are the ones that perform best out of sample. Portability is a criterion that all sciences implicitly value; our induction framework makes this explicit.

Armed with this framework, we provide an algorithm that induces new theories (Sections 4.2–4.7). We first search for functions by running symbolic regression subject to an algorithmic bindability constraint. We instantiate bindability using large language models (LLMs); then to make search run more efficiently, we also build a random forest proxy trained on LLM verdicts. We then need to run binding in the other direction, turning each function the search returns into a theory. Naively asking an LLM to attach theories to functions produces narrow descriptions that fail to port widely; to combat this issue we use an iterative LLM loop that explicitly asks it to attach more and more portable theories.

Findings. The specific setting we focus on is financial analysts' earnings forecasts, where our goal is to induce new theories of expectations. Our first step (Section 2) is to benchmark the predictive performance of existing theories and understand whether there is room for improvement. We build a machine-learning algorithm $\hat{m}$ and find that it achieves nearly double the combined R^2 of five known theories from Afrouzi et al. (2023). The goal of the rest of the paper is then to distill the machine-learning algorithm into theories.

In Section 5 we use this procedure to induce theories of expectations. First, our algorithm does not just produce one theory: it produces ten, each finding additional R^2 over the previous ones. Second, the best two together already achieve a higher R^2 (23.4%) than the combination of all five known theories (21.8%). And all ten together explain 27.5% of the variation in forecast revisions, closing 32.5% of the gap between the known theories and the machine-learning algorithm. We take this to be substantial progress, because R^2 over and above what is known is hard to find: when we add the known theories one at a time in order of publication, the gains level off quickly, and the

last two add only 0.6%.

Qualitatively, the theories are diverse and range from modified versions of theories we know to ones that look quite different. The theory with the highest R^2, which we call *sticky (unless there's bad news) expectations*, generalizes sticky expectations (Mankiw and Reis, 2002; Woodford, 2003; Coibion and Gorodnichenko, 2015) with two components: a dependence on the history of observed news, and a stickiness that falls after bad news. Other theories look very different: one says forecasters revise down when recent news has been volatile, and two say forecasters respond to market prices more than to other news of the same informativeness. We can empirically demonstrate the diversity of our theories too: the average pairwise correlation between two of our theories is 0.17, against 0.47 between the known theories. Finally, each of the theories is portable well beyond analyst forecasts as well, as we can see from their descriptions in Table 2.

Testing our theories in other settings is not the main goal of this paper, but it is nonetheless interesting to see how they perform. The easiest setting for us to benchmark is annual earnings forecast revisions. Our two best theories again match or beat any of the known theories, and the collection of all ten closes 18.9% of the gap between the known theories and a machine-learning benchmark.

Finally, we also empirically demonstrate the necessity of binding in our procedure. First, we show that running symbolic regression without a bindability constraint produces a function that predicts well but cannot be ported into other contexts. Second, we demonstrate that an attempt to port the ten functions we discovered via (our bindable version of) function search directly to annual forecasts, without using any words, produces predictions with an R^2 of approximately zero.

Where this leaves us. What should one make of these results? On the one hand, we should take the R^2 gains seriously, especially since the theories are portable and appear to work in a new setting. Their improvements are also of the kind that has driven progress in this literature before: refinements of known mechanisms alongside new ones, written in ways that are familiar and parseable to economists. On the other hand, we do not think they reflect fundamental breakthroughs that reshape how we think about expectations. What would it take to generate such breakthroughs?

We see the contribution of this paper as not only the theories we have already induced, but also setting up the conditions to start moving toward such breakthroughs. There are three ways we could get there, each requiring more innovation than the last. First, we ran the procedure on a fairly bare-bones dataset: though we have a wide range of financial variables, there is no unstructured data that could capture further context about either firms (e.g. 10-K filings) or analysts (e.g. their written reports). Enriching the inputs gives our existing algorithm a better chance to find richer and more unexpected theories, much as the hypotheses generated by the machine-learning algorithm of Ludwig and Mullainathan (2024) would have been out of reach without rich image data on defendants' mugshots. Second, we can easily improve our algorithm in many ways that our framework already accommodates. For instance, in addition to requiring the search to deliver portable theories, we could ask it to match stylized facts that an algorithm can check, such as overreaction (Section 4.7). Third, we may achieve even larger breakthroughs by extending our framework to accommodate properties we do not yet know how to formalize, such as those that only appear once we derive a theory's implications for other outcomes (e.g. whether a theory of expectations generates momentum in an asset-pricing model). Adding such a criterion would first require defining it precisely enough for an algorithm to check, just as this paper gives a clear and checkable notion of portability.

We also note that across the history of computing, many of the most important innovations

came at the level of frameworks and algorithms, yet there was often a long lag between those ideas and truly transformative results. Progress in language models is a good example. Some of the key ideas came early: training neural networks to predict the next word dates back at least to Elman (1990), representing words as learned embeddings dates back to Bengio et al. (2003), and the transformer architecture to Vaswani et al. (2017). Yet it took many years and other complementary advances, such as faster chips, larger datasets and better training methods, to deliver the language models we see today. If our framework ultimately delivers breakthroughs, it will be because others continue to build on it in novel and perhaps unanticipated ways.

Related work. Our paper is most closely related to the literature that studies the intersection of machine learning and economic theory; see Camerer (2019), Mullainathan and Rambachan (2025), and Liang (2025) for reviews. Fudenberg and Liang (2019) use a machine-learning predictor of initial play to find the games where existing theories perform poorly, which then guide the researcher to write down an improved model. Hirasawa et al. (2025) do a similar exercise for mixed-strategy play. Mullainathan and Rambachan (forthcoming) use machine learning to automatically generate anomalies for a theory, and Ludwig and Mullainathan (2024) generate morphed pairs of mugshots from which a human can name interpretable hypotheses. Manning et al. (2024) use language models to generate social-science hypotheses and test them in simulated experiments. Our work differs from these in that we are the first to offer a framework for theory induction that formalizes portability, and deliver an improvable and adaptable algorithm that induces theories from it. Andrews et al. (2025) study the transfer performance of economic theories and machine-learning models under distribution shift, whereas we are interested in portability across contexts with different variables (e.g. macro forecasts vs. financial forecasts). Ellis et al. (2024) port machine-learning algorithms across contexts by imposing specific assumptions about how their predictions should change when moving from two inputs to three inputs. We bypass this problem by inducing theories rather than trying to directly transfer machine-learning models. Fudenberg et al. (2022) use machine-learning algorithms as a benchmark to measure the completeness of economic theories, a measure which is then used in Fudenberg and Puri (2023) and Hirasawa et al. (2025). More broadly, we contribute to the development of methods that extract interpretable structure from machine-learning models (e.g. Mullainathan and Rambachan, forthcoming; Cranmer et al., 2020).

Finally, we contribute to the literature on expectation formation. A long theoretical literature studies how people form expectations (Muth, 1961; Lucas, 1972; Cagan, 1956; Mankiw and Reis, 2002; Coibion and Gorodnichenko, 2015; Sims, 2003; Woodford, 2003; Fuster et al., 2010; Bordalo et al., 2018, 2019). We contribute to this line of work by proposing several new theories. An empirical literature then tests these models in the data, often in the form of documenting predictable forecast errors (Coibion and Gorodnichenko, 2012, 2015; Greenwood and Shleifer, 2014; Coibion et al., 2018a; Bordalo et al., 2020; De la O and Myers, 2021; Afrouzi et al., 2023), including some that compare forecasts to machine-learning benchmarks (Bianchi et al., 2022; de Silva and Thesmar, 2024). We also contribute to the empirical literature by benchmarking theories' R^2 rather than just testing for forecast errors.

2 Predicting analyst expectations

2.1 Analyst earnings forecasts

Our goal in this paper is to develop a general procedure for inducing new theories from machine learning. Before doing so, we first need to pick a setting in which to induce theories. We pick

expectations (i.e. how people form beliefs) because they are central to economics: In macroeconomics, inflation expectations are central objects and are the subject of a long literature (Coibion and Gorodnichenko, 2012, 2015; Coibion et al., 2018a). In asset pricing, a large recent literature connects expectations of returns and cash flows with asset prices (Greenwood and Shleifer, 2014; Bordalo et al., 2019; De la O and Myers, 2021). And in corporate finance, investment depends on managers' expectations of earnings growth (Gennaioli et al., 2016).

There is an equally long theoretical literature on expectations, which offers a useful benchmark for any theories we induce. Its starting point is rational expectations (Muth, 1961; Lucas, 1972); much of the work since then focuses on modeling deviations from rationality (Mankiw and Reis, 2002; Sims, 2003; Woodford, 2003; Bordalo et al., 2018).

If we want to use machine learning to induce theories of expectations, what data should we use? Any dataset we use needs *many rows*: machine learning is data-hungry. It also needs *many columns*, so that there is a rich enough set of features from which we can induce new theories. And finally, we would ideally like a setting of economic importance.

To satisfy these three constraints, we use data from professional financial analysts' earnings forecasts from IBES (the Institutional Brokers' Estimate System). In particular, we focus on analysts' consensus quarterly earnings forecasts. This dataset contains 486,943 firm-quarters, which satisfies our many rows requirement.

We can also construct a very large set of relevant columns for this dataset: any relevant information about a firm in a given quarter is a reasonable column to have. We have 9,606 columns, including past realized earnings, past consensus forecasts, analyst disagreement statistics, and a wide range of firm-level financial variables.[1] The full list is in Table A1, and Appendix A.1 describes how we build them.[2]

Finally, the stakes are high: producing forecasts is analysts' full-time job, and the reports containing them are sold.

Notation and the forecast revision. Throughout, i indexes firms and t indexes quarters. We write $Y_{i,t}$ for firm i's realized earnings per share in quarter t, and $F_t\big[Y_{i,t+h}\big]$ for the consensus forecast, made at time t, of firm i's earnings per share in quarter $t+h$.[3] Summary statistics for the IBES quarterly panel are in Table A2.[4]

A theory of expectations should aim to explain the forecast $F_t\big[Y_{i,t+1}\big]$. But it turns out that, in this setting, the lagged consensus $F_{t-1}\big[Y_{i,t+1}\big]$ already attains an R^2 of 92.4% on its own because revisions are small relative to the variation in forecasts across firms and over time. The harder and more interesting question is what predicts forecasts over and above lagged forecasts. Therefore, all of our quantitative benchmarks in this paper will be relative to explaining the forecast revision

$$\Delta F_{i,t} \;=\; F_t\big[Y_{i,t+1}\big] - F_{t-1}\big[Y_{i,t+1}\big],$$

and we will induce theories of forecasts by first trying to explain the forecast revision and then turning them into theories of forecasts by adding back on the lagged forecast (Section 4).

[1] Future work can build on our results by enriching the data with unstructured data such as firms' 10-K filings or analyst reports.

[2] We do not include any macroeconomic data: there are relatively few macro data points, so there is not much data from which machine learning can reliably learn out of sample.

[3] Concretely, "made at time t" is the consensus at the *second* monthly snapshot after firm i's announcement of quarter t's earnings. The first snapshot after the announcement can still contain stale individual forecasts made before $Y_{i,t}$ was announced; by the second, every analyst will have had the chance to update. If only one consensus snapshot exists in the window between announcements, we use it.

[4] Summary statistics are evaluated on the entire dataset.

2.2 Known theories versus the black box

Before putting in the effort to induce new theories in this setting, we should first start with a simpler question: do the current theories already explain all the predictable variation?

To answer this question, we first build a black-box predictor of forecast revisions, which we call $\hat{m}$. If the predictive performance of $\hat{m}$ sits above the R^2 attained by the combination of known theories, that is a sign that there is still predictive signal left that a new theory could potentially discover. $\hat{m}$ predicts $\Delta F_{i,t}$ using the 9,606 columns of Table A1; it is a stacked ensemble of gradient-boosted trees, neural networks, and sequence models, and Appendix A.4 gives the details of how we build it.

To benchmark our machine-learning predictor, we construct a sample split at the firm level: 30% of the 14,917 firms in the panel are randomly assigned to a holdout, and the rest are randomly assigned (stratified by firm coverage and size) to 5 training+validation folds (see Figure A1 for a visual). The benchmarks in this section are out of sample, pooled over the five folds; Appendix A.2 gives the details of the split. In the end $\hat{m}$ attains an out-of-sample R^2 of 0.402 on forecast revisions.

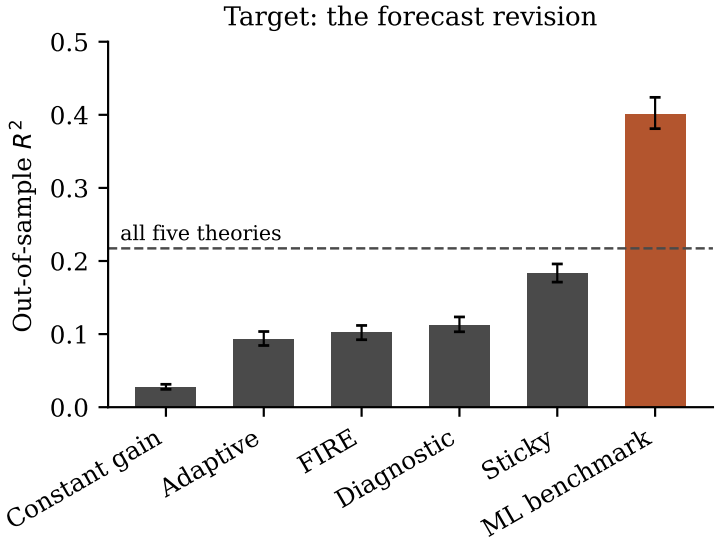
Next, we need to pick a set of known theories to benchmark our machine-learning predictor against. For simplicity, we take the set of five known theories from Afrouzi et al. (2023): adaptive expectations, full-information rational expectations, sticky expectations, diagnostic expectations, and constant-gain learning.[5] Descriptions of the five theories are shown in Table A3 in the appendix.

These theories are written about the forecast, not the revision, so we first derive each one's implication for the revision (Appendix A.5). Some of them also reference objects we cannot compute: rational expectations, for example, uses the true conditional expectation of next quarter's earnings given everything known at t. For the pragmatic purpose of having benchmarks, we replace it with $\hat{\mathbb{E}}_t\big[Y_{i,t+1}\big]$, a machine-learning forecast of next quarter's earnings from all fundamental information in our dataset known at t. More details about our construction of these theory benchmarks are in Appendix A.5.

The results are in Figure 1. We target the forecast revision $\Delta F_{i,t}$, and we compare the machine-learning benchmark $\hat{m}$ to the five known theories one at a time, as well as to the combination of all five in a linear regression (the dashed line). The best single theory is sticky expectations, with an R^2 of 0.183. All five known theories together reach 0.217. $\hat{m}$, on the other hand, reaches 0.402, *almost twice the five known theories combined*: machine learning has discovered a remarkable amount of new signal in forecast revisions. This large gap indicates that there is likely room for significantly better theories to be induced.[6] Appendix Figure A2 repeats the comparison with $\hat{m}$, rather than the forecast revision, as the target, and finds a similar gap: all five known theories together explain only 0.579 of the variation in $\hat{m}$.

[5] They also include extrapolative expectations, but it turns out to have the same implied functional form for revisions as adaptive expectations.

[6] Like $\hat{m}$, the five known theories are evaluated out of sample: their free parameters are estimated on four folds and the R^2 is computed on the held-out fifth, and we rotate through all five folds and pool the held-out predictions.



Notes. The target is the forecast revision $\Delta F_{i,t}$, computed on the 283,800 firm-quarters in folds 1–5, pooled over the five-fold rotation (each fold scored by a fit on the other four). Each theory's expectation term $\hat{\mathbb{E}}$ is fit out of fold. Whiskers are 95% bootstrap intervals; the bootstrap resamples firms.

Figure 1: Out-of-sample R^2 of the five known theories, compared with the machine-learning benchmark.

Our goal in the remaining sections is to develop a procedure that distills our machine-learning algorithm into new theories that close this gap. In the following section, we first develop a framework that formalizes the notion of theories.

3 Theories and binding

3.1 Theories and binding

Before inducing any theories, we first formalize what a theory is and how theories make predictions across contexts; examples are in the following subsection.

Definition 1 (Theory)**.** *A theory is a pair $T = (\mathcal{F}_T, D_T)$. $\mathcal{F}_T$ is a class of functions $f\colon \mathcal{X}^* \to \mathcal{Y}^*$ from the theory's inputs $x^* \in \mathcal{X}^*$ to its outcome $y^* \in \mathcal{Y}^*$. D_T is the theory's description in words.*

Definition 2 (Context)**.** *A context is a triple $C = (\mathcal{X}_C, \mathcal{Y}_C, D_C)$, where $\mathcal{X}_C$ is the input space, $\mathcal{Y}_C$ is the output space, and D_C is a verbal description of the context and its variables.*

How does a theory make predictions in a context? The first step is that a researcher needs to *bind* the variables of the context, x_C, to the variables of the theory, x^*. This is already carried out informally every time a researcher applies a theory to a dataset; our goal here is to formalize it so that algorithms can begin to use and induce new theories.

Definition 3 (Binding)**.** *The* binding *of T in C is either a function from the context's inputs x_C to the theory's variables x^*, or $\emptyset$ if the theory does not apply to the context:*

$$B_{T,C}\colon \mathcal{X}_C \to \mathcal{X}^* \quad \text{or} \quad B_{T,C} = \emptyset.$$

The binding function comes from the verbal descriptions of the theory and the context via a mapping $(T, C) \mapsto B_{T,C}$.[7] *When* $B_{T,C} \neq \emptyset$*, the predictions of* T *in* C *are therefore the function class* $\mathcal{F}_T \circ B_{T,C}$*, and we say that* T binds to f *in* C *if* $f \in \mathcal{F}_T \circ B_{T,C}$.[8]

3.2 Why a theory needs words

Why does a theory need a description D_T, and why do we need binding? We are often used to thinking of economic theories as just the set of allowable functions $\mathcal{F}_T$. To see why that is not enough, consider the $\mathcal{F}_T$ of a well-known theory with its description stripped out:

$$\mathcal{F}_T = \Big\{ y^* = g^{-1}\Big(\sum_{k=1}^{n} x^*_{n+k}\, g\big(x^*_0 + x^*_k\big)\Big) - x^*_0 : g' > 0 \Big\}.$$

What is this theory? It is likely not immediately obvious from the functional form itself.

What if we bring back the description? If

> D_T = "people value risky options by weighing the utility of the wealth each payoff leaves them with by its probability: x^*_0 is the person's wealth w, $x^*_1, \ldots, x^*_n$ are a lottery's payoffs $v_1, \ldots, v_n$, $x^*_{n+1}, \ldots, x^*_{2n}$ are their probabilities $p_1, \ldots, p_n$, and y^* is the most the person would pay for the lottery,"

then T becomes recognizable: it is expected utility. Of course, nobody writes expected utility in terms of x^*; we write it in the more recognizable form WTP $= u^{-1}\big(\sum_k p_k\, u(w + v_k)\big) - w$. But the point is: $\mathcal{F}_T$ is mathematically equivalent whether we write it using x^* or p_k, v_k and w, so the theory must gain its meaning from the words behind the variables. Usually this is implicit: we may already think "probability" when we see p_k. We just want to point out that however one does it, it is always crucial to communicate the words behind the variables.

Binding without words. We just saw that theories are unrecognizable without words. Is this just a communication problem, or do the words actually do substantive work? We argue it is the latter. Imagine applying expected utility to a variety of settings, such as how much insurance to buy (Mossin, 1968) or how to price assets (Lucas, 1978). How would you know how to map it into these contexts if all you had was the $\mathcal{F}_T$ written above, without words telling you what each of the x^*'s means? To make this concrete, let's return to this paper's domain of expectations and consider an even simpler theory, extrapolative expectations (Metzler, 1941). It has

$$\mathcal{F}_T = \big\{ y^* = x^*_1 + \beta\,(x^*_1 - x^*_2) : \beta \geq 0 \big\},$$

> D_T = "forecasts extrapolate the latest trend in realized outcomes: x^*_1 and x^*_2 are the latest two realized values of a time series, Y_t and Y_{t-1}, and y^* is the forecast of its next value, $F_t[Y_{t+1}]$."

Now imagine applying $\mathcal{F}_T$ without D_T to our analyst earnings expectations context. Which of the 9,606 columns in our context maps to x^*_1 and which to x^*_2? Nothing in $\mathcal{F}_T$ tells you, and there are $9{,}606 \times 9{,}605 \approx 92$ million ways to make the mapping (Figure 2). $\mathcal{F}_T$ itself makes almost no meaningful restrictions.

[7] In a slight abuse of terminology, we will also use the word binding to refer to this mapping, as well as to the general link we need to make between equations or data and semantic content.

[8] For simplicity we assume the theory's outcome y^* corresponds directly to the context's output, so that the functions in $\mathcal{F}_T \circ B_{T,C} = \{f \circ B_{T,C} : f \in \mathcal{F}_T\}$ produce outputs in $\mathcal{Y}_C$.

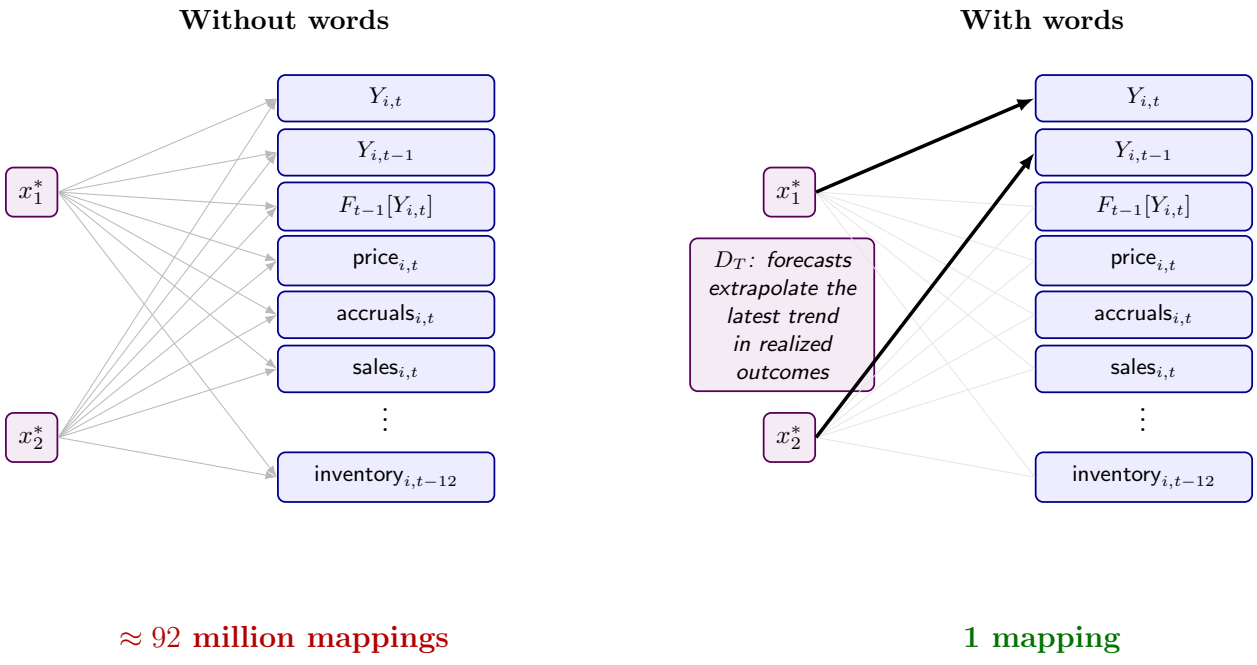


Figure 2: Binding the analyst context's columns to extrapolative expectations, without and with the words.

Binding with words. Once we bring back the theory's description, it becomes obvious how to apply $\mathcal{F}_T$ to analyst expectations: we read the descriptions D_C and D_T of the context and the theory, which tell us how to *bind* the former's variables (x_C) to the latter's (x^*), as in Definition 3. To see this concretely, we bind extrapolative expectations into three contexts (also shown in Figure 3):

- *Analysts' earnings forecasts.* D_C says that the setting is professional analysts forecasting quarterly earnings and that the inputs are past earnings, past consensus forecasts, past financial fundamentals, and so on. The latest trend in realized outcomes is the change in the firm's earnings, so

$$B_{T,C}(x_C) = \big(Y_{i,t},\, Y_{i,t-1}\big), \qquad \mathcal{F}_T \circ B_{T,C} = \big\{Y_{i,t} + \beta\,(Y_{i,t} - Y_{i,t-1}) : \beta \geq 0\big\}.$$

- *Professional forecasters' inflation forecasts.* D_C says that the setting is professional forecasters forecasting inflation. The latest trend in realized outcomes is the change in inflation, so

$$B_{T,C}(x_C) = \big(\pi_t,\, \pi_{t-1}\big), \qquad \mathcal{F}_T \circ B_{T,C} = \big\{\pi_t + \beta\,(\pi_t - \pi_{t-1}) : \beta \geq 0\big\}.$$

- *Experimental subjects valuing lotteries.* D_C says that the setting is experimental subjects valuing lotteries one at a time. This is not a context that involves forecasting or trend extrapolation, so $B_{T,C} = \emptyset$.

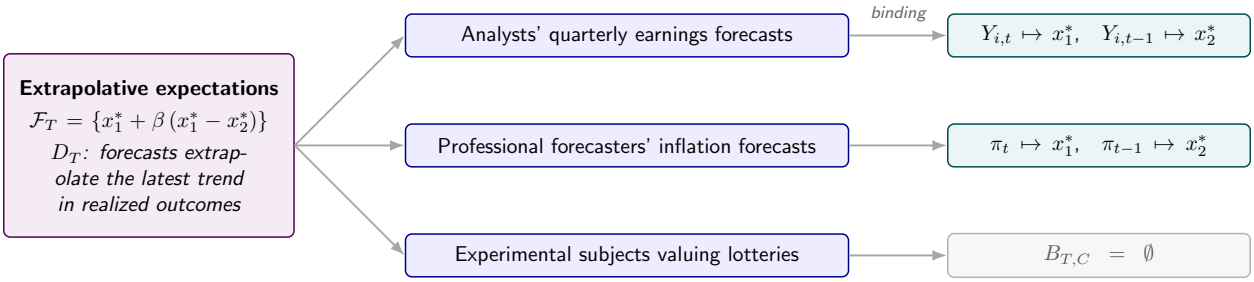


Figure 3: Binding extrapolative expectations into three contexts.

Appendix B.2 gives more examples, including one where binding not only selects columns from $\mathcal{X}_C$ but also transforms them. Binding provides a huge amount of information: in the analyst setting it reduces the 92 million mappings to a single one, determined by the words. Proposition 1 (Appendix B.4) quantifies exactly how much information binding supplies.

Relation to existing notions of theories. How does our definition of a theory, which involves descriptions, relate to existing notions in the literature? One long line of papers views economic theories, in a given context, as just the set of allowable functions a theory permits. [9] If all we care about is one context, a function class that lives directly on its variables is all we need; in our framework, that is $\mathcal{F}_T \circ B_{T,C}$. But if we want a theory to apply across contexts with different variables, that is insufficient: a function class directly defined on the variables in analysts' earnings forecasts will not exist in inflation forecasts. Words and binding are necessary for the same theory to make predictions in both.

Three properties of binding. In our examples of binding, we encountered a few interesting properties. Each of them is already captured by Definition 3, but is worth making explicit. First, as the lottery example above shows for extrapolative expectations, binding may return $\emptyset$: this happens when the theory's description D_T does not apply to the context, or when D_T does not actually describe $\mathcal{F}_T$ (D_T should tell you how to apply $\mathcal{F}_T$).[10] Second, to do binding we needed the context's and the theory's descriptions *but not the realized data*. This captures the idea that a theory needs to make falsifiable predictions before any data is collected; it also allows the theory to make predictions in new settings with no data. Third, the binding map $(T, C) \mapsto B_{T,C}$ was determined by the words' semantic meaning; it is therefore not a mathematically defined object. It is nonetheless well defined — we were able to use D_T and D_C to produce bindings, and the fact that others would bind in the same way means that a theory's predictions are fixed. In this paper, because we are interested in automatically inducing theories, we will often use LLMs to carry out binding. We note that LLMs are the natural tool for this problem because binding relies on words.

[9] A theory's predictive success (Selten, 1991; Harless and Camerer, 1994), its restrictiveness (Fudenberg et al., 2026), its completeness (Fudenberg et al., 2022), and its anomalies (Mullainathan and Rambachan, forthcoming) are all defined relative to this set.

[10] For example, take $\mathcal{F}_T$ a class of neural networks and D_T = "forecasts extrapolate the latest trend in realized outcomes."

4 The induction problem

4.1 Three goals of induction

What is the goal when we induce theories from data in some context? We think induction ought to produce theories with three properties: predictive accuracy, portability, and atomicity. As we will see next, words and binding are crucial to formalizing each of them.

Predictive accuracy. We obviously want theories to predict well in the context we induce them in. How do we formalize this? One might not have expected words to be important here, but Section 3 demonstrates that binding with words is the necessary first step. After binding turns $\mathcal{F}_T$ into $\mathcal{F}_T \circ B_{T,C}$ on the context's variables, there is still one extra step needed to measure a theory's predictive accuracy: we need to use the data to select a single function f^* from $\mathcal{F}_T \circ B_{T,C}$ with the highest R^2, which we can then evaluate directly. [11] Appendix B.1 gives the details.

Portability. We also want our theories to port to a wide range of other settings. [12] Why do we want portability? First, as discussed in the introduction, portability is the crucial property of a theory that distinguishes it from machine learning, and is what makes it useful when we encounter settings with very little or no data, as in macroeconomics. There is also a second, more unexpected benefit of wanting portable theories. Induction necessarily faces an underdetermination problem: there are always countless theories that fit the observed data equally well but have different implications elsewhere (Duhem, 1906; Quine, 1951; Goodman, 1955). This shows up in our framework too: for any function (e.g. $f = Y_{i,t} + \beta\,(Y_{i,t} - Y_{i,t-1})$ in the analyst context) there are many theories that bind to it (e.g. $D_T =$ "people extrapolate the latest trend when forecasting a firm's financial fundamentals," and extrapolative expectations, which says the same about any realized outcome). To do theory induction, we therefore need a tiebreaking rule between different, equally well-fitting theories, which we take to be maximizing portability. Why is this a reasonable criterion? We often think the theories that perform best out of sample are the portable ones. If Occam's razor is the idea that the theories that perform best out of sample are the simple ones, then this second, unnamed razor is the idea that the portable ones do. This principle already pervades how we do science across disciplines: in physics, for instance, we often want universal laws. This paper gives this thought a formal criterion (shown below).

How should we define portability? Binding once again proves helpful:

Definition 4 (Portability)**.** *Relative to a finite set of relevant contexts* $\mathcal{C}$,[13] *the portability of* T *is the number of contexts in* $\mathcal{C}$ *it binds to,*

$$\text{portability}(T) = \big|\{C \in \mathcal{C} : B_{T,C} \neq \emptyset\}\big|.$$

Note that what portability says is that a theory binds and makes predictions in other contexts, *not* that those predictions are necessarily accurate. This is intentional, and unavoidable for any theory we induce from data in some context C: as the problem of induction highlights (Hume, 1748; Goodman, 1955), there is no logical reason why patterns from one context should also hold in

[11] This captures the idea that we can use realized data to calibrate the free parameters within a theory.

[12] Ultimately we want the theory to predict well in those other settings too. But with data from only one context, for example analyst expectations, nothing can guarantee that; the most we can ask of the data we have is that the theory makes predictions in other settings.

[13] We assume finiteness just so that we can easily determine and rank the sizes of sets.

another. The best we can ask for is that the theories we induce at least bind and *make predictions* elsewhere.[14]

In this paper we are interested in induced theories relative to the collection $\mathcal{C}$ of contexts whose output $\mathcal{Y}_C$ is some expectation. Extrapolative expectations has a high portability relative to this $\mathcal{C}$, while a theory that says "firms with high accruals have their forecasts revised down" has much lower portability: it only really applies to financial forecasts.[15]

Atomicity. Finally, we want our theories to be simple, which in this paper we will refer to as *atomic*.[16] We will once again use words to formalize this: we define a theory as atomic if its description D_T states a single mechanism. Why do we restrict D_T rather than $\mathcal{F}_T$? Some complex $\mathcal{F}_T$ can be described very simply: constant-gain learning (Malmendier and Nagel, 2016) forms forecasts from a regression on the entire history of past outcomes, re-estimated every period with weights that decline on older observations, but its description, "forecasters learn by regression, discounting old observations," is a single mechanism. And some simple functional forms do not describe any economic mechanism: $\sin(x_1^*)^{\cos(x_2^*)}$ has no economic meaning, whatever x_1^* and x_2^* are. We cap the number of mechanisms in D_T at one just for simplicity, though allowing two or three mechanisms would work as well.

Which D_T are atomic? As with binding earlier in this section, there is no mathematical definition, because D_T is a verbal description. Nonetheless, it is a criterion that researchers can judge and typically agree on. Appendix B.3 lists examples of descriptions we do and do not consider atomic. Since we are interested in having algorithms induce theories, in this paper we will sometimes use LLMs to make this judgment.

4.2 Turning the goals into an algorithm

How can we use algorithms to produce theories that satisfy the three criteria above? A natural attempt is to find the best-fitting function (i.e., $\hat{m}$ of Section 2) and to turn it into a theory by finding some theory $(\mathcal{F}_T, D_T)$ that binds to it. Unfortunately, this will not produce any theory we like: the only way to do this is to attach some extremely long D_T that literally narrates the variables and operations of $\hat{m}$, such as "earnings at time t minus accruals at time t, all multiplied by the price change from $t-1$ to t, plus ..." This is neither atomic nor portable to any setting where the set of variables changes.

Since we are interested in inducing portable theories, the induction procedure should first search only over what we call *bindable functions*: the functions that some atomic theory with sufficient portability binds to,

$$\mathcal{F}_C^B = \bigcup_{\substack{T\,:\,T \text{ atomic,} \\ \text{portability}(T) \geq \underline{b}}} \mathcal{F}_T \circ B_{T,C},$$

where $\underline{b}$ is a minimum portability. Induction then searches over bindable functions,

$$f^* \in \arg\max\ R^2(f) \quad \text{s.t.} \quad f \in \mathcal{F}_C^B,$$

[14] Inducing theories from data in many contexts pushes this problem back rather than solving it: if we induce using contexts $C_1, \ldots, C_n$, there is no logical reason why any theory will predict well in the next context C_{n+1} we encounter.

[15] We note that defining portability relative to $\mathcal{C}$ is intentional: different researchers may have different sets of contexts they want to understand, and so we allow portability to be tailored to whichever settings a researcher cares about.

[16] We use a new word, atomic, rather than a word like "simple" that other researchers may have already attempted to formalize in different ways.

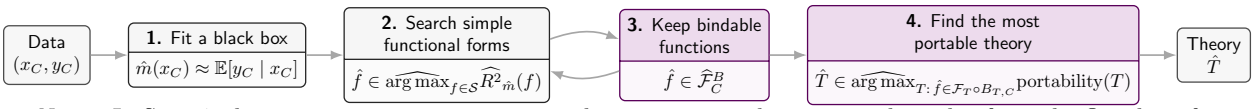
and then takes the most portable atomic theory that binds to it,

$$T^* \in \underset{\substack{T:T \text{ atomic},\\ f^* \in \mathcal{F}_T \circ B_{T,C}}}{\arg\max} \text{portability}(T).$$

The notion of bindable functions helps us understand when a function learned in one context can apply to another. If two contexts share the same input and output variables (e.g. analyst expectations before and after COVID), then any function learned in one can be applied directly to the other.[17] But when the input and output variables differ in two contexts (e.g. analyst expectations and macroeconomic forecasts), applying a fixed function directly is no longer meaningful. The functions that can still move to other contexts are the bindable ones: we can recover a theory from them, which can then be ported. In the example from our introduction, we intuitively see $\text{forecast}_t[\text{earnings}_{t+1}] = \text{earnings}_t + \beta\,(\text{earnings}_t - \text{earnings}_{t-1})$ as portable precisely because it is a bindable function.

Our goal in this section is to develop a general and modular induction algorithm that implements the steps above. We want it to be general because we hope it can work in any setting, and we want it to be modular, so that future researchers can improve or adapt the inputs into the algorithm based on their preferences. For each module, we lay out the general principles behind it and explain our specific implementation for analyst forecasts; Section 4.7 collects the possible improvements and adaptations.

We lay out the modules in Figure 4, which consists of four steps.[18] (One of the main themes of Section 3 is that binding and words are important; we mark these parts in purple in this section.) The first is to fit a black-box model that captures all the predictable variation, as we did in Section 2. Then we specify a search space $\mathcal{S}$ of simple functional forms that bindable functions typically take, and search over it for the functions that explain the black box as well as possible. Next we need a method that retains only the bindable functions; this turns out to be one of the most important steps, and also what is most novel relative to existing methods. Finally, we find the most portable theory that binds to the function, and iterate the procedure to produce many theories.



Notes. In Step 4, the maximization is, more precisely, over atomic theories; we drop this from the flowchart for readability.

Figure 4: The four steps of the induction algorithm.

4.3 Step 1: fitting a black-box algorithm

The goal of Step 1 is a black-box algorithm $\hat{m}$ that maximizes the R^2 for predicting y_C from x_C. We already did this in Section 2.

One might wonder why we need this step at all. As written, Step 2 searches for the functions that best explain $\hat{m}$. Why not have Step 2 maximize the R^2 for explaining y_C directly? We could, but targeting $\hat{m}$ improves performance substantially by ensuring we do not overfit during search. In particular, the search algorithms of Step 2 sometimes lack methods for ensuring out-of-sample performance, while black-box algorithms like neural nets already come with techniques (e.g. weight

[17] This is transfer under distribution shift, and it is studied in Andrews et al. (2025) (though even in this case, applying the same function across different distributions carries no guarantee that it should continue to predict well).

[18] This flowchart tells us how to induce a single theory; we briefly discuss how to modify it to induce many theories in Section 4.7.

decay, dropout) to ensure we do not overfit. Therefore we split the task into two steps: first we fit $\hat{m}$ to maximize out-of-sample performance, then the search finds the f that best predicts $\hat{m}$. We note that overfitting in Step 2 becomes a much smaller problem when targeting $\hat{m}$: it has less variance than y_C, and because it is a function, we can evaluate it at any x_C rather than just those already in our dataset.[19] Full details about how we evaluate a function's R^2 for explaining $\hat{m}$ are in Appendix A.3.[20]

4.4 Step 2: symbolic regression

Our goal is to specify a search space $\mathcal{S}$ with functions that predict well and are also bindable. There is a tension in choosing $\mathcal{S}$: it has to be rich enough to contain functions that predict well, but if it is too rich, like the class of neural nets, almost nothing it will return will be bindable.

As mentioned in Section 3, the shape of a functional form does not by itself determine whether it is bindable; that also depends on what its variables mean. It is nonetheless the case that bindable functions tend to have a similar mathematical structure, as in Figure 5: they generally consist of a few variables combined using a few operations. How do we search over these simple formulas?

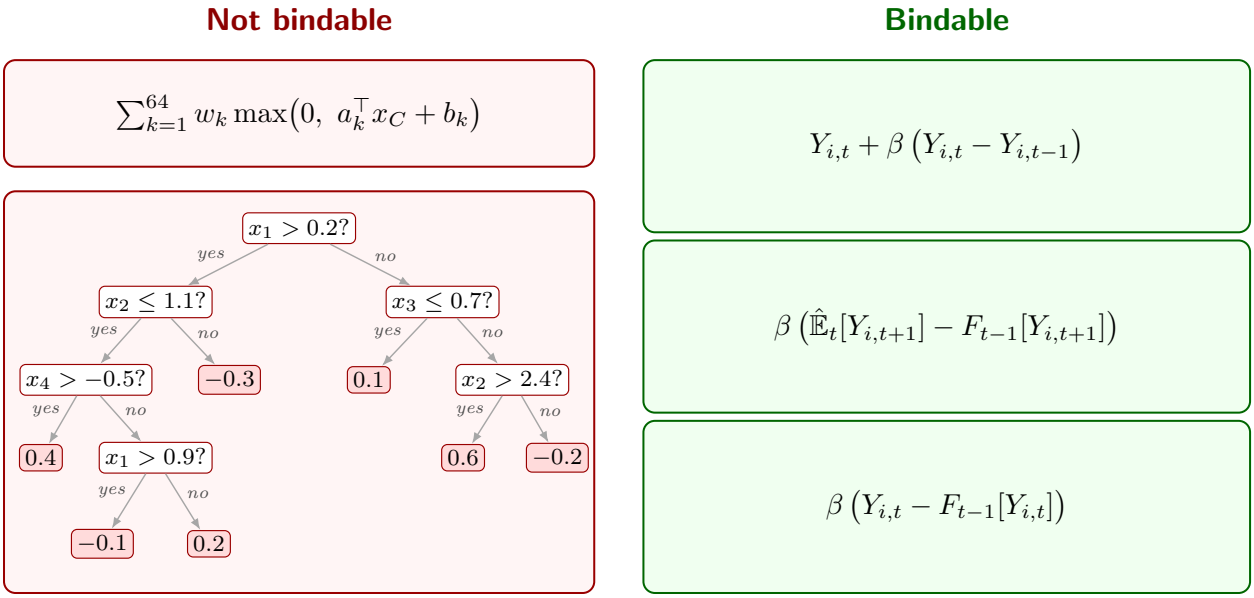


Figure 5: The mathematical structure of bindable and unbindable functions.

A common method for this search is *symbolic regression*. It builds formulas from a set of input variables and a set of operations, and uses a genetic algorithm to search over formulas that predict well, subject to a size constraint on the number of variables and operations (Figure 6). The genetic algorithm starts from a population of formulas, scores each by its R^2 for explaining $\hat{m}$, keeps the best, and builds the next generation from them by swapping variables or operations (mutation), trading subcomponents of two formulas (crossover), and adding new random formulas (immigrants). Repeated over many generations, the best formulas improve. Appendix C.2 describes the genetic algorithm in full.

[19] In practice, evaluating $\hat{m}$ at the x_C's in our data already works well, so we do not generate any additional x_C's.

[20] Our evaluations in Section 5 still use the R^2 for explaining y_C: we do not want to introduce approximation error from $\hat{m}$ when evaluating.

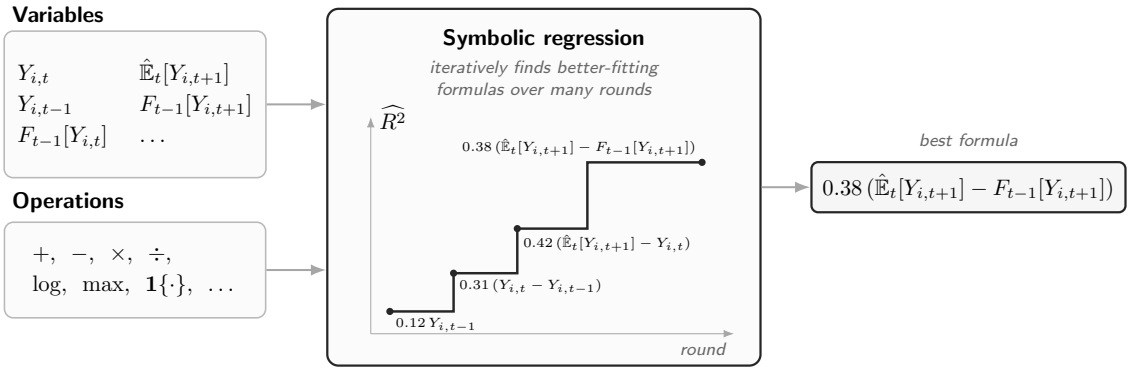


Figure 6: Symbolic regression builds formulas from the variables and operations.

Many machine-learning methods have made their way into economics, but symbolic regression is probably less familiar to most readers. Perhaps surprisingly, it is actually quite an old idea: Koza (1990) introduced the term, and an early application was actually in economics: Koza (1994) recovered the exchange equation $MV = PQ$ from US macroeconomic data. The method then took a back seat during the deep-learning revolution: on pure prediction tasks, black-box algorithms like neural nets tended to perform best (LeCun et al., 2015). In recent years it has come back across the sciences as an interpretable alternative to black-box algorithms (Schmidt and Lipson, 2009; Udrescu and Tegmark, 2020; Cranmer et al., 2020). For example, Lemos et al. (2023) recover Newton's law of gravitation by running symbolic regression on solar-system trajectories.

Choosing the variables and operations. To use symbolic regression, we first have to specify the variables and operations that go into it. Our general principle is to include the variables and operations that have some chance of showing up in a bindable function in this context, and to leave out those that essentially never do.

To be concrete, start with the operations. We include the standard ones that show up in economic theories: the arithmetic operations, log, max, sign and threshold devices. A full list is in Appendix C.1. We leave out operations like sine and cosine, which one might want in physics but which tend not to show up in economic theories.[21]

For the variables, a starting point is the 9,606 columns used to build $\hat{m}$ (Table A1). One big thing is missing from these columns: concepts like rational expectations. Though one could approximate rational expectations as some combination of past earnings and financial information, it likely involves a very large number of variables (exceeding the size constraint of symbolic regression), and it would be hard to recognize it from the functional form itself. So to allow symbolic regression to use these concepts, we add them directly as variables, by building rational-expectations proxies such as $\hat{\mathbb{E}}_t[Y_{i,t+1}]$, a machine-learning estimate of next quarter's earnings (Section 2.2). We also add in the predictions of known theories, to build on the literature.

We also take columns out. Most of the 9,606 columns are firm-level financial variables, such as accruals and sales, which are not naturally portable outside of firm-specific contexts. We therefore take out all financial variables aside from earnings, prices, and returns.[22][23] We also take out the

[21] In principle nothing stops a theory of expectations from using sine or cosine, but in practice it is hard to imagine one. And looking ahead to Section 4.5, if an LLM looking at a function cannot name a mechanism behind it, the algorithm cannot proceed, even if a theory exists in principle.

[22] Prices and returns do have potential analogs in other settings, e.g. in macro it could be the aggregate stock market price/return, or some measure of consumer sentiment.

[23] The information in accruals, sales and the other dropped columns is not entirely lost: $\hat{\mathbb{E}}_t[Y_{i,t+1}]$ is trained on

columns with too many missing rows, leaving us with 521 variables. Full implementation details are in Appendix C.1.

We also want to note that our algorithm does not automatically determine what the variables and operations ought to be. This is intentional: in principle we could further automate the procedure by having an LLM choose the variables and operations. But our goal is not full automation but rather the amplification of researcher creativity, and forcing an LLM to select inputs needlessly restricts the capabilities of our algorithm. More generally, almost all machine-learning algorithms depend on researcher input: neural-net performance depends on the quality of the features constructed, and even LLM outputs depend on the quality of prompts.

4.5 Step 3: restricting search to bindable functions

Symbolic regression on its own often returns functions that predict well but are not bindable (Section 5.5 shows a concrete example). We point out that there are many reasons why a relatively simple functional form might not be bindable. For example, even the bindability of $x_1 - x_2$ depends on the semantic meaning of the two variables: if x_1 is a firm's earnings and x_2 is its accruals, $x_1 - x_2$ is atomic but is not portable to settings without firms. If x_1 is the number of analysts who revised their forecast up two months ago and x_2 is last month's stock price, then $x_1 - x_2$ combines different units and is also unbindable. Since one cannot proceed with the final step (finding the most portable theory) if the output of search is unbindable, we need a way of restricting search to just the bindable functions.

The bindability checker. We need an algorithm that checks whether a function f is bindable. Recall from Section 4.2 that f is bindable if some atomic theory with sufficient portability binds to it. There are two challenges. First, we cannot just enumerate all such theories. Second, there is no mathematical formula for binding, so any implementation must be approximate.

We solve the first problem by having an LLM, the *proposer*, read f and propose a small set of descriptions that might explain it. We solve the second problem with another LLM, the *binder*: it reads each proposed description together with our context's description D_C and, without seeing the original f, writes down the class of functions the description implies in our context. We deem f bindable if one of these classes contains it (Figure 7).[24] Appendix D gives both prompts.

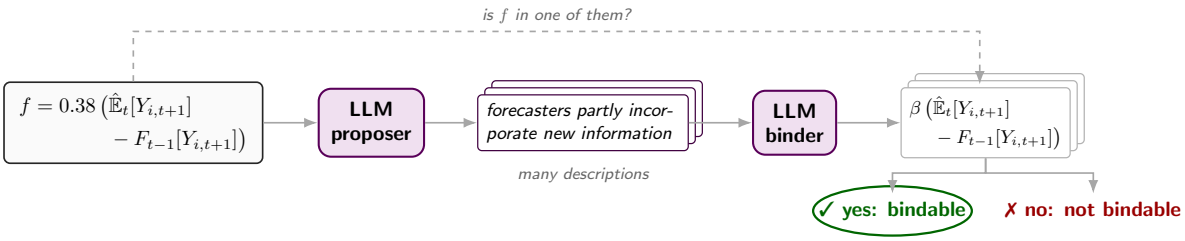


Figure 7: The bindability checker. The LLM proposer reads f and writes many descriptions that might explain it; the LLM binder reads each description and writes the class of functions it implies in our context. f is bindable if one of these classes contains it.

them, and it is part of many bindable functions.

[24] We note that the proposer LLM is not proposing theories in the sense of Section 3: it returns only descriptions D, not full theories $(\mathcal{F}_T, D_T)$. If it also proposed $\mathcal{F}_T$, it could pass too many f's by copying their functional form into $\mathcal{F}_T$ and narrating their variables in D_T, so that binding returns f by construction. With a description alone this kind of cheating is harder: the binder never sees f, so f comes back only if the words pin down its form.

One challenge remains: the search checks a very large number of candidate functions, and the two LLM calls are slow and expensive, so we cannot run them on every function the search visits. So inside the search we use a fast approximation instead: a random forest algorithm trained to predict whether the LLM procedure deems a function bindable, and symbolic regression keeps only functions that the random forest deems to have a greater than 1/2 chance of being bindable. Then, at the end of the search, we use the LLM proposer and binder as a way to "project" symbolic regression's outputs onto the set of bindable functions. Full details of the entire process are given in Appendix C.3.

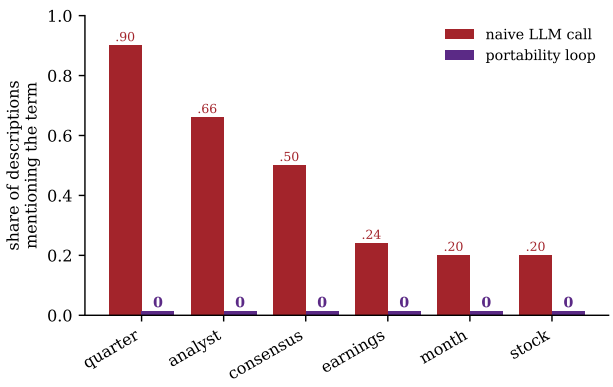
Researcher selection. At the end, the search returns a set of (likely) bindable functions. The final, optional step is that we, the researchers, can accept or reject what comes out. In particular, we sometimes reject the best-fitting function because the LLM either mistakenly returns an unbindable function or also returns a much simpler function with almost the same R^2, which we accept instead. [25] This step is not strictly necessary: it can be removed if one is willing to accept unbindable functions occasionally being returned.

4.6 Step 4: the portability loop

At this point, our symbolic regression with the bindability checker has delivered some $\hat{f}$ that explains forecast revisions. We first add back $F_{t-1}[Y_{i,t+1}]$ to $\hat{f}$, since we want a theory of forecasts. Our last major challenge is then to convert $F_{t-1}[Y_{i,t+1}] + \hat{f}$ into a maximally portable theory.

A natural thing to do is just to prompt an LLM to propose a theory $(\mathcal{F}_T, D_T)$ that binds to $\hat{f}$. We do exactly that (prompt in Figure A6) and quickly encounter a problem: while the LLM's descriptions are accurate, they are far too specific. Figure 8 shows how often the proposed D_T for the functions delivered by the earlier steps of our algorithm (Table A14) mention analyst-specific terms. Any theory that mentions "quarters" will not port to other contexts with different time units.

[25] To evaluate our algorithm, we think it is important to keep the researcher's job in this loop constrained to just selection. Giving the researcher more freedom — say, handing them the search output and asking them to return their own theory — would also be a valid induction procedure, but it may reflect the researcher's preexisting insight, at which point it is not clear how much of the final theory is due to our algorithm being helpful versus the researcher being smart.



Three descriptions from the naive LLM call

- "Sticky updating: **analysts** keep last **quarter's** forecast of the target and add a fraction of an exponentially decaying average of recent gaps between fresh model predictions and prior **consensus**."
- "**Analysts** keep last **quarter's** forecast of the target and shade it down in proportion to how volatile recent fundamental news has been, measured by the dispersion of statistical-forecast revisions."
- "**Analysts** keep last **quarter's** forecast of the current target **quarter**, then shade it down by a small fixed fraction of the earlier **consensus** forecast for the just-realized **quarter**."

Notes. Left: the share of the LLM's one-sentence descriptions of our ten functions (five samples each, 50 descriptions per prompt) that mention each analyst-specific term, for the naive LLM call (red) and the final round of the portability loop (purple). No description from the portability loop mentions any of the terms. Right: three descriptions from the naive LLM call, one per function, with the analyst-setting-specific terms in red.

Figure 8: Share of descriptions mentioning analyst-specific terms, from a naive LLM call and from the portability loop.

How do we fix this problem? First, note that "propose a theory" is not itself a clear instruction: as we discussed in Section 4.1, countless theories bind to any given function. The LLM implicitly adjudicates between them by giving the most specific theories, perhaps to limit the chance of a hallucination. But we want a different tiebreaker: maximize portability. So the first fix is to define portability explicitly in the prompt and tell the LLM to maximize it (Figure A7). Second, it helps to run a "portability loop", shown in Figure 9, where we specify some other set of contexts we want our theory to apply to and ask the LLM to iteratively produce theories with higher portability.

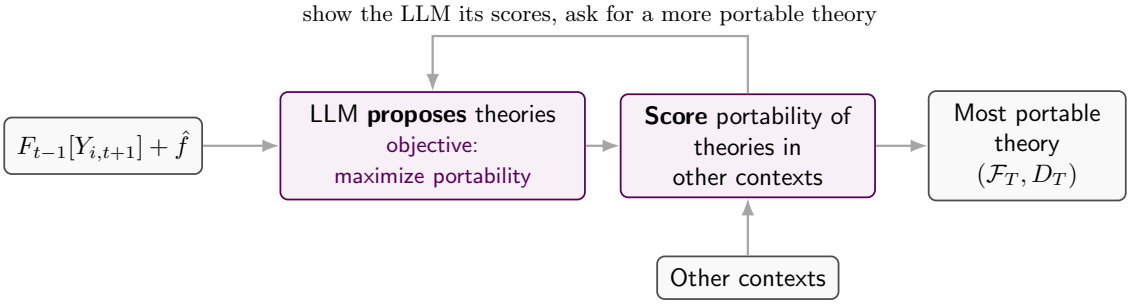


Figure 9: The portability loop: an LLM is told to iteratively generate more and more portable theories.

We implement $\mathcal{C}$ with ten expectations contexts from the economics literature (the full list is in Appendix E), and run three rounds of the loop where the LLM proposes 5 theories each time. After doing this, setting-specific terms like "quarter" or "analyst" go away (Figure 8).

4.7 Summary, extensions and limits

In summary, our induction algorithm is shown in Figure 10. As written, this only delivers a single theory; what if we want more? Simply running our existing algorithm repeatedly is insufficient:

every run produces very similar results. So to get a collection of diverse theories, we set the objective for each additional theory to be its marginal R^2 over and above the theories already found.[26]

We want to reiterate that the algorithm is both improvable and adaptable, but also that it has limitations: there are things it is *not* capable of. Table 1 lists, for each step, where better tools would help (e.g. a higher-R^2 $\hat{m}$ or a more accurate bindability checker), where researchers can tailor the algorithm to their own interests (e.g. the variables and operations, or the contexts portability is scored on), and what lies beyond what our framework can currently express (e.g. porting a theory to a different type of outcome variable).

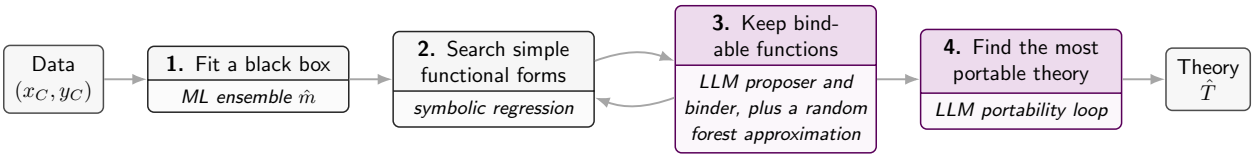


Figure 10: Our implementation of each step of the induction algorithm.

	1. Black box	**2. Symbolic regression**	**3. Keep bindable functions**	**4. Portability loop**
Improvable	A higher-R^2 $\hat{m}$	The variables and operations	A more accurate bindability checker	A proposer whose theories are more portable
Adaptable	—	The variables and operations	Properties of f an algorithm can check	The contexts portability is scored on
Needs new conceptual tools	—	—	Properties of f an algorithm cannot check	Porting to a different type of outcome variable

Notes. Steps as in Figure 4. *Improvable*: better tools for the same problem. *Adaptable*: researcher choices our framework already expresses, each a choice of search inputs, a property of f that can be checked by an algorithm, or a choice of contexts. *Needs new conceptual tools*: wishes our framework cannot yet express.

Table 1: What in our algorithm can be improved, adapted, or requires new conceptual tools.

Improvable. There are two main ways in which the implementation of our algorithm can be improved.

Better machine-learning algorithms. Several steps are clear-cut machine-learning tasks, and improving performance on any of them would improve the whole algorithm; our current performance represents the floor on what is achievable. In Step 1, if better machine-learning algorithms are developed, the black box $\hat{m}$ can reach a higher R^2. In Step 2, symbolic regression performs better if future work develops better search algorithms. In Step 3, one could get more accurate LLM bindability checks, perhaps by fine-tuning their verdicts on human labels. In Step 4, one of the biggest limitations of using frontier LLMs in the portability loop is that the descriptions they produce are generally fairly specific and do not have a great deal of variety. This is not a huge

[26] In practice we made two small deviations from this protocol. First, on some rounds we selected more than one theory, when there were qualitatively different theories that both had good predictive power. Second, in the final round, instead of another run to maximize marginal R^2, we reran the previous rounds under different random seeds and combined the results.

problem for the functions delivered by our procedure, which are new but can be described using language that is already familiar to the literature. Producing truly novel theories, far outside of the range of what we typically write down, may require building language models that can themselves produce a greater variety of descriptions.

Better domain-specific knowledge. A second set of improvements comes from embedding domain-specific expertise into the algorithm's inputs. One way is to bring in more data the researcher thinks is relevant (e.g. unstructured data such as earnings calls or 10-K filings); this expands the range of theories our algorithm can produce. Improvements can also come from encoding domain knowledge into the variables and operations. We saw this earlier in Section 4.4: had we not added in rational-expectations proxies like $\hat{\mathbb{E}}_t[Y_{i,t+1}]$, the highest-$R^2$ theories we will see in Section 5 would not have been attainable.

Adaptable. The researcher can also customize the theories the algorithm returns, and there are a few ways to do this. The most obvious is to expand the bindability check of Step 3: this check can be expanded to include any other property that can be checked by an algorithm. For example, if a researcher wants to only induce theories related to attention, this step could reject any function where the LLM description of f does not include the word "attention". We can also ask that the function reproduce any stylized fact that can be checked by plugging f into the data. For instance, we can restrict f to only those functions that capture overreaction, as measured by a negative coefficient in a regression of forecast errors on forecast revisions (as in Coibion and Gorodnichenko 2015). There are also other ways to customize what the algorithm returns. One is through the variables and operations: a researcher who wants expectations theories that do not use past forecasts is free to remove all of those variables from the search. We can also specify the kind of settings we want the theory to port to, by changing the other contexts the portability loop scores on.

Needs new conceptual tools. There are things one might want in induced theories that our framework cannot yet accommodate. These are the properties of functions or descriptions that we do not yet know how to check algorithmically, because we lack good formalizations of them. For instance, we might only want to search for functions that are the product of some psychological microfoundation that applies beyond expectations; at the moment we do not have any way of checking this, because we do not have a clear definition of "microfoundation." Our current framework also cannot check properties of f that rely on first deriving testable implications of f for a different type of outcome. One example is whether f predicts momentum and long-run reversal when embedded in an equilibrium asset-pricing model.[27] That is not to say that this must remain infeasible forever; future work may find a way to formalize microfoundations or derivations in a way that makes these properties checkable by an algorithm, in a similar way to how this paper makes portability checkable. We hope future work will develop extensions of our framework to accommodate desiderata like these.

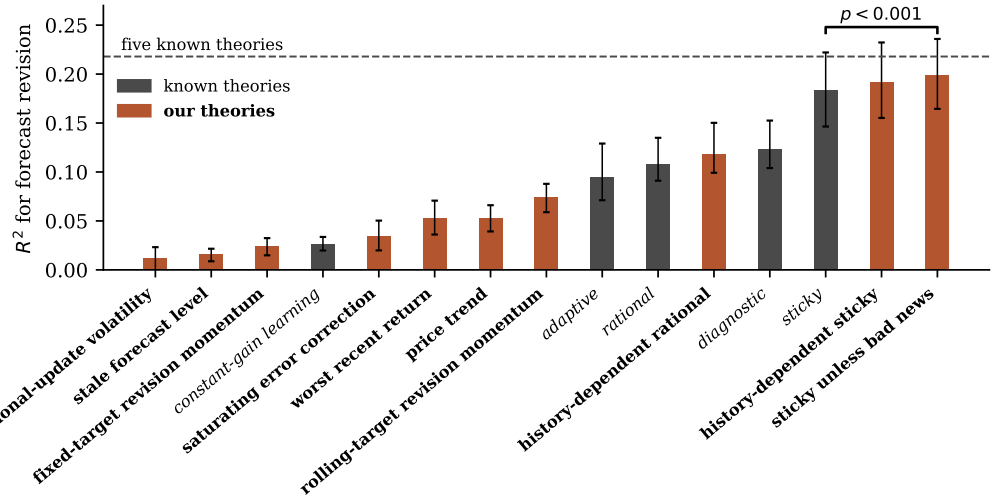
[27] Note that this is qualitatively different from checking whether f produces overreaction as measured by a regression of forecast errors on forecast revisions: the former requires doing a derivation in an equilibrium model, whereas the latter takes the data as given and can be checked by applying f to them and running a regression (showing that f produces overreaction in general, beyond a given dataset, would again require a derivation).

5 Inducing theories of expectations

The main finding of Section 2 was that a black-box machine-learning model trained to predict analysts' forecast revisions strongly outperforms the known theories in the literature. We then developed a framework (Section 3) and an algorithm (Section 4) to induce theories. Our algorithm is able to induce not only one but ten theories of expectations, each of which adds marginal R^2 over the previous ones; full descriptions of all of them are in Table A15.

5.1 Predictive accuracy of the induced theories

We first ask how well our theories predict forecast revisions. In Figure 11 we take Figure 1 from Section 2 and add our 10 theories, shown in orange. Notably, our best theory, which we study in more detail in Section 5.2, has a higher R^2 than any known theory. Its R^2 for forecast revisions is 19.8%, against 18.3% for sticky expectations, the best known theory (difference is 1.6 percentage points, 95% CI [1.0, 2.2]). Figure A14 in the appendix shows how well the same induced theories explain the machine-learning benchmark $\hat{m}$, and Tables A14 and A16 give the expressions and fitted coefficients of the ten theories when bound to quarterly forecasts.

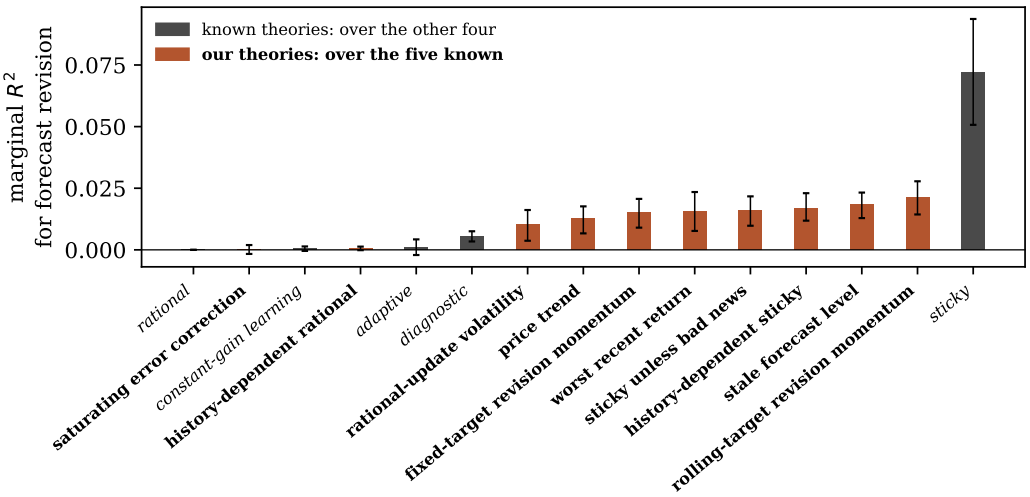


Notes. The target is the forecast revision $\Delta F_{i,t}$. Each theory's coefficients, including the internal constants of our theories, are fit on folds 1–4 and R^2 is reported on fold 5, with 95% confidence intervals from a firm-level bootstrap (200 draws). Note that the confidence intervals here come from bootstrapping fold 5 only, which is why they are wider than those of Figure 1, which bootstraps from all five folds. The short description of each induced theory is written by the authors (not the induction algorithm).

Figure 11: Out-of-sample R^2 of the known theories of Figure 1 (black) and our 10 induced theories (orange).

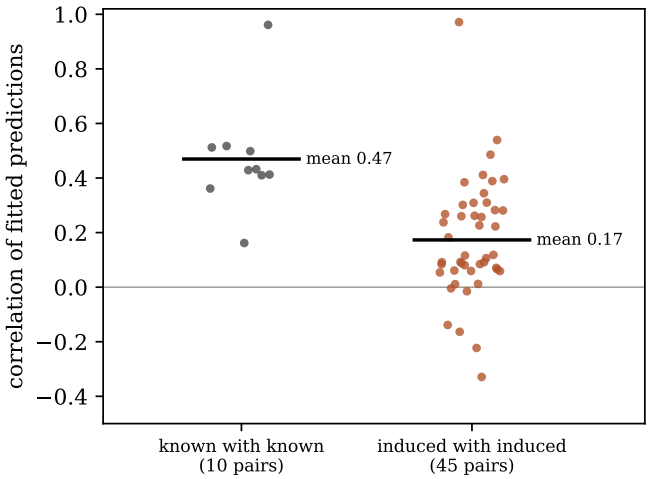
We note that many of our theories do not outperform the R^2 of the known theories. This is a byproduct of our search procedure: as described in Section 4.7, in order to get a diverse set of theories, we induce each additional theory by how much additional R^2 it explains over and above the theories already induced, rather than the R^2 it attains on its own. To benchmark these marginal contributions, Figure 12 shows the marginal R^2 of each of our 10 theories over the five known theories and of each known theory over the other four; 8 of our 10 theories have a marginal R^2 whose 95% confidence interval lies above zero, against 2 of the five known theories. Our resulting theories are diverse; one can see this from reading the descriptions in Table 2, and

we can also quantify it explicitly: the average correlation between two of our induced theories is 0.17, compared to 0.47 between the known theories (Figure 13).



Notes. The target is the forecast revision $\Delta F_{i,t}$. For each of our theories, the marginal R^2 is the R^2 of the five known theories plus that theory minus the R^2 of the five known theories; for each known theory, the R^2 of all five minus the R^2 of the other four. Rational expectations gets 0 because it is nested by diagnostic expectations. Coefficients, including the internal constants of our theories, are fit on folds 1–4 and R^2 is reported on fold 5, with 95% confidence intervals from a paired firm-level bootstrap of fold 5 (200 draws).

Figure 12: Marginal R^2 of each theory on quarterly forecasts: ours over the five known theories, each known theory over the other four.

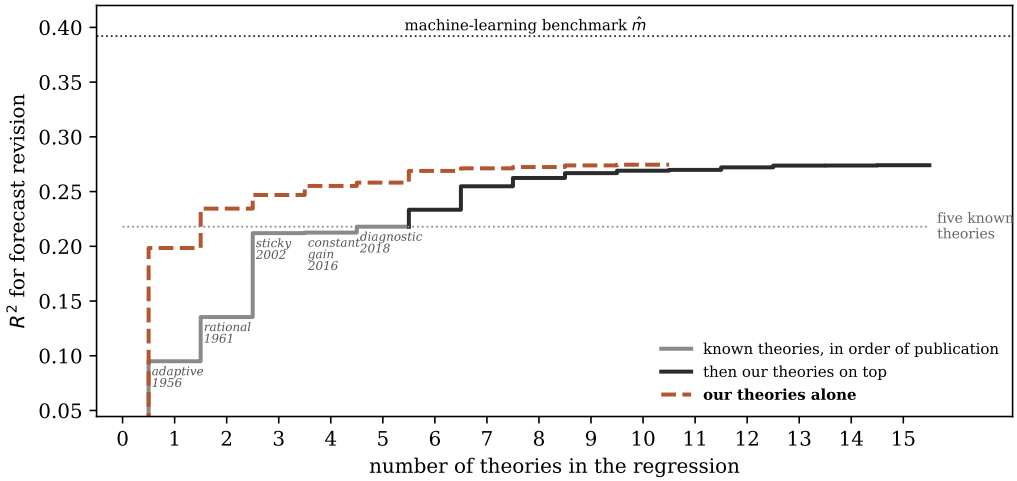


Notes. Each dot is the correlation between either two known theories ($5 \times 4/2 = 10$ pairs) or two induced theories ($10 \times 9/2 = 45$ pairs). The black line is the group mean. The two pairs above 0.9 are rational and diagnostic expectations (diagnostic nests rational) and history-dependent sticky and sticky (unless there's bad news) expectations (the second nests the first).

Figure 13: Our induced theories are less pairwise correlated than known theories.

Finally, we document how much we learn collectively from the theories we induce, in Figure 14. We first note that it is hard to add R^2 over and above what is known. The grey line adds the

five known theories one at a time in the order they were published[28] (each time fitting one linear regression of the forecast revision on all the theories added so far). The earliest theories make a lot of progress: adaptive and rational expectations together reach an R^2 of 13.5%, and sticky expectations brings it to 21.2%. Progress then levels off: constant-gain learning and diagnostic expectations together add only 0.6%. From this, one might expect new theories to add little marginal R^2 as well. This is not the case: adding our 10 theories on top of the known theories (the black line) still makes substantial progress, and all 10 together close 32.3% of the gap between the five known theories and our machine-learning predictor $\hat{m}$.[29] We can also start from scratch instead of adding our theories to what is known, as in the orange line. Two of our theories alone already predict forecast revisions better than the five known theories together (23.4% against 21.8%), and all 10 without the known theories close 32.5% of the gap, better than if we included the known theories (32.3%).[30] [31] Figure A15 in the appendix shows the same panel with the ML benchmark as the target.



Notes. The target is the forecast revision $\Delta F_{i,t}$. Each step is one linear regression of the forecast revision on all the theories added so far, fit on folds 1–4 and scored on fold 5. The grey line adds the five known theories in order of publication. The black line then adds our theories in greedy order on top of the five known theories; the orange line adds our theories in greedy order with no known theory. The orders are in Table A17 in the appendix. At each step the constants inside our theories are refit jointly on folds 1–4. The dotted lines are the five known theories together and the ML benchmark $\hat{m}$.

Figure 14: R^2 as theories are added one at a time: the five known theories in order of publication, then ours on top of them or on their own.

What do we make of the 32.5% of the gap being closed? As the small additions from constant-gain learning and diagnostic expectations in Figure 14 show, getting marginal R^2 over and above what is known is hard, so we take this to be substantial progress. It is by no means the end of

[28] Constant-gain learning has several specifications; like Afrouzi et al. (2023), we aim to follow the specification of Malmendier and Nagel (2016).

[29] We add our theories in a greedy order: the first theory added is the one that, together with the five known theories, gives the highest R^2; next, the theory that, together with the five known theories and the first, gives the highest R^2, and so on. The orange line uses the same rule starting from no theory.

[30] This is possible because the R^2 is out of sample: a smaller collection of theories can predict better out of sample than a larger one.

[31] We do encode the known theories in the columns our search can use, so it is perhaps not surprising that our theories incorporate what is in the known theories.

the story, though: we hope others will improve our induction procedure in the ways outlined in Section 4.7 to induce even better theories.

5.2 The highest R^2 theory: sticky (unless there's bad news) expectations

What are the 10 theories we induce? We go over one of them in detail here and a few more in Section 5.3; the full list of all 10 is in Table A15. The theory we go through in detail is the one with the highest R^2, a generalization of sticky expectations which we call *sticky (unless there's bad news) expectations*.[32] Figure 15 shows the theory $(\mathcal{F}_T, D_T)$ that our algorithm of Section 4 returns, what each of its variables binds to in our context, and its fitted values.[33] A walkthrough of how our search found this theory is in Appendix F.2.

Theory $T = (\mathcal{F}_T, D_T)$

Forecasters move their previous forecast partway toward an objective forecast. They move further after a large disappointing surprise and also react to a history of gaps.

$$\mathcal{F}_T = \left\{ F_{(t-1)}[Y_{(t+1)}] + \beta \left(1 + \delta\, \mathbf{1}\{e_t \le -\kappa\}\right) \frac{G_0 + \gamma\, G_1 + \gamma^2\, G_2 + \cdots}{1 + \gamma + \gamma^2 + \cdots} \;:\; \beta > 0,\ \gamma \in [0,1],\ \delta, \kappa \in \mathbb{R} \right\}$$

The theory's words (D_T)	**Bound in quarterly analyst EPS**
$F_t[Y_{(t+1)}]$ Forecast of next period's value.	$F_t[Y_{i,t+1}]$, forecast of next quarter's EPS
$F_{(t-1)}[Y_{(t+1)}]$ Forecast of next period's value, one period earlier.	$F_{t-1}[Y_{i,t+1}]$, last quarter's forecast of next quarter's EPS
$G_j, j = 0, \ldots, J$ The gap j periods ago between an objective forecast and the forecaster's own lagged forecast.	$\hat{\mathbb{E}}_{t-j}\left[Y_{i,t-j+1}\right] - F_{t-j-1}[Y_{i,t-j+1}]$ $(j = 0, \ldots, 12)$, the ML forecast minus the earlier consensus, j quarters ago
e_t The latest forecast error.	$Y_{i,t} - F_{t-1}[Y_{i,t}]$, the most recent EPS surprise

Parameter	Role	Estimate on quarterly analyst EPS
β	Share of the information gap closed every period	0.314 [0.258, 0.381]
γ	Weight on older information gaps	0.245 [0.185, 0.309]
δ	Bad news update strength boost	0.54 [0.39, 0.81]
κ	Bad news threshold	0.050 [0.010, 0.454]

Figure 15: Sticky (unless there's bad news) expectations, its binding to quarterly analyst EPS forecasts, and estimated coefficients.

[32] One might ask whether D_T is atomic, given that it contains a few different components. It is: the bad-news term multiplies both the response to the current gap and its dependence on past gaps, so the theory cannot be split into separate additive parts.

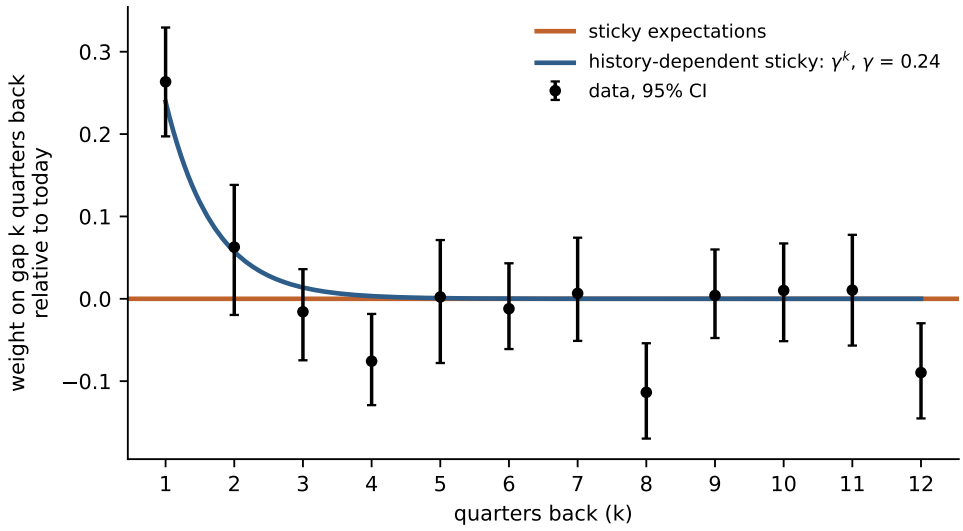
[33] We rewrite the summary and the variable descriptions of D_T in plainer words; Appendix F.2 gives them as the algorithm returned them. We also rename the constants, to make for easier reading: gain, rho, extra_gain and theta become β, γ, $\beta\delta$ and $-\kappa$. These coefficients, as well as all of the coefficients reported in this section, are fit on fold 5 of the training+validation data. In practice, 21.4% of forecast errors fall below the bad news threshold of $-\kappa = -5$ cents.

What's new in our theory. The theory captures three phenomena. The first is that expectations are sticky: the coefficient β is positive and below one, so forecasters move only part way to the objective model estimate. This is already a well-known fact in the literature (Coibion and Gorodnichenko, 2015; Bouchaud et al., 2019). ($\gamma = 0$ and $\delta = 0$ correspond to sticky expectations, $F_t[Y_{t+1}] = F_{t-1}[Y_{t+1}] + \beta\big(\mathbb{E}_t[Y_{t+1}] - F_{t-1}[Y_{t+1}]\big)$, where we relax the $\mathbb{E}_t$ term to a generic "objective model estimate".)

The second phenomenon is history dependence: the update depends not only on G_0, but also $G_1, G_2, \ldots$. That is, information that was outstanding in the past, G_k, continues to predict today's forecast update over and above any contribution it has to today's outstanding information G_0.[34] With $\delta = 0$ and $\gamma \in [0, 1]$ the theory is *history-dependent sticky expectations*, which our procedure also returns as a theory of its own (Appendix F.3). To see the effect of history clearly, Figure 16 regresses $\Delta F_{i,t}$ on the outstanding information from k periods ago (with the bad news gate off), $g_{i,t-k} = \hat{\mathbb{E}}_{t-k}\big[Y_{i,t-k+1}\big] - F_{t-k-1}\big[Y_{i,t-k+1}\big]$ for $k = 0, 1, \ldots, 12$,

$$\Delta F_{i,t} \;=\; a + \sum_{k=0}^{12} b_k \, g_{i,t-k} + u_{i,t},$$

and plots each coefficient relative to b_0, b_k/b_0. Plain sticky expectations says only $g_{i,t}$ enters, so $b_k/b_0 = 0$ for every $k \geq 1$; history-dependent sticky expectations says $b_k/b_0 = \gamma^k$. The geometric decay of history-dependent sticky expectations does a pretty good job of capturing the coefficients b_k/b_0, aside from $k = 4, 8, 12$ (the same quarter one, two and three years ago), which are, interestingly, *negative*. This may reflect some difference based on seasonality.

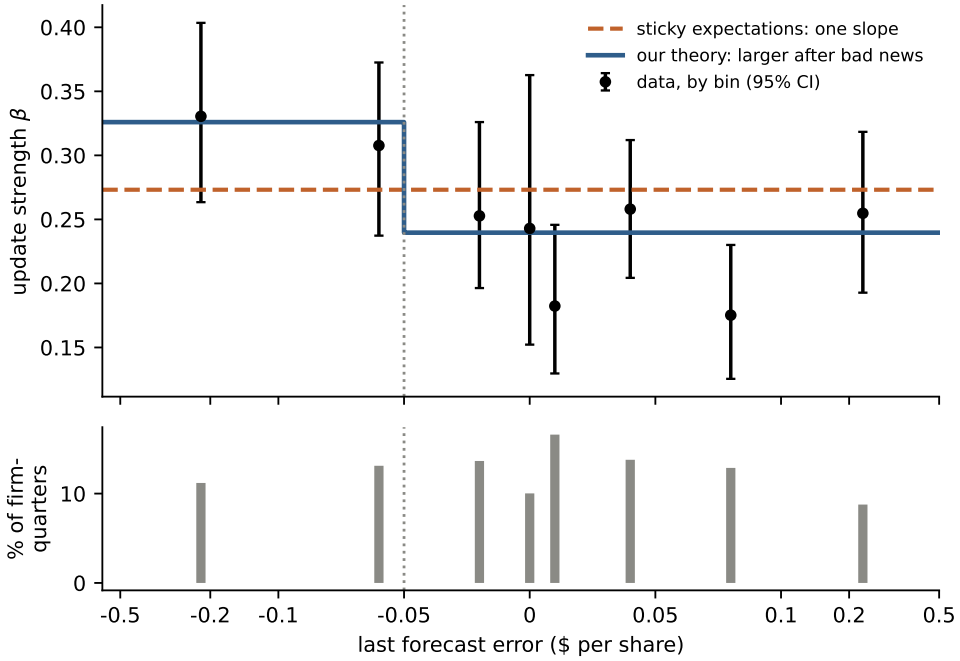


Notes. One regression of the forecast revision on the gaps, on fold 5 (no bad news gate): $\Delta F_{i,t} = a + \sum_{k=0}^{12} b_k \, g_{i,t-k} + u_{i,t}$. Dots are b_k/b_0 with 95% confidence intervals from a firm-block bootstrap (500 draws, each rerunning the regression). The curve is γ^k with $\gamma = 0.239$, history-dependent sticky expectations fit directly on $\Delta F_{i,t}$ (Appendix F.3); the flat line is plain sticky expectations.

Figure 16: Gaps from previous quarters still move today's revision.

[34] A reader may wonder whether this is just another way of saying that forecast revisions are positively autocorrelated, which is well documented. Our new theory says more than that: positive autocorrelation of revisions is *implied by* plain sticky expectations with no weighted-history term at all. But under plain stickiness the current gap is a sufficient statistic — once G_0 is known, earlier gaps carry no additional predictive weight.

The third phenomenon is the bad news gate, which turns on when the most recent forecast error e_t is sufficiently negative — i.e. realized outcomes are well below expectations — in which case the response increases from β to $\beta(1+\delta)$. This means that revisions are sticky in general, but are less sticky when realized outcomes are sufficiently below expectations.[35] A natural interpretation of this could be that low realized outcomes also cause forecasters to pay more attention to other information. Figure 17 isolates the bad news gate. Setting the history of gaps aside, it asks how much of today's information gap forecasters close as a function of how bad the most recent earnings surprise is. If revisions were equally sticky everywhere, the dots would lie on the dashed line. Instead, it appears that the response is higher after negative earnings surprises, though the relationship is not quite the step function that our theory predicts.



Notes. Top: within each bin b of the last forecast error $e_{i,t}$, the update strength β_b, the slope of $\Delta F_{i,t} = a_b + \beta_b\, g_{i,t} + u_{i,t}$ (today's gap only, no history dependence), fold 5, with 95% confidence intervals from a firm-block bootstrap (500 draws). Errors come in whole cents; $e_{i,t} = 0$ is its own bin and, moving out from zero, whole-cent values are pooled until a bin holds about an eighth of the sample. The dashed line is sticky expectations, one slope with a separate intercept per bin: $\Delta F_{i,t} = a_b + \beta\, g_{i,t} + u_{i,t}$. The step is the bad news gate, $\Delta F_{i,t} = a_b + \beta\big(1 + \delta\,\mathbf{1}\{e_{i,t} \le -\kappa\}\big)\, g_{i,t} + u_{i,t}$, with the threshold $-\hat{\kappa}$ (dotted line) chosen on a grid of quantiles of $e_{i,t}$ in the same model with a single intercept. Bottom: the share of firm-quarters in each bin. The horizontal axis is linear within ± 0.1 dollars per share and compressed beyond.

Figure 17: Update strength is higher after bad news.

[35] Past work has documented related ideas, though they differ from our theory in important ways. Coibion and Gorodnichenko (2015) document that information rigidity in macroeconomic forecasts is lower during recessions, though they test this through regressing forecast errors on forecast revisions and do not propose a theory to capture this fact. Our theory also says "news" is specifically about realized outcomes of the variable being forecasted rather than bad news about the economy at large. On the theoretical side, Epstein and Schneider (2008) have a model of ambiguity aversion where agents react more strongly to bad news than good news. Our theory also has a two-part structure: when realized earnings come in sufficiently below expectations, the forecaster reacts more to non-earnings news as well, which a model of ambiguity would not capture. We also point out that our theory is discovered automatically by our algorithm with no reference to this existing work.

Where else it applies. Our theory makes reasonable predictions in many other relevant contexts across economics; Table A12 in the appendix gives examples. We also explicitly test portability to one other context in Section 5.4.

What could be improved. We note that there are multiple theories that can generalize our finding that there is less stickiness after low $Y_{i,t} - F_{t-1}[Y_{i,t}]$ in the analyst context. It could be that there is less stickiness after a negative forecast error, which is how our algorithm interprets it. Or it could be that there is less stickiness when a salient piece of information indicates bad news, which is also low $Y_{i,t} - F_{t-1}[Y_{i,t}]$ in the earnings context, but for inflation it would be inflation coming in *above* expectations. The latter theory intuitively feels more sensible (and is also more portable, because bad news applies in settings where there is no forecast error). We note that the summary of the theory given by our portability loop ("update more strongly after a large disappointing surprise"; the full summary is in Appendix F.5) captures this general bad news idea, but it nonetheless names a narrower concept (e_t = the latest forecast error) when naming its variables. A better LLM portability loop could also produce a more general description for e_t.

5.3 The other theories

Table 2 lists all ten theories, each with a short portable description and its R^2 on quarterly forecast revisions. We have nine other theories, seven of which can be categorized into groups (Table A15 gives each in full). Three are close relatives of known theories: a version of the theory we have just discussed without the bad news gate ($\delta = 0$; *history-dependent sticky expectations*, Appendix F.3), *history-dependent rational* expectations, and a variant of adaptive expectations in which the response to the latest surprise is capped in the tails (*saturating error correction*). Two say forecasters react to market prices: one to the *price trend* over some fixed period, and another to the *worst recent return*. We note that these theories are more interesting than they might seem because, as Figure 12 shows, they achieve marginal R^2 over and above the five known theories. Sticky expectations already includes information from the price, so the fact that these theories get additional R^2 suggests that forecasters may react more to prices than to other news of similar informativeness.[36] There are then two theories documenting momentum in forecasters' previous revisions: one is *fixed-target revision momentum* (successive forecasts of $Y_{i,t+1}$), and another is *rolling-target revision momentum* (successive one-period-ahead forecasts). The other two do not naturally fit into groups: one says that forecasts revise down when recent objective news is volatile (*rational-update volatility*), and the other says that forecasts mean-revert (*stale forecast level*).

[36] The coefficient on each of these price theories is positive when we control for the five known theories.

Theory	Forecasts...	R^2
The theory of Section 5.2		
Sticky (unless there's bad news)	... move partway from the last forecast toward an objective forecast, and further after a large disappointment	19.8%
Close relatives of known theories		
History-dependent sticky	... move partway toward an objective forecast, and still respond to information from several periods back	19.1%
History-dependent rational	... follow the recent revisions of an objective forecast	11.8%
Saturating error correction	... correct for the last forecast error, but only up to a cap	3.5%
Reactions to market prices		
Price trend	... follow the *average* recent trend in a related market price	5.3%
Worst recent return	... follow the *worst* recent trend in a related market price	5.3%
Momentum in forecasters' revisions		
Fixed-target revision momentum	... of a fixed *target* show momentum: they keep moving in the direction they were last revised	2.4%
Rolling-target revision momentum	... at a fixed *horizon* show momentum: they keep moving in the direction they were last revised	7.4%
Other		
Rational-update volatility	... are revised down after volatile news	1.2%
Stale forecast level	... mean-revert	1.6%

Notes. Descriptions are the authors' plain-words condensations of each theory's D_T; Table A15 gives D_T as the algorithm returned it. R^2 is for the theory fit alone on the held-out fold 5, target the quarterly forecast revision (as in Figure 11).

Table 2: Short descriptions of each induced theory.

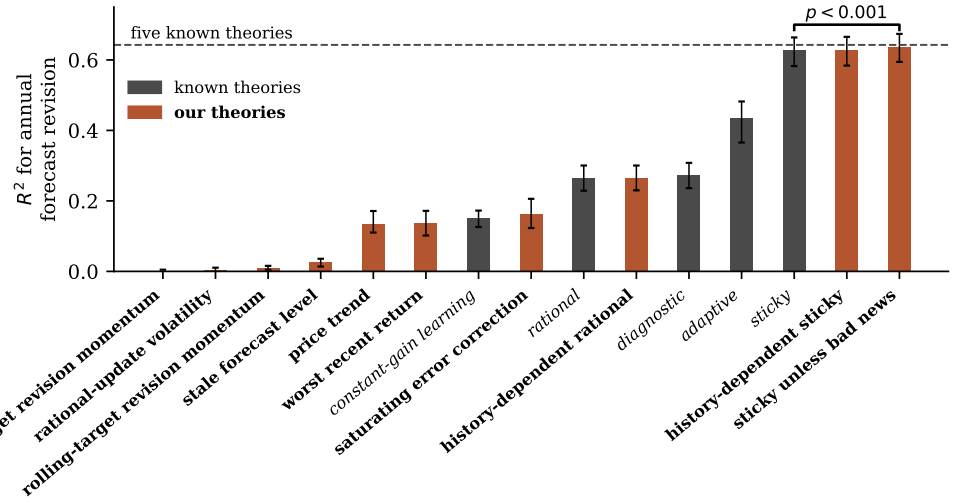
5.4 Portability to annual forecasts

As per our framework in Section 3, the theories we have induced are portable. This means that our theories can make predictions in new settings, but, as discussed in Section 4.1, one cannot expect a priori that the predictions will be good; this is the problem of induction (Hume, 1748; Goodman, 1955). Nonetheless, in science we hope that theories induced in one context will predict well in another, and we next evaluate whether that is true of our theories.

One wrinkle is that each of the ten theories above ports to a different set of contexts, and it is tricky to find a single context that all ten port to. So to make things simple, we port our theories (induced on analyst quarterly forecasts) to analyst annual earnings forecasts. The target there is the revision from the two-year-ahead forecast of a fiscal year's earnings to the one-year-ahead forecast of the same year's earnings.[37] We bind each theory into the annual context by hand, with details in Appendix G.

Figure 18 repeats Figure 11 on annual forecasts, documenting the R^2 of each theory on its own. Our best theory, sticky (unless there's bad news) expectations, reaches an R^2 of 63.7% (95% confidence interval [59.4%, 67.4%]) on annual forecast revisions, against 62.8% for sticky expectations (difference is 0.9 percentage points, 95% CI [0.4, 1.4]).

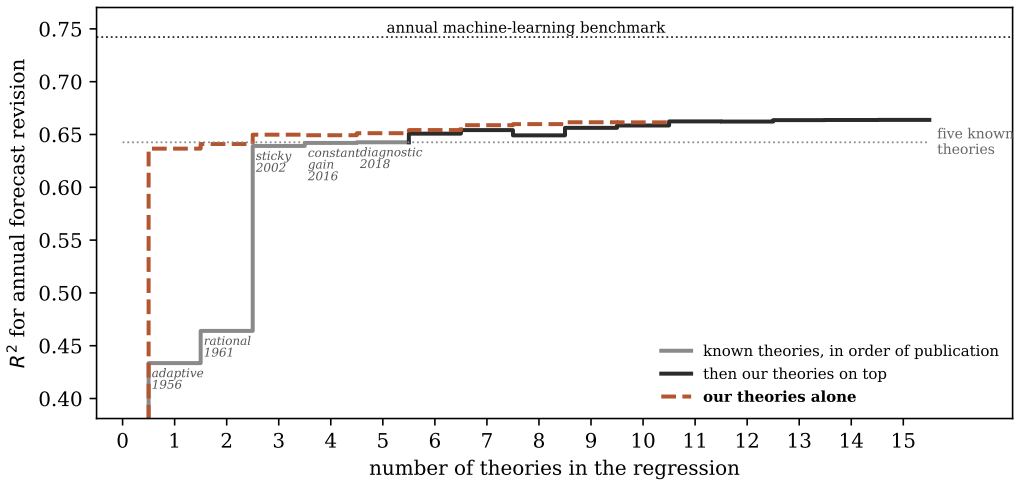
[37] With τ indexing years, $\Delta F_{i,\tau} = F_\tau[Y_{i,\tau+1}] - F_{\tau-1}[Y_{i,\tau+1}]$. The annual ML benchmark is built the same way as $\hat{m}$, on the annual features in Table A20.



Notes. The target is the annual forecast revision $\Delta F_{i,\tau}$. The five known theories are built on the annual panel; the internal constants of our ten theories are refit on annual folds 1–4. Each theory is fit alone on annual folds 1–4 and R^2 is reported on annual fold 5, with 95% confidence intervals from a firm-level bootstrap (200 draws). The dashed line is the five known theories together. The p-value of the bracket is from a paired firm-level bootstrap of the difference on annual fold 5 (1,000 draws).

Figure 18: Out-of-sample R^2 on annual forecasts of the five known theories (grey) and our induced theories (orange).

We can also benchmark the R^2 that all ten of our theories collectively achieve, which is shown in Figure 19 (a mirror of Figure 14). Adding in the known theories first by publication date, we again see a leveling-off effect: adaptive, rational and sticky expectations reach 63.9%, and constant-gain learning and diagnostic expectations together add only 0.3%. Added on top of the known theories (the black line), our theories nonetheless still make substantial progress and close 21.3% of the gap to the annual ML benchmark. On their own (the orange line), the first three of our theories combined already beat the five known theories combined, and all 10 close 18.9% (95% confidence interval [5.4%, 31.8%]) of the gap; though this is a smaller share than the quarterly 32.5%, we still take this to be substantial progress given the difficulty of achieving marginal R^2 gains.



Notes. The target is the annual forecast revision $\Delta F_{i,\tau}$. As Figure 14, on the annual sample: each step is one linear regression of the annual forecast revision on all the theories added so far. The grey line adds the five known theories in order of publication; the black line then adds our theories, induced on quarterly forecasts and bound into the annual context by hand, in greedy order on top of the five known theories; the orange line adds our theories in greedy order with no known theory. At each step the constants inside our theories are refit jointly on annual folds 1–4. The orders are in Table A17 in the appendix. The dotted lines are the five known theories together and the annual ML benchmark.

Figure 19: R^2 on annual forecasts as theories are added one at a time: the five known theories in order of publication, then ours on top of them or on their own.

We also want to point out that the machine-learning benchmark $\hat{m}$ of Section 2 cannot even make predictions here, because it uses variables (e.g. quarterly forecasts) that are not in the feature set for annual revisions.[38] This illustrates the purpose of having theories: even if we only want to move to a slightly different setting, machine learning predictors as well as many bindable functions may not port.

5.5 The importance of words in induction

One central theme of this paper, which we argued for in our framework and again in our algorithm, is that words and binding are crucial to induction. We now close by demonstrating this empirically. We do this by removing words and binding from two parts of the process: search and porting.

Search without binding. What would it mean to search without using words? Obviously, if one just constructs a machine-learning benchmark $\hat{m} \approx E[y_C \mid x_C]$, it will not be bindable: this was our earlier point that induction is not just finding a theory that binds to $\hat{m}$ (Section 4.1). But even the symbolic regression algorithm we use to search for simple functional forms (Section 4.4) will not output bindable functions unless we embed an explicit checker into the search.

To see this in practice, we run symbolic regression, but instead of the bindability check of our algorithm, we put in the standard complexity penalty used in computer science (Cranmer, 2023): a cap on the number of variables and operations in the formula.[39] The results are shown in Figure 20. The outputted function explains 64.0% of the variation in $\hat{m}$, significantly higher than the 52.9%

[38] This is because the feature set for annual revisions cannot contain any forecast information between year $\tau - 1$ and year τ, and the quarterly feature set contains forecast information up until one quarter before τ.

[39] We set the cap to the number of variables and operations of the highest-R^2 theory we induce in Section 5.

of our best-predicting theory. But the higher predictive accuracy comes at the cost of bindability: there is no way to convert it into a mechanism that can port to other contexts.

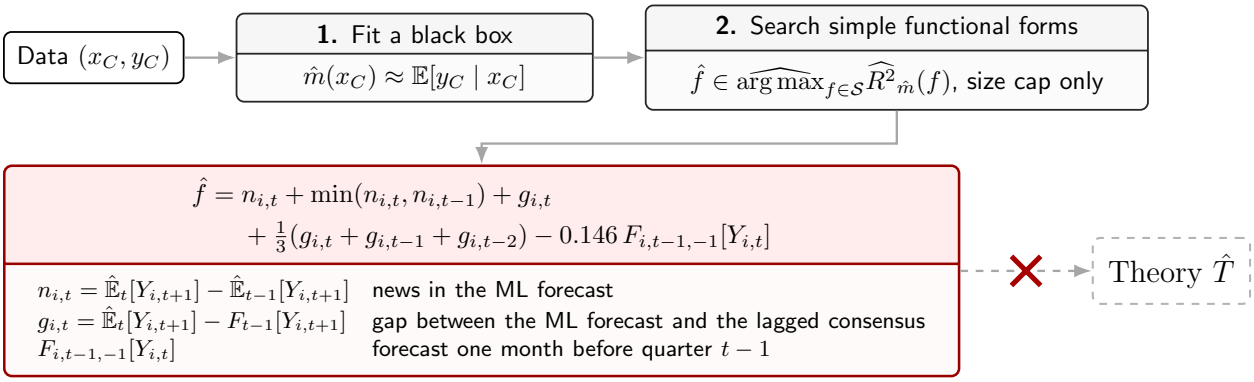


Figure 20: Symbolic regression with no bindability check returns functions that predict well but cannot be converted into theories.

What is the lesson here? It is not simply that symbolic regression is not enough and its outputs require interpretation: it is hard to imagine interpreting the $\hat{f}$ of Figure 20 in a meaningful way. Rather, it is that a constraint on bindability, $f \in \widehat{\mathcal{F}}^B_C$, must itself be a central part of the search process, as we implement in Section 4.5.

Porting without binding. Our procedure ported the functions f discovered on quarterly forecasts to annual forecasts by first attaching portable words to f and then binding it to annual forecasts. Could we have ported these functions directly to annual forecasts without using any words and binding? If we want to fully eliminate the role of words, we should first rename all the variables of the function to $x_1, x_2, \ldots$ (as in our example in Section 3.2) and rename the columns of annual forecasts to `col_1`, `col_2`, ... We can then ask an LLM to "bind" the function to annual forecasts by choosing some mapping from `col_j` into x_i (Figure 21), and measure the resulting R^2, which is shown in Figure 22.

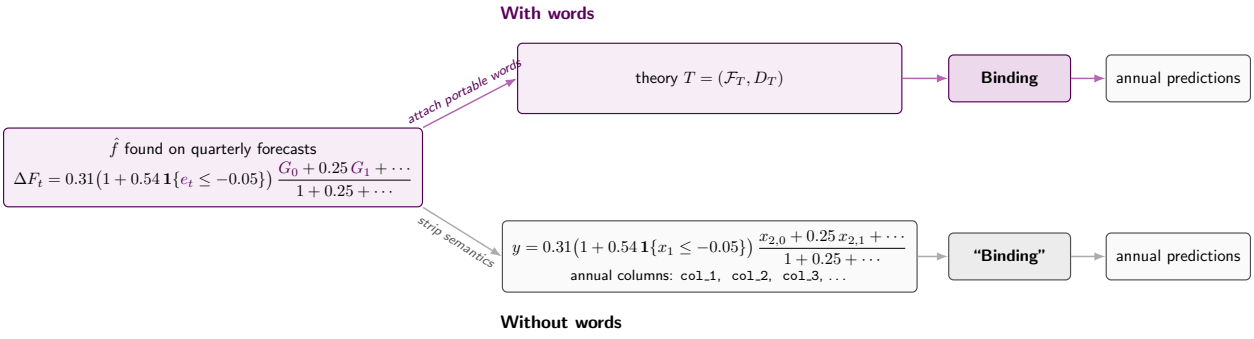
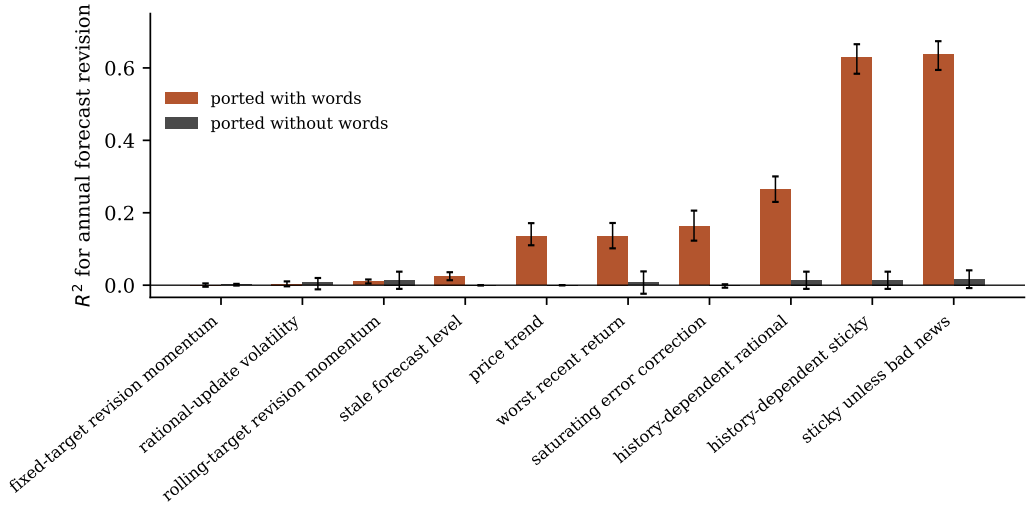


Figure 21: Porting a function on quarterly forecasts to annual forecasts with and without words.

This exercise fails: when porting without words, each of the functions attains an R^2 approximately equal to zero (Figure 22). Of course, this is a fairly absurd exercise, since without words, the "binding" step in Figure 21 is just matching variables to columns at random, and most random combinations have minimal predictive power. This is exactly the point we made conceptually in

Figure 2, when we said that there are an absurdly large number of ways to "bind" without words, now demonstrated empirically.



Notes. Each of the ten functions discovered on quarterly forecasts, fit alone on annual forecast revisions and ported with words (the port of Section 5.4, as in Figure 18) and without words (the variables and the annual columns stripped of their names and mapped by an LLM). Internal constants refit on annual folds 1–4, R^2 on annual fold 5, 95% confidence intervals from a firm-level bootstrap (200 draws).

Figure 22: Out-of-sample R^2 of the functions discovered on quarterly forecasts, ported to annual forecasts with and without words.

6 Conclusion

Across a wide variety of domains, machine learning algorithms have been shown to outpredict economic theories (Peysakhovich and Naecker, 2017; Fudenberg and Liang, 2019; Fudenberg et al., 2022). Nonetheless, we as researchers are often still interested in producing theories, because they provide insights that are portable across settings. Machine learning algorithms, on the other hand, only predict in the setting they are trained on. Recent work at the intersection of machine learning and economic theory has made progress at generating interpretable insights from black box models, but still requires a human to produce the final theory (Fudenberg and Liang, 2019; Mullainathan and Rambachan, forthcoming; Ludwig and Mullainathan, 2024).

In this paper, we develop an end-to-end procedure that induces theories directly from machine learning algorithms. This first requires conceptually understanding what makes a theory portable. This turns out to be its words, combined with binding: the act of applying those words to make predictions in new contexts. Then we implement binding via large language models, which allows us to distill the machine learning algorithm into functions that correspond to portable theories. We then invert the binding process and map those functions back onto maximally portable theories. We run our procedure on financial analysts' quarterly earnings forecasts and induce ten new theories of expectations. Together they close a third of the gap between the five known theories of expectations from Afrouzi et al. (2023) and the machine learning algorithm, and we also show that they predict well on annual forecasts.

Our theories add meaningful new R^2 to existing theories and come with descriptions that can be carried to other settings. But, as discussed in the introduction, we do not think any of them represents a breakthrough in our understanding of expectations. Nonetheless, we think our framework

and algorithm lay a foundation that future work can extend to get there, and Section 4.7 lays out several suggestions: giving the algorithm richer data, requiring the search to deliver other desirable properties that an algorithm can already check, and enriching our framework to incorporate other desiderata that we cannot yet check (such as whether a theory delivers particular implications once its consequences are derived in a larger model).

Finally, we also think that our formalization of binding, and of the role of words in theories, raises other interesting avenues for research outside of induction. First, we can try to systematically evaluate the vagueness of a theory, as measured by whether there is consensus about its binding function (e.g. in prospect theory (Kahneman and Tversky, 1979), researchers often disagree about how to bind the reference point). With new LLM tools we can not only document existing disagreement about binding but also develop automatic methods for evaluating the vagueness of new proposed theories. Second, we can try to broaden existing theories by searching for more general descriptions that leave their predictions unchanged where they already apply, but also bind in new contexts. Third, we can try to innovate on the binding function itself by finding ways to apply the words of existing theories to contexts we previously thought were outside their scope. For instance, this is what Becker (1968, 1973) did when he took price theory to crime and marriage; an algorithmic binder could search for such applications systematically. These are all open questions that we hope future work will explore.

A Data and sample

This appendix collects the details behind Section 2: how the data are built, how the sample is split, how the machine-learning benchmark $\hat{m}$ is built, and how the five known theories are made estimable.

A.1 Data construction

Concretely, we take past realized earnings, past consensus forecasts, analyst disagreement statistics (e.g. dispersion, range), and a wide range of firm-level variables (the ones used by Zhang et al. (2025) and van Binsbergen et al. (2023)). Our data construction is based on the replication package of Zhang et al. (2025). We also expand their feature set by taking lags and adding in cross-sectional variables such as the dispersion. The full list of columns is in Table A1. Summary statistics for the quarterly panel are in Table A2.

Category	Variable	Lag clock	Most recent	Lag depth	Columns
Announced earnings	quarterly EPS	quarterly	latest announced by F_t	13 quarters	37
Announced earnings	quarterly EPS	annual	the quarter of the last fiscal-year report, at or before F_{t-1}	8 years	22
Announced earnings	annual EPS	annual	latest announced fiscal year by F_t	9 years	25
Consensus forecasts, monthly snapshots	forecast level, horizons $q1, q2, q3, y1, y2$	monthly	at F_{t-1}	13 months	65
Consensus forecasts, monthly snapshots	month-over-month revision, horizons $q1, q2, q3, y1, y2$	monthly	into F_{t-1}	12 months	60
Consensus forecasts, quarterly snapshots	forecast level, horizons $q1, q2, q3$	quarterly	at F_{t-1}	12 quarters	36
Consensus forecasts, quarterly snapshots	forecast error, realized minus consensus	quarterly	latest announced before F_t	12 quarters	12
Consensus forecasts, quarterly snapshots	snapshot-over-snapshot revision	quarterly	at F_{t-1}	12 quarters	12
Consensus forecasts, annual snapshots	forecast level, horizons $y1, y2$	annual	the quarter of the last fiscal-year report, at or before F_{t-1}	8 years	16
Consensus forecasts, annual snapshots	forecast error, realized minus consensus	annual	latest announced before F_t	8 years	8
Consensus forecasts, annual snapshots	snapshot-over-snapshot revision	annual	the quarter of the last fiscal-year report, at or before F_{t-1}	8 years	8
Fundamentals	the 99 accounting ratios and market variables of Zhang, Zhu, and Linnainmaa (2025): the WRDS financial-ratio suite, CRSP market variables, and per-share fundamentals recovered from the ratios	monthly	the month before F_t	12 months	3,366
Fundamentals	the same 99	quarterly	at F_{t-1}	12 quarters	3,366
Fundamentals	the same 99	annual	the quarter of the last fiscal-year report, at or before F_{t-1}	8 years	2,178
Disagreement	forecast standard deviation, per horizon	monthly	at F_{t-1}	13 months	65
Disagreement	forecast high minus low, per horizon	monthly	at F_{t-1}	13 months	65
Disagreement	forecast mean minus median, per horizon	monthly	at F_{t-1}	13 months	65
Disagreement	analysts in the consensus, per horizon	monthly	at F_{t-1}	13 months	65
Disagreement	analysts revising up, per horizon	monthly	at F_{t-1}	13 months	65
Disagreement	analysts revising down, per horizon	monthly	at F_{t-1}	13 months	65
Coverage history	length of the current coverage run, the gap in coverage before it, number of earlier coverage runs, coverage breaks in the past year	—	—	lagless	5
Total					9,606

Notes. Raw variables only — no constructed features beyond per-variable trend and volatility over the lag windows, counted inside their rows. Clocks: t indexes quarters; F_t is recorded on quarter t's statement date. Quarter steps walk the firm's previous statement dates; year steps walk its fiscal-year-report quarters — quarters at which a fiscal year had just been announced — counted strictly before the row — equivalently, at or before F_{t-1} — so "the quarter of the last fiscal-year report, at or before F_{t-1}" is well defined on every row and sits between one quarter and one year back — "before F_{t-1}" restricts the report, and the reading at its quarter can be as recent as F_{t-1} itself. "Most recent" differs by data type. The predicted revision runs from F_{t-1} to F_t, so consensus and disagreement readings stop at F_{t-1} — anything later is the target itself. Fundamentals are realized information the analyst reacts to; they run up to the month before F_t (month-end-dated, hence the one-month delay). Announced earnings are announcement-dated: a reading enters the moment it is announced, up to F_t itself. Past forecast errors resolve at announcement too: the newest error is the quarter announced just before F_t, and no consensus information from inside the window enters because the error's forecast component was recorded at or before F_{t-1}. Source: the replication package.

Table A1: The raw feature set, quarterly target.

Panel A: Sample						
Firms						14,917
Firm–quarters						486,943
Quarters per firm (median)						19
Sample period						1984–2024
Panel B: Variables						
	N	Mean	SD	P10	P50	P90
Forecast $F_t[Y_{i,t+1}]$	486,943	0.33	0.60	−0.18	0.24	0.93
Realized outcome $Y_{i,t+1}$	486,943	0.32	0.70	−0.25	0.23	0.98
Forecast error $Y_{i,t+1} - F_t[Y_{i,t+1}]$	486,943	−0.01	0.30	−0.16	0.01	0.13
Forecast revision $\Delta F_{i,t}$	405,787	−0.03	0.16	−0.13	−0.01	0.05
Analysts per consensus	486,943	6.1	5.8	1	4	14

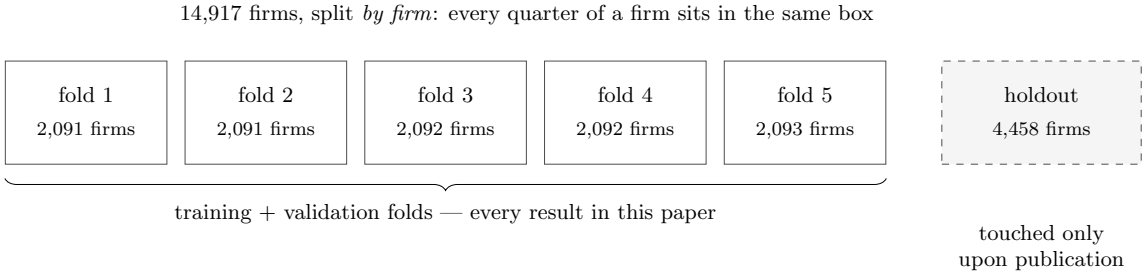
Notes. Quarterly earnings per share in split-adjusted dollars per share; values are not deflated by the share price. The forecast is the consensus one quarter ahead. N varies with each variable's coverage: realized earnings and the consensus forecast cover every firm–quarter of the panel; the forecast revision additionally requires that the prior consensus snapshot exist, hence its smaller N. All raw variables (including $F_t[Y_{i,t+1}]$ and $Y_{i,t+1}$) are winsorized at the 1%/99% level within month; $\Delta F_{i,t}$ is constructed from the winsorized levels and is not winsorized again.

Table A2: Summary statistics for the IBES quarterly panel. Full panel: 486,943 firm-quarters.

A.2 The sample split

We split by firm: 30% of the 14,917 firms go to a holdout, and the rest are randomly assigned (stratified by firm coverage and size) to 5 training+validation folds (Figure A1). This ensures that different quarters of the same firm always sit in the same fold/holdout. Because folds 1–4 are then used to search for theories (Section 4), every result for our induced theories will be scored on fold 5 alone; upon publication we will evaluate the results on the holdout as well.

One might wonder why we do not use a rolling time split. The reason is that while finance is often interested in problems like return predictability, for which generalization across time is crucial, we are interested in theory induction, for which the relevant generalization is *transfer* across settings. We study the transfer performance of our theories in Section 5. For the purposes of induction, taking firms as the unit of observation also gives us far more effective data to induce from than taking quarters as the unit.



Notes. Assignment is at the firm level: 30% of the 14,917 firms go to the holdout, and the rest are randomly assigned (stratified by firm activity and size) to 5 equal training+validation folds, so every quarter of a firm sits in the same block. The holdout is untouched until publication.

Figure A1: The firm-level sample split.

A.3 Computing $\widehat{R^2}$

As shown in Figure A1, our training and validation sample has five folds. For the purposes of theory induction, we treat folds 1–4 as the search data. Since we will be doing lots of fitting during search, it is important to have an out-of-sample estimate of $\widehat{R^2}_{\hat{m}}$. For a fixed functional form, there are two kinds of constants to estimate. For example, in $\alpha\,\mathbf{1}\{x \geq c\}$, α is an outer coefficient and the threshold c is an inner constant. For a fixed inner constant (e.g. c), the outer coefficients (e.g. α) are fit by linear regression, which has a closed form, so we can quickly estimate $\widehat{R^2}_{\hat{m}}$ out of sample: we rotate the held-out fold within folds 1–4 and fit the outer coefficients on the other three. The inner constants (e.g. c) have no closed-form solution and must be found by numerical search, so for speed we fit them once on all of folds 1–4 together.

In Section 5, when we evaluate the R^2 of theories, we fit all coefficients on folds 1–4 and evaluate the R^2 for explaining y_C on fold 5. The reported coefficients, by contrast, are fit on fold 5 alone (Table A16), because folds 1–4 were used to pick the theories.

A.4 The machine-learning benchmark

We build $\hat{m}$ as a stacked ensemble of gradient-boosted trees, neural networks, and sequence models. We fit our model on the five training+validation folds via rotating cross-validation: predictions on fold i are trained on all folds excluding i. For the neural networks and sequence models we use a multitask learning approach (Caruana, 1997) — the model is pretrained on hundreds of related auxiliary targets (e.g. synthetic expectations produced by known theories, forecasts of other horizons) and then fine-tuned on $\Delta F_{i,t}$. We produce 24 individual models across the three architectures and combine them using non-negative least squares.

Timing of the inputs. $\hat{m}$ uses all fundamental information about firm i dated t or earlier, and all forecast information dated $t-1$ or earlier. The asymmetry is necessary: forecasts made between $t-1$ and t would leak the very revision being predicted. Existing theories implicitly follow this asymmetry as well: adaptive expectations, for instance, writes the revision in terms of $Y_{i,t} - F_{t-1}[Y_{i,t}]$. This uses fundamental information dated t, which is necessary, since the goal of these theories is to learn how forecasts change as *a function* of what happens at time t. What none of these theories is allowed to do, though, is to use forecasts dated t, because that would be circular: it makes no sense to build a model of time-t forecasts (or revisions) using time-t forecasts.

A.5 The five known theories

Before we can benchmark these theories against $\hat{m}$, we need to do a little bit of extra work: these are generally written as theories of the forecast itself, not of the forecast revision. So our first task is to derive each theory's implication for forecast revisions. The results are shown in column two of Table A3.

Next, once we derive the implications of these theories for forecast revisions, we need to decide how to estimate them. The main complication is that theories like rational expectations reference objects like $\mathbb{E}\big[Y_{i,t+1} \mid \mathcal{I}_{t-k}\big]$, which are not directly computable: we can neither observe the full information set $\mathcal{I}_{t-k}$ nor compute a true conditional expectation. The literature's standard approach to rational expectations is to derive testable implications (e.g. that forecast errors are not predictable) and then reject them; but since our goal is not to reject theories but to benchmark their predictive accuracy, that is not sufficient. Strictly speaking, if we were to follow the literature's norm in applying rational expectations to our setting, the theory would make no predictions. But for the pragmatic purpose of having some benchmarks, we build proxies for rational-expectations terms: in place of $\mathbb{E}\big[Y_{i,t+1} \mid \mathcal{I}_{t-k}\big]$ we use $\hat{\mathbb{E}}_{t-k}\big[Y_{i,t+1}\big]$, which is itself a machine-learning model designed to predict $Y_{i,t+1}$ from all fundamental information in our dataset known at time $t-k$.[40] The full estimated form of all five theories is shown in the 'Estimated' rows of Table A3.

We aim to follow the constant-gain learning specification of Malmendier and Nagel (2016), but it does not apply naturally to consensus forecasts, because the consensus has no "age": in their specification, each cohort learns from the inflation it has lived through. To operationalize it, we assume the consensus learns from the firm's entire history of earnings. Many firms have only a short history, so constant-gain learning is often run on a handful of observations, which can give explosive forecasts. So we winsorize the constant-gain-learning forecast revision at the 1st and 99th percentiles.

We also make small modifications to diagnostic and rational expectations: taken literally, diagnostic expectations puts a coefficient of one on the rational term $\hat{\mathbb{E}}_t[Y_{i,t+1}] - \hat{\mathbb{E}}_{t-1}[Y_{i,t+1}]$, and so does rational expectations. To be generous to these theories and allow better fits, we put free parameters on these terms as well.

For the combination of all five theories, we take the estimated terms in the 'Estimated' rows of Table A3 and put them all into one linear regression. Writing $\hat{\mathbb{E}}_s$ for $\hat{\mathbb{E}}_s\big[Y_{i,t+1}\big]$, the regression is

$$\beta_1(\hat{\mathbb{E}}_t - \hat{\mathbb{E}}_{t-1}) + \beta_2(Y_{i,t} - F_{t-1}[Y_{i,t}]) + \beta_3(\hat{\mathbb{E}}_t - F_{t-1}[Y_{i,t+1}]) + \beta_4[(\hat{\mathbb{E}}_t - \hat{\mathbb{E}}_{t-1}) - (\hat{\mathbb{E}}_{t-1} - \hat{\mathbb{E}}_{t-2})] + \beta_5(F_t^{\mathrm{CG}} - F_{t-1}^{\mathrm{CG}}),$$

where the β_4 (diagnostic) term does not include the part that is already encapsulated by rational expectations (the β_1 term), and F^{CG} is the constant-gain-learning forecast, defined in Table A3.

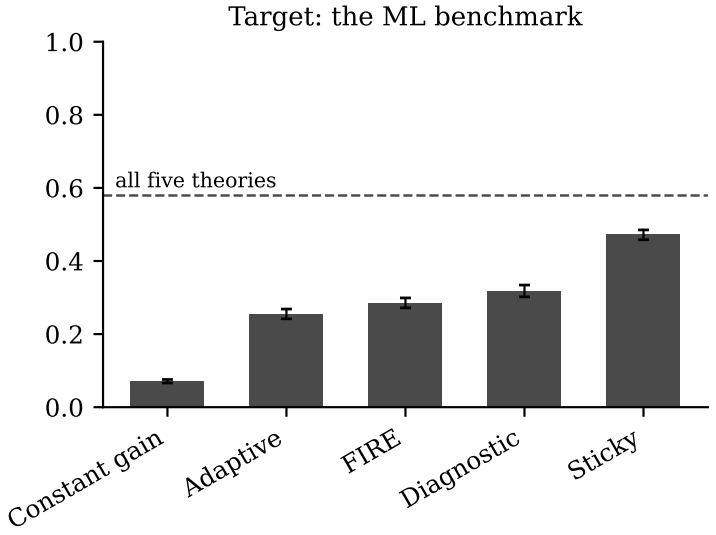
[40] $\hat{\mathbb{E}}$ is also computed out of fold: its predictions on fold i are trained on all training+validation folds excluding i.

Full-information rational expectations (FIRE)	Forecast:	$\mathbb{E}[Y_{i,t+1} \mid \mathcal{I}_t]$
	Revision:	$\mathbb{E}[Y_{i,t+1} \mid \mathcal{I}_t] - \mathbb{E}[Y_{i,t+1} \mid \mathcal{I}_{t-1}]$
	Estimated:	$\beta^{\dagger}\big(\hat{\mathbb{E}}_t[Y_{i,t+1}] - \hat{\mathbb{E}}_{t-1}[Y_{i,t+1}]\big)$
Adaptive expectations	Forecast:	$F_{t-1}[Y_{i,t+1}] + \lambda\big(Y_{i,t} - F_{t-1}[Y_{i,t}]\big)$
	Revision:	$\lambda\big(Y_{i,t} - F_{t-1}[Y_{i,t}]\big)$
	Estimated:	$\beta^{\dagger}\big(Y_{i,t} - F_{t-1}[Y_{i,t}]\big)$
Sticky expectations	Forecast:	$(1-\lambda)\,F_{t-1}[Y_{i,t+1}] + \lambda\,\mathbb{E}[Y_{i,t+1} \mid \mathcal{I}_t]$
	Revision:	$\lambda\big(\mathbb{E}[Y_{i,t+1} \mid \mathcal{I}_t] - F_{t-1}[Y_{i,t+1}]\big)$
	Estimated:	$\beta^{\dagger}\big(\hat{\mathbb{E}}_t[Y_{i,t+1}] - F_{t-1}[Y_{i,t+1}]\big)$
Diagnostic expectations	Forecast:	$\mathbb{E}[Y_{i,t+1} \mid \mathcal{I}_t] + \theta\big(\mathbb{E}[Y_{i,t+1} \mid \mathcal{I}_t] - \mathbb{E}[Y_{i,t+1} \mid \mathcal{I}_{t-1}]\big)$
	Revision:	$\big(\mathbb{E}[Y_{i,t+1} \mid \mathcal{I}_t] - \mathbb{E}[Y_{i,t+1} \mid \mathcal{I}_{t-1}]\big) + \theta\big[\big(\mathbb{E}[Y_{i,t+1} \mid \mathcal{I}_t] - \mathbb{E}[Y_{i,t+1} \mid \mathcal{I}_{t-1}]\big) - \big(\mathbb{E}[Y_{i,t+1} \mid \mathcal{I}_{t-1}] - \mathbb{E}[Y_{i,t+1} \mid \mathcal{I}_{t-2}]\big)\big]$
	Estimated:	$\beta^{\dagger}\big[\big(\hat{\mathbb{E}}_t[Y_{i,t+1}] - \hat{\mathbb{E}}_{t-1}[Y_{i,t+1}]\big) + \theta^{\dagger}\big[\big(\hat{\mathbb{E}}_t[Y_{i,t+1}] - \hat{\mathbb{E}}_{t-1}[Y_{i,t+1}]\big) - \big(\hat{\mathbb{E}}_{t-1}[Y_{i,t+1}] - \hat{\mathbb{E}}_{t-2}[Y_{i,t+1}]\big)\big]\big]$
Constant-gain learning	Forecast:	$F_t^{\mathrm{CG}}[Y_{i,t+1} \mid \gamma] \equiv \hat{a}_i + \hat{b}_i\,Y_{i,t}$, a within-firm least-squares fit with geometric gain $\gamma^{\dagger}$
	Revision:	$F_t^{\mathrm{CG}}[Y_{i,t+1} \mid \gamma] - F_{t-1}^{\mathrm{CG}}[Y_{i,t+1} \mid \gamma]$
	Estimated:	$\beta^{\dagger}\big(F_t^{\mathrm{CG}}[Y_{i,t+1} \mid \gamma] - F_{t-1}^{\mathrm{CG}}[Y_{i,t+1} \mid \gamma]\big)$

Notes. Free parameters are marked † (λ is absorbed into β wherever it enters multiplicatively). $\hat{\mathbb{E}}_{t-k}[Y_{i,t+1}]$ is a machine-learning forecast of $Y_{i,t+1}$ from all fundamental information in the dataset known at $t-k$, fit on the training folds only. The constant-gain forecast fits $(\hat{a}_i, \hat{b}_i) = \arg\min_{a,b} \sum_{s<t} \gamma^{\,t-s}\big(Y_{i,s+1} - a - b\,Y_{i,s}\big)^2$ within firm, $\gamma^{\dagger}$ common to all firms, and $F_{t-1}^{\mathrm{CG}}[Y_{i,t+1} \mid \gamma]$ iterates the perceived law one step: $\hat{a}_i(1+\hat{b}_i) + \hat{b}_i^2\,Y_{i,t-1}$. In the implementation the gain is fit to maximize predictive R^2; the fitted value is $\gamma = 1$ (equal weights — the least-squares-learning limit). Many firms have only a short earnings history, so the constant-gain fit can give explosive forecasts; we winsorize the constant-gain forecast revision at the 1st and 99th percentiles.

Table A3: The five known theories and their implications for forecast revisions.

Figure A2 asks instead how well the individual theories can explain the machine-learning benchmark itself, as well as how well all five together in a linear regression can.



Notes. Same sample, folds and fits as Figure 1, with the machine-learning benchmark $\hat{m}$ as the target instead of the revision $\Delta F_{i,t}$.

Figure A2: Out-of-sample R^2 of the five known theories, with the machine-learning benchmark as the target.

B Theories and binding: details and examples

This appendix collects the formal material behind Section 3: how a bound theory makes a prediction, more examples of binding, which descriptions state a single mechanism, what binding buys, induction as a search over theories, and what the words buy in induction, with proofs.

B.1 Theory use: bind, select, predict

Given a theory $T = (\mathcal{F}_T, D_T)$ and a context C, binding produces the collection of allowable functions $\mathcal{F}_T \circ B_{T,C}$. To make predictions, the researcher then selects one allowable function using data from the context: given a loss function L and the distribution $\mathcal{P}_C$ of inputs $x_C \in \mathcal{X}_C$ and outputs $y_C \in \mathcal{Y}_C$ in context C, the selected function is

$$f^*_{T,C} = \arg \min_{f \in \mathcal{F}_T \circ B_{T,C}} \mathbb{E}_{\mathcal{P}_C}\big[L(f(x_C), y_C)\big],$$

and the theory's prediction at an input x_C is $f^*_{T,C}(x_C)$. In practice we observe a finite collection of input–output pairs $(x_{C,i}, y_{C,i})$, $i = 1, \ldots, N$, drawn from $\mathcal{P}_C$, and select the function on that sample instead,

$$\hat{f}_{T,C} = \arg \min_{f \in \mathcal{F}_T \circ B_{T,C}} \frac{1}{N} \sum_{i=1}^{N} L\big(f(x_{C,i}), y_{C,i}\big).$$

In other words, binding specifies what the theory permits in a context, and the data is used to pick among the allowable functions.

B.2 More examples of binding

Example: adaptive expectations in two contexts. Adaptive expectations is the theory with inputs $x^* = (x_1^*, x_2^*)$ — the latest realized value and the forecast of that value made one period earlier — outcome y^*, the forecast of the next value, function class

$$\mathcal{F}_T = \left\{x_2^* + \beta\,(x_1^* - x_2^*) : \beta \in \mathbb{R}\right\},$$

and description D_T = "forecasters adjust their lagged one-period-ahead forecast in proportion to the most recent forecast error." Bound to the analyst context of Section 3.2,

$$B_{T,C_{\text{analyst}}}(x_C) = \big(Y_{i,t},\ F_{t-1}[Y_{i,t}]\big), \qquad \mathcal{F}_T \circ B_{T,C_{\text{analyst}}} = \left\{F_{t-1}[Y_{i,t}] + \beta\,(Y_{i,t} - F_{t-1}[Y_{i,t}]) : \beta \in \mathbb{R}\right\}.$$

Bound to the inflation context of Section 3.2,

$$B_{T,C_{\text{inflation}}}(x_C) = \big(\pi_t,\ F_{t-1}[\pi_t]\big), \qquad \mathcal{F}_T \circ B_{T,C_{\text{inflation}}} = \left\{F_{t-1}[\pi_t] + \beta\,(\pi_t - F_{t-1}[\pi_t]) : \beta \in \mathbb{R}\right\}.$$

Example: choice under risk. Take expected utility, with the function class and description of Section 3.2. Consider a context where the setting is experimental subjects valuing lotteries one at a time; the inputs are the subject's endowment w and a binary lottery that pays a with probability q and b otherwise, $\mathcal{X}_C = \{(w, a, b, q)\}$; and the output is the elicited certainty equivalent. The binding sends the endowment to wealth and fills the unused payoff slots with zeros:

$$B_{T,C}(w, a, b, q) = (w;\ a,\ b,\ 0, \ldots, 0;\ q,\ 1-q,\ 0, \ldots, 0),$$

and the allowable class is $\{u^{-1}(q\,u(w+a) + (1-q)\,u(w+b)) - w\}$. Binding does not have to be only variable selection from $\mathcal{X}_C$, as in the expectations contexts; it can also involve computations, like $p_2 = 1 - q$. Note that binding expected utility into the analyst context produces $B_{T,C_{\text{analyst}}} = \emptyset$: there are no payoffs or probabilities in forecasting settings.

B.3 Which descriptions state a single mechanism

Section 4.1 requires that the description D_T state a single mechanism. Below are examples of descriptions we do and do not count as a single mechanism.

A single mechanism:

- "Forecasts extrapolate the latest trend in realized outcomes" (extrapolative expectations)
- "Forecasters adjust their lagged one-period-ahead forecast in proportion to the most recent forecast error" (adaptive expectations)
- "Forecasters learn by regression, discounting old observations" (constant-gain learning)
- "Analysts move their forecast toward the latest realized earnings by a fraction that falls with the number of quarters they have covered the firm"
- "People value a lottery at its expected value plus a premium proportional to how much its best outcome exceeds the best outcome of the previous lottery they valued"
- "Households spend a fixed fraction of a windfall in the month it arrives, and the fraction is higher when the windfall arrives as a separate deposit rather than as part of regular pay"

Not a single mechanism:

- A conjunction of many mechanisms — forecasters incorporate new information only partially, extrapolate recent changes in realized outcomes, and respond less if news is announced on a Tuesday
- A transcription of a computation, such as "column 17 times the rolling skew of column 243"
- A description of a mathematical object that doesn't describe any economic mechanism, such as "the set of neural networks"

B.4 What binding buys when applying theories

Why do we need words and binding in our formalism? The words are what allow the theory to map its function class $\mathcal{F}_T$ into a new context in the right way. If you were to know $\mathcal{F}_T$ without having access to D_T and D_C, then there are many ways to plug $\mathcal{F}_T$ into the context. To make this precise, consider a context C with inputs $x \in \mathcal{X}_C = \mathbb{R}^p$ and a theory T^* with $k \leq p$ variables x^*, and suppose the theory is "correct", meaning $y = f^*(x)$, where $f^* \in \mathcal{F}_{T^*} \circ B_{T^*,C}$.

Our goal is to recover the true function f^*. We measure what we know about it by the size of the smallest set of functions we know it belongs to. Let us suppose that the set of all theories is $\mathcal{T} = \{(\mathcal{F}_{T_1}, D_{T_1}), \ldots, (\mathcal{F}_{T_M}, D_{T_M})\}$, each with k variables x^*. For the proposition below, assume every binding $B_{T,C}$ is a permutation binding:[41] for some permutation $P\colon \{1,\ldots,k\} \to \{1,\ldots,p\}$,

$$B_{T,C}(x) = B_P(x) = \big(x_{P(1)}, \ldots, x_{P(k)}\big).$$

The following proposition demonstrates that if we want to learn f^*, knowing the binding is much more important than just knowing $\mathcal{F}_{T^*}$:

Proposition 1 (What binding buys when applying theories)**.** *Suppose the function class $\mathcal{F}_{T_i}$ of each theory contains $G < \infty$ distinct functions.*[42] *Suppose also that the sets $\mathcal{F}_{T_i} \circ B_P$ are pairwise disjoint over all pairs (i, P). Consider three states of knowledge:*

(a) we know that $\mathcal{F}_{T^*}$ is one of $\mathcal{F}_{T_1}, \ldots, \mathcal{F}_{T_M}$, but do not know the binding function, i.e. $f^* \in \mathcal{F}^{(a)} = \bigcup_{i=1}^{M} \bigcup_P \mathcal{F}_{T_i} \circ B_P$;

(b) we know $\mathcal{F}_{T^*}$, but not its binding function, i.e. $f^* \in \mathcal{F}^{(b)} = \bigcup_P \mathcal{F}_{T^*} \circ B_P$;

(c) we know $\mathcal{F}_{T^*}$ and $B_{T^*,C}$, i.e. $f^* \in \mathcal{F}^{(c)} = \mathcal{F}_{T^*} \circ B_{T^*,C}$.

Then

$$\frac{|\mathcal{F}^{(a)}|}{|\mathcal{F}^{(b)}|} = M, \qquad \frac{|\mathcal{F}^{(b)}|}{|\mathcal{F}^{(c)}|} = \frac{p!}{(p-k)!}.$$

As $p \to \infty$, the second ratio dwarfs the first.

The proposition says that if our dataset has many columns p, there is a huge information gain from knowing the binding function. For instance, in our analyst context $p = 9{,}606$, so even at $k = 2$ the ratio $|\mathcal{F}^{(b)}|/|\mathcal{F}^{(c)}|$ is already $9{,}606 \times 9{,}605 \approx 92$ million.

[41] This is not an important assumption, but it allows us to quantify the value of binding precisely.

[42] In practice our theories contain an infinite number of functions; we discretize here for illustrative purposes only.

Proof. By assumption, the sets $\mathcal{F}_{T_i} \circ B_P$ are pairwise disjoint, and each has G elements, since B_P maps $\mathbb{R}^p$ onto $\mathbb{R}^k$, so $g \mapsto g \circ B_P$ is one-to-one. In (a), every $g \circ B_P$ with $g \in \mathcal{F}_{T_i}$ for some i is consistent with what we know, and nothing else is, so the smallest set we know f^* belongs to is $\mathcal{F}^{(a)}$, the union of $M \cdot p!/(p-k)!$ disjoint sets of G functions each: $|\mathcal{F}^{(a)}| = MG \cdot p!/(p-k)!$. In (b), only the i with $\mathcal{F}_{T_i} = \mathcal{F}_{T^*}$ remains, so the smallest set is $\mathcal{F}^{(b)}$, the union of $p!/(p-k)!$ disjoint sets of G functions: $|\mathcal{F}^{(b)}| = G \cdot p!/(p-k)!$. In (c), only the P with $B_P = B_{T^*,C}$ remains, so the smallest set is $\mathcal{F}^{(c)}$: $|\mathcal{F}^{(c)}| = G$. Dividing gives the two ratios. For $k \geq 1$, $p!/(p-k)! \geq p$, which goes to infinity with p while M stays fixed.

C Function search: further details

This appendix collects the details of Steps 2 and 3 of the induction algorithm (Sections 4.4–4.5): the variables and operations, the genetic algorithm, the random forest inside the search, what happens after the search, and the search with no words.

C.1 The variables and operations

This appendix lists the variables and operations of Section 4.4 in full: the variables, the families of variables the aggregates act on, and the operations.

In our implementation, some operations act on a *family* of variables with a common meaning, such as all past lags of a revision, and return one number, such as their mean or minimum. We use families so that aggregates act only on collections of variables that make sense together: we can take the trend of a firm's last eight quarters of earnings, but it makes no sense to take the trend across earnings, price, and a forecast. For example, for any given variable, we create a family that lists all of its past lags; we also create families for forecasts of different horizons — the one-, two-, and three-quarter-ahead forecasts at any given time are also a family. The variables are listed in Table A4 and the families in Table A5.

The full set we include is the standard algebraic operations ($+$, $-$, $\times$, $\div$), log, exponentiation, square, sign, absolute value, threshold devices (a gate $\mathbf{1}\{a \geq b\}$ and a smooth version of it), and a deadzone (which returns zero inside a band around zero and responds linearly beyond it). On families we allow these aggregates: mean, max, min, standard deviation, range, a trend, and a geometrically weighted average. Table A6 gives the formula for each.

Group	Count	Definition
EPS_quarter_*	20	announced quarterly EPS, lags 0–12 quarters and 1–7 years
EPS_annual_*	8	announced annual EPS, lags 0–7 years
F_q{1--3}_*	15	consensus forecast level, quarterly horizons $q1$–$q3$, monthly snapshots 0–12 months back
F_y{1--2}_*	18	consensus forecast level, annual horizons $y1$–$y2$, monthly snapshots 0–12 months back
Fq{1--3}_*	36	consensus as of past quarter-slots, horizons $q1$–$q3$, lags 1–12 quarters
Fy{1--2}_*	13	consensus as of past year-slots, horizons $y1$–$y2$, lags 1–7 years
rev_q_*	12	consensus revision at the quarter-slot, lags 1–12 quarters
rev_y_*	3	consensus revision at the year-slot, lags 1–3 years
drev_q{1--3}_*	12	month-over-month consensus revision, quarterly horizons, snapshots 0–12 months back
drev_y{1--2}_*	16	month-over-month consensus revision, annual horizons, snapshots 0–12 months back
fe_q_*	12	past forecast error, realized minus consensus, lags 1–12 quarters
fe_y_*	6	past forecast error, annual, lags 1–6 years
meanmed_*	33	analyst mean minus median, per horizon, snapshots 0–12 months back
stdev_*	32	analyst-forecast standard deviation, per horizon, snapshots 0–12 months back
range_*	33	analyst-forecast high minus low, per horizon, snapshots 0–12 months back
numest_*	33	analysts in the consensus, per horizon, snapshots 0–12 months back
numup_*	33	analysts revising up since the prior month, snapshots 0–12 months back
numdown_*	33	analysts revising down since the prior month, snapshots 0–12 months back
prc_*	12	share price, lags 1–12 months
ret_*	12	monthly return, lags 1–12 months
theory_adaptive_*	20	adaptive-expectations predicted revision, quarterly and annual lags
theory_constant_*	21	constant-gain-learning predicted revision, quarterly and annual lags
theory_diagnostic_*	13	diagnostic-expectations predicted revision, quarterly lags
theory_fire_*	14	FIRE predicted revision, quarterly and annual lags
theory_sticky_*	16	sticky-expectations predicted revision, quarterly and annual lags
yhat_*	45	machine-learning conditional-expectation estimates of future earnings, per horizon, quarterly and annual lags
Total	**521**	grouped into 74 ordered families (lag ladders and fixed-horizon ladders)

Notes. The 521 columns the search may draw on, grouped into 74 ordered families (lag ladders and fixed-horizon ladders). Admission is AFTER the missing-data screen: of the full feature set, a column enters only if it is non-missing on at least half of the evaluation sample (or is control-exempt); the survivors' remaining missing entries are zero-filled. These are the variables for the quarterly target: consensus snapshot groups are anchored at the monthly IBES statement date, and the target-slot suffix carried by the panel's column names is dropped here. No future-dated columns and no cross-firm information enter. Source: the replication package.

Table A4: The variables.

Variable	Families	Definition (each family ordered newest to oldest)
Panel A: Lag-ladder families — one variable walked back through its lags (63)		
announced EPS	3	quarterly EPS over lags 0–12 quarters and 1–7 years; annual EPS over lags 0–7 years
consensus monthly snapshot	5	one family per horizon $\{q1, q2, q3, y1, y2\}$, over monthly snapshots going back at a fixed announcement date
consensus quarterly/annual snapshots	5	the same, but at a fixed horizon rather than a fixed announcement date: $\{q1, q2, q3\}$ over lags 1–12 quarters; $\{y1, y2\}$ over lags in years
quarterly/annual revisions	2	at fixed horizons: the $q2 \to q1$ revision, lags 1–12 quarters; the $y2 \to y1$ revision, lags 1–3 years
monthly revision	4	month-over-month revision at a fixed announcement date, one family per horizon $\{q1, q2, y1, y2\}$, over monthly snapshots
forecast error	2	realized minus consensus, lags 1–12 quarters; annual, lags 1–6 years
mean − median	5	one family per horizon $\{q1, q2, q3, y1, y2\}$, over monthly snapshots
dispersion (st. dev.)	5	one family per horizon $\{q1, q2, q3, y1, y2\}$, over monthly snapshots
range (high − low)	5	one family per horizon $\{q1, q2, q3, y1, y2\}$, over monthly snapshots
number of analysts	5	one family per horizon $\{q1, q2, q3, y1, y2\}$, over monthly snapshots
analysts revising up	5	one family per horizon $\{q1, q2, q3, y1, y2\}$, over monthly snapshots
analysts revising down	5	one family per horizon $\{q1, q2, q3, y1, y2\}$, over monthly snapshots
share price	1	lags 1–12 months
monthly return	1	lags 1–12 months
theory-predicted revision, quarterly	5	one family per theory: adaptive, constant-gain, diagnostic, FIRE, sticky, over lags 0–12 quarters
theory-predicted revision, annual	2	adaptive over lags 0–6 years; constant-gain over lags 0–7 years
$\hat{y}$ (ML estimate)	3	one family per horizon $\{h1, h2, h3\}$, over lags 0–12 quarters
Panel B: Horizon-ladder families — one variable at a fixed lag, walked out across horizons (11)		
consensus monthly snapshot	10	at each fixed snapshot lag 0–4 months: one family across quarterly horizons $q1 \to q3$, one across annual horizons $y1 \to y2$
$\hat{y}$ (ML estimate)	1	the current $\hat{y}$ across horizons $h1 \to h3$
Total	**74**	

Notes. The 74 ordered families the aggregates may act on. A lag-ladder family is one variable of Table A4 at successive lags, ordered newest first; a horizon-ladder family is one variable at a fixed lag across forecast horizons, ordered nearest horizon first. Family lengths vary (2–13 members) because members are drawn from the 521 admitted columns: a member that fails the missing-data screen drops out of its family. Source: the replication package.

Table A5: The families of variables.

	Definition
Panel A: Operations	
`add`(a, b)	$a + b$
`sub`(a, b)	$a - b$
`mul`(a, b)	$a \cdot b$
`pdiv`(a, b)	a/b if $\lvert b\rvert \geq 10^{-6}$, else 1
`gate`(a, b)	$\mathbf{1}\{a \geq b\}$
`sign`(a)	$\text{sign}(a) \in \{-1, 0, 1\}$
`abs`(a)	$\lvert a\rvert$
`square`(a)	a^2
`exp`(a)	e^a (non-finite results become missing)
`log`(a)	$\log\big(\max(a,\, e^{-10})\big)$
`sgate`$(a;\, c, s)$	$\sigma\big((a-c)/\lvert s\rvert\big) = \dfrac{1}{1+e^{-(a-c)/\lvert s\rvert}}$; $\lvert s\rvert \to 0$ recovers the hard gate
`dead`$(a;\, c)$	$\text{sign}(a) \cdot \max(\lvert a\rvert - \lvert c\rvert,\, 0)$
Panel B: Aggregators over a family $F = (x_t, x_{t-1}, \ldots, x_{t-m})$, age $a = 0$ newest	
`agg_mean`(F, k)	mean over the k most recent members $\{x_{t-a}\}_{a<k}$, $2 \leq k \leq \lvert F\rvert$
`agg_max`(F, k)	$\max_{a<k} x_{t-a}$
`agg_min`(F, k)	$\min_{a<k} x_{t-a}$
`agg_trend`(F, k)	$\hat{\beta}$ from the per-row OLS of $\{x_{t-k+1}, \ldots, x_t\}$ on the time index $0, \ldots, k-1$
`agg_std`(F, k)	$\sqrt{\frac{1}{k}\sum_{a<k}(x_{t-a} - \bar{x})^2}$
`agg_range`(F, k)	$\max_{a<k} x_{t-a} - \min_{a<k} x_{t-a}$
`agg_decay`(F, c)	$\dfrac{\sum_{a=0}^{m} w_a\, x_{t-a}}{\sum_{a=0}^{m} \lvert w_a\rvert}$, $w_a = c^a$ if $c \geq 0$; $w_0 = 1$, $w_a = -\lvert c\rvert^a$ $(a \geq 1)$ if $c < 0$; $\lvert c\rvert \leq 50$

Notes. Every searched expression is a composition of the Panel-A operations over the admitted variables and the Panel-B aggregates of their ordered families ($|F|$ = family length; missing members are dropped and weights renormalized row by row; $c \to 0$ collapses `agg_decay` to the newest member, $c = 1$ to the plain mean, $c < 0$ puts negative weight on the entire past). All operations propagate missing values and never emit infinities. No operators beyond this list exist in the search space. Source: the replication package.

Table A6: The operations and aggregates.

C.2 The genetic algorithm

This appendix describes the genetic algorithm behind the symbolic regression of Section 4.4.

The formal objective of symbolic regression here is to maximize fit subject to a node budget,

$$\max_{f \in \mathcal{S}} \widehat{R^2}(f) \quad \text{s.t.} \quad \text{nodes}(f) \leq k,$$

where nodes(f) counts the variables and operations in f.

Symbolic regression has three steps. First, it produces a functional form with free constants:

$$a_1\, Y_{i,t} \;-\; a_2 \,\text{sign}\big(F_{t-1}[Y_{i,t}] - a_3\big).$$

Second, it fills in the constants so that the function actually produces concrete predictions:

$$0.3\, Y_{i,t} \;-\; 1.2 \,\text{sign}\big(F_{t-1}[Y_{i,t}] - 0.4\big).$$

Third, once a function f is in hand, it scores f on $\widehat{R^2}(f)$, among functions with nodes$(f) \leq k$.

Symbolic regression then needs to decide how to fill in the first two steps. The second is simple: we can easily optimize the constants $a_1, a_2, \ldots$ because the space of constants is continuous. The first is trickier, because the space is discrete. Symbolic regression typically uses a *genetic algorithm*: that is, it keeps a population of candidate forms over a sequence of many generations, and aims to improve the candidates with each generation. Its moves are:

1. **Mutation.** A candidate functional form is altered randomly: a variable is replaced randomly by another, or an entire set of variables is moved onto a different spine. For instance, a mutation on $a_1\, Y_{i,t} - a_2 \operatorname{sign}(F_{t-1}[Y_{i,t}] - a_3)$ can return $a_1\, Y_{i,t-4} - a_2 \operatorname{sign}(F_{t-1}[Y_{i,t}] - a_3)$: $Y_{i,t}$ has been replaced by $Y_{i,t-4}$.

2. **Crossover.** Two functional forms sharing a spine can produce a child on that same spine: for each variable in the spine, the child inherits the variable from one of the two parents at random, so the child's variables are a mix of its parents' variables. For instance, the parents $a_1\, Y_{i,t} - a_2 \operatorname{sign}(F_{t-1}[Y_{i,t}] - a_3)$ and $b_1\, \Delta F_{i,t-1} - b_2 \operatorname{sign}(Y_{i,t-4} - b_3)$ share the spine $c_1 v_1 - c_2 \operatorname{sign}(v_2 - c_3)$; one possible child is $a_1\, Y_{i,t} - a_2 \operatorname{sign}(Y_{i,t-4} - a_3)$.

These two are the only moves that alter existing candidates; each generation is also injected with a few random functional forms, to make sure the search explores broadly.

To ensure each generation improves on the last, after scoring each of the candidates in a given generation, we keep the best scorers and produce new forms from them.[43] The next generation's forms are then produced from this surviving population by mutation and crossover. The full pseudocode is given by Procedure C.2 below.

Procedure: Symbolic regression

For data, a set of variables and operations, the fit measure $\widehat{R^2}(f)$ and a node budget k.

Step 1. Initialize a population of functional forms from $\mathcal{S}$, constants left free.

Step 2. For each form, fit the constants with a continuous optimizer, making it a literal function f.

Step 3. Score each f on $\widehat{R^2}(f)$; discard forms with more than k nodes.

Step 4. Keep the best scorers as the surviving population.

Step 5. Produce new forms from the surviving population by mutation and crossover, plus a few fresh random forms; return to Step 2 and iterate for a fixed number of generations.

Step 6. Return the best-scoring function.

C.3 The bindability check

This appendix gives the details behind the bindability check of Section 4.5.

[43] More specifically, what is kept is a *population* of fixed size: the candidates with the highest $\widehat{R^2}$ among those within the node budget.

The LLM bindability check. As in Figure 7, the proposer reads f and proposes a small set of descriptions that might explain it, and the binder reads each description together with our context's description D_C and, without seeing f, writes down the class of functions the description implies; f is deemed bindable if one of these classes contains it, and unbindable otherwise. To be generous to the binder, f counts as contained in a class the binder writes down when the two have the same shape and the same set of variables; the constants and the ordering of the variables are free.

A fast bindability checker. Inside the search, we use a random forest that approximates the LLM verdicts based on features of the function (Figure A3). We generate many example functions, label each with the two LLM calls, and train the random forest to predict the label; its prediction $\hat{p}_{\text{bind}}(f)$ is the estimated probability that f is deemed bindable. $\hat{p}_{\text{bind}}$ is fast enough to embed directly inside the search, so we can have symbolic regression solve

$$\widehat{\arg\max}_{f \in \mathcal{S}} \widehat{R^2}_{\hat{m}}(f) \quad \text{s.t.} \quad \hat{p}_{\text{bind}}(f) \geq 0.50.$$

The rest of this subsection gives exactly how the example functions are sampled and how the random forest is built.

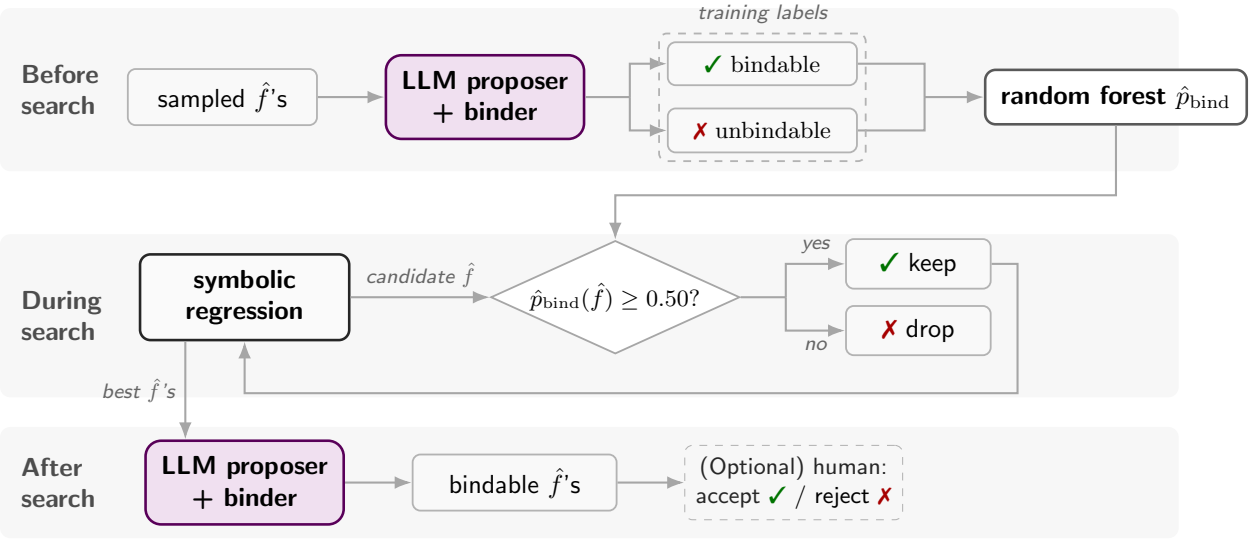


Figure A3: Checking for bindability. Before the search, we train a random forest $\hat{p}_{\text{bind}}$ on sampled functions that the LLM proposer and binder label bindable or unbindable. During the search, symbolic regression keeps a candidate $\hat{f}$ only if $\hat{p}_{\text{bind}}(\hat{f}) \geq 0.50$, and only kept candidates breed the next generation. After the search, the LLM proposer and binder check the best $\hat{f}$'s again and return the bindable ones, which the researcher can optionally accept or reject.

Spines. The main reason why most functions built from the variables and operations are not bindable is that most functional shapes do not correspond to any theory: for instance, $v_1 v_2 v_3 + v_4 / v_5$ is unlikely to correspond to any theory regardless of how we fill in the different v's. We decompose a function f into a spine and a fill, $f = s(f)\langle v(f)\rangle$, where $s(f)$ is the shape of the function (with variables taken out) and $v(f)$ contains the list of variables and families (without the functional form). Concretely, for $f = Y_{i,t} - F_{t-1}[Y_{i,t}]$, the spine is $s(f) = v_1 - v_2$ and the fill is $v(f) = (Y_{i,t},\, F_{t-1}[Y_{i,t}])$. We first prespecify a set of spines that an economic mechanism could plausibly describe, for instance $v_1 - v_2$ or $v_1(v_2 - v_3)$, and set $\hat{p}_{\text{bind}}(f) = 0$ whenever the spine of f is not in

that set. Our sampling procedure is then to draw a spine from the set and fill its slots at random. A single-slot spine — a bare variable or a bare aggregate — is automatically deemed bindable. Spines carry no unary operations (i.e. functions of one variable like log or exp). Taking the log of a variable just measures the same object differently, so folding operations like log, exp and $|\cdot|$ into the shape would multiply the set of spines without adding much structure. Instead we impose a budget of two unary operations per candidate to enforce simplicity. The full list of spines is in Table A7.

Labels. Each sampled $f = s\langle v \rangle$ is then labeled by running the proposer and binder, which deem it bindable or unbindable. Each call also yields training data beyond the originally sampled f. Running the proposer and binder on f typically binds back functions f' other than f, because the proposer casts a wide net — but still subject to the condition that it only names descriptions that are very likely to be theories. What that means is that every f' the binder returns is likely bindable, and can therefore be labeled bindable. Though f' was not itself labeled by the proposer and binder, the condition we are approximating is that some theory binds to it, which is true of f'.[44] We give some examples of labels for different fills in Table A8.

The random forest approximation. Repeated sampling gives a collection of labeled functions, which we split into a training set $\mathcal{E}_T$ and a validation set $\mathcal{E}_V$. With training labels in hand, our final step is to learn from them. This is an ordinary supervised learning problem: we fit $\hat{p}_{\text{bind}}$ on $\mathcal{E}_T$ with a random forest, using various features of the filled variables. The full feature list is in Table A9. $\hat{p}_{\text{bind}}$ is then evaluated on the held-out $\mathcal{E}_V$, where it reaches an AUC of 0.959 and an accuracy of 86.7%; Table A10 reports both on the training and held-out examples.

C.4 After the search: the LLM projection and selection

This appendix writes out what happens after the search in each round.

Why an LLM projection. A function with a high $\hat{p}_{\text{bind}}$ is likely, but not always, bindable: $\hat{p}_{\text{bind}}$ is after all only a proxy for the LLM calls. So at the end of the search we run the LLM proposer and binder on the best functions the search found, this time not as a bindability checker but as a "projection" that distills each $\hat{f}$ into its bindable essence: the proposer describes what in $\hat{f}$ a mechanism can explain, and the binder writes the class of functions that description implies (Figure A3). For example, if $\hat{f} = 0.4\left(\hat{\mathbb{E}}_t[Y_{i,t+1}] - F_{t-1}[Y_{i,t+1}]\right) + 0.03\, F_{t-4}[Y_{i,t-3}]$, the proposer describes the first term as sticky updating toward the machine-learning forecast, and the binder returns $\beta\left(\hat{\mathbb{E}}_t[Y_{i,t+1}] - F_{t-1}[Y_{i,t+1}]\right)$, thereby pruning away the second term.[45]

The output of the search. The search does not only return the $\widehat{\arg\max}$ of $\widehat{R^2}_{\hat{m}}(f)$ subject to $\hat{p}_{\text{bind}}(f) \geq 0.50$ of Section 4.5. It actually produces the whole Pareto frontier: the set of candidates that no other candidate beats on both $\widehat{R^2}_{\hat{m}}$ and $\hat{p}_{\text{bind}}$ simultaneously,

$$\hat{\Pi} \;=\; \widehat{\text{Pareto}}_{f \in \mathcal{S}}\Big(\widehat{R^2}_{\hat{m}}(f) \;\; \text{against} \;\; \hat{p}_{\text{bind}}(f)\Big).$$

[44] A general principle in machine learning: it is typically better to have more training data, even if the training data is a bit noisy.

[45] The best functions here are the Pareto frontier of $\widehat{R^2}_{\hat{m}}$ against $\hat{p}_{\text{bind}}$, not only the best function with $\hat{p}_{\text{bind}} \geq 0.50$.

The LLM projection. For a search output $\hat{f}$ in the Pareto set $\hat{\Pi}$, the proposer first proposes some descriptions $\hat{D}_1, \hat{D}_2, \ldots$, and the binder writes the class of functions $\hat{\mathcal{F}}_{\hat{D}_j, C}$ each description implies. Refitting the constants then gives the bindable functions that come back:

$$\hat{f}_j = \arg\max_{f \in \hat{\mathcal{F}}_{\hat{D}_j, C}} \widehat{R^2}_{\hat{m}}(f).$$

Doing this for every $\hat{f}$ in $\hat{\Pi}$ gives a collection $\mathcal{M}$ of pairs $(\hat{D}_j, \hat{f}_j)$ of a description and a bindable function.

Selection. The researcher then selects from $\mathcal{M}$ by picking the highest-scoring pair, but can reject a pair that is not bindable, or prefer a much simpler one at nearly the same fit, as discussed in Section 4.5.

Spine	Slots	Template (schematic)
single_col	v	x_1
single_agg	A	A_1
gap_vv	v, v	$x_1 - x_2$
gap_av	A, v	$A_1 - x_1$
gap_aa	A, A	$A_1 - A_2$
sum3	v, v, v	$x_1 + (x_2 - x_3)$
ratio_vv	v, v	$\frac{x_1}{x_2}$
ratio_av	A, v	$\frac{A_1}{x_1}$
ratio_va	v, A	$\frac{x_1}{A_1}$
ratio_gap_v	v, v, v	$\frac{(x_1 - x_2)}{x_3}$
ratio_v_floor_v	v, v	$\frac{x_1}{(c_1 + x_2)}$
ratio_a_floor_v	A, v	$\frac{A_1}{(c_1 + x_1)}$
ratio_v_floor_a	v, A	$\frac{x_1}{(c_1 + A_1)}$
ratio_a_floor_a	A, A	$\frac{A_1}{(c_1 + A_2)}$
ratio_gap_floor_v	v, v, v	$\frac{(x_1 - x_2)}{(c_1 + x_3)}$
prod_vv	v, v	$x_1 \cdot x_2$
prod_av	A, v	$A_1 \cdot x_1$
prod_aa	A, A	$A_1 \cdot A_2$
factored_gap	v, v, v	$(x_1 - x_2) \cdot x_3$
gated_self_v	v	$x_1 \cdot \mathbf{1}\{x_1 \geq c_1\}$
gated_other_v	v, v	$x_1 \cdot \mathbf{1}\{x_2 \geq c_1\}$
gated_self_a	A	$A_1 \cdot \mathbf{1}\{A_1 \geq c_1\}$
gated_other_a	A, v	$A_1 \cdot \mathbf{1}\{x_1 \geq c_1\}$
gated_gap	v, v, v	$(x_1 - x_2) \cdot \mathbf{1}\{x_3 \geq c_1\}$
one_sided_gap	v, v	$(x_1 - x_2) \cdot \mathbf{1}\{(x_1 - x_2) \geq c_1\}$
asym_triple_v	v, v	$x_1 \cdot x_2 \cdot \mathbf{1}\{x_2 \geq c_1\}$
asym_triple_a	A, v	$A_1 \cdot x_1 \cdot \mathbf{1}\{A_1 \geq c_1\}$
sgated_self_v	v	$x_1 \cdot \sigma\left(\frac{x_1 - c_1}{c_2}\right)$
sgated_other_v	v, v	$x_1 \cdot \sigma\left(\frac{x_2 - c_1}{c_2}\right)$
sgated_self_a	A	$A_1 \cdot \sigma\left(\frac{A_1 - c_1}{c_2}\right)$
sgated_other_a	A, v	$A_1 \cdot \sigma\left(\frac{x_1 - c_1}{c_2}\right)$
sgated_gap	v, v, v	$(x_1 - x_2) \cdot \sigma\left(\frac{x_3 - c_1}{c_2}\right)$
sgated_one_sided_gap	v, v	$(x_1 - x_2) \cdot \sigma\left(\frac{(x_1 - x_2) - c_1}{c_2}\right)$
sgated_asym_triple_v	v, v	$x_1 \cdot x_2 \cdot \sigma\left(\frac{x_2 - c_1}{c_2}\right)$
sgated_asym_triple_a	A, v	$A_1 \cdot x_1 \cdot \sigma\left(\frac{A_1 - c_1}{c_2}\right)$
gated_by_agg_v	v, A	$x_1 \cdot \mathbf{1}\{A_1 \geq c_1\}$

Spine	Slots	Template (schematic)
sgated_by_agg_v	v, A	$x_1 \cdot \sigma\left(\frac{A_1 - c_1}{c_2}\right)$
gated_by_agg_a	A, A	$A_1 \cdot \mathbf{1}\{A_2 \geq c_1\}$
sgated_by_agg_a	A, A	$A_1 \cdot \sigma\left(\frac{A_2 - c_1}{c_2}\right)$
damped_other_v	v, v	$x_1 \cdot (c_1 + c_2 \cdot \mathbf{1}\{x_2 \geq c_3\})$
sdamped_other_v	v, v	$x_1 \cdot (c_1 + c_2 \cdot \sigma\left(\frac{x_2 - c_3}{c_4}\right))$
damped_self_v	v	$x_1 \cdot (c_1 + c_2 \cdot \mathbf{1}\{x_1 \geq c_3\})$
sdamped_self_v	v	$x_1 \cdot (c_1 + c_2 \cdot \sigma\left(\frac{x_1 - c_3}{c_4}\right))$
damped_agg	A, v	$A_1 \cdot (c_1 + c_2 \cdot \mathbf{1}\{x_1 \geq c_3\})$
sdamped_agg	A, v	$A_1 \cdot (c_1 + c_2 \cdot \sigma\left(\frac{x_1 - c_3}{c_4}\right))$
damped_self_a	A	$A_1 \cdot (c_1 + c_2 \cdot \mathbf{1}\{A_1 \geq c_3\})$
sdamped_self_a	A	$A_1 \cdot (c_1 + c_2 \cdot \sigma\left(\frac{A_1 - c_3}{c_4}\right))$
prod_ratio_v_av	v, A, v	$x_1 \cdot \frac{A_1}{x_2}$
prod_ratio_a_av	A, A, v	$A_2 \cdot \frac{A_1}{x_1}$
prod_ratio_v_aa	v, A, A	$x_1 \cdot \frac{A_1}{A_2}$
prod_ratio_v_vv	v, v, v	$x_3 \cdot \frac{x_1}{x_2}$
prod_ratio_v_va	v, v, A	$x_1 \cdot \frac{x_2}{A_1}$
prod_ratio_a_vv	A, v, v	$A_1 \cdot \frac{x_1}{x_2}$
ratio_prod_vv_v	v, v, v	$\frac{x_1 \cdot x_2}{x_3}$
ratio_prod_av_v	A, v, v	$\frac{A_1 \cdot x_1}{x_2}$
prod_ratio_gap_v	v, v, v, v	$x_4 \cdot \frac{(x_1 - x_2)}{x_3}$
ratio_v_floor_ratio_vv	v, v, v	$\frac{x_1}{(c_1 + \frac{x_2}{x_3})}$
ratio_v_floor_ratio_av	v, v, A	$\frac{x_1}{(c_1 + \frac{A_1}{x_2})}$
self_gated_ratio_vv	v, v, v	$x_3 \cdot \mathbf{1}\{\frac{x_1}{x_2} \geq c_1\} \cdot \frac{x_1}{x_2}$
self_gated_ratio_av	A, v, v	$x_2 \cdot \mathbf{1}\{\frac{A_1}{x_1} \geq c_1\} \cdot \frac{A_1}{x_1}$
self_gated_ratio_aa	A, A, v	$x_1 \cdot \mathbf{1}\{\frac{A_1}{A_2} \geq c_1\} \cdot \frac{A_1}{A_2}$
gated_ratio_vvv	v, v, v, v	$x_4 \cdot \mathbf{1}\{x_3 \geq c_1\} \cdot \frac{x_1}{x_2}$
gated_ratio_av	A, v, v, v	$x_3 \cdot \mathbf{1}\{x_2 \geq c_1\} \cdot \frac{A_1}{x_1}$
gated_ratio_av_by_agg	A, A, v, v	$x_2 \cdot \mathbf{1}\{A_2 \geq c_1\} \cdot \frac{A_1}{x_1}$
gated_by_own_ratio_v	v, v	$x_2 \cdot \mathbf{1}\{\frac{x_2}{x_1} \geq c_1\}$
gated_by_own_ratio_a	A, v	$A_1 \cdot \mathbf{1}\{\frac{A_1}{x_1} \geq c_1\}$
gated_prod_v	v, v, v	$x_1 \cdot x_2 \cdot \mathbf{1}\{x_3 \geq c_1\}$
one_sided_gap_av	A, v	$(A_1 - x_1) \cdot \mathbf{1}\{(A_1 - x_1) \geq c_1\}$
damped_by_ratio_vv	v, v, v	$x_3 \cdot (c_1 + c_2 \cdot \frac{x_1}{x_2})$
damped_by_ratio_av	A, v, v	$x_2 \cdot (c_1 + c_2 \cdot \frac{A_1}{x_1})$
ratio_self_gated_num_v	v, v, v	$x_3 \cdot \frac{x_2 \cdot \mathbf{1}\{x_2 \geq c_1\}}{x_1}$
ratio_self_gated_num_a	A, v, v	$x_2 \cdot \frac{A_1 \cdot \mathbf{1}\{A_1 \geq c_1\}}{x_1}$

Notes. Schematic templates: x_i = single admitted columns (v-slots, in slot order), A_i = aggregates over an ordered family (A-slots), c_i = free constants fitted per expression, σ = the logistic function, $\mathbf{1}\{\cdot\}$ = the hard gate (protected division throughout: a tiny denominator returns 1). Templates are rendered from each spine's build function and are schematic in that they omit the fitted outer coefficient and the optional unary/dead-zone decorations (at most two) on slot terms. What is absent bounds the search: any mechanism requiring a shape not in this list — nested gates, deeper ratio towers, more than four interacting terms, time-varying coefficients — was unreachable by construction. Source: the replication package (72 spine templates).

Table A7: The prespecified spines.

Examples of labels for different fills, grouped by spine

Panel A. Spine `gap_vv`: $c\,(x_1 - x_2)$	
$\ell = 1$ *(a theory binds)*	
$x_1 = F_{t-1}[Y_{i,t}],\ \ x_2 = Y_{i,t}$	"Revisions respond to the gap between the prior consensus and the earnings it forecast."
$x_1 = Y_{i,t},\ \ x_2 = Y_{i,t-4}$	"Revisions respond to year-over-year growth in announced earnings (same fiscal quarter, prior year)."
$\ell = 0$ *(nothing binds)*	
$x_1 = \text{price}_{i,t},\ \ x_2 = \text{numup}_{i,t}$	The two arguments are in different units — one is a price, one is a count — so their difference means nothing.
$x_1 = Y_{i,t-3},\ \ x_2 = F_{t-12}[Y_{i,t-11}]$	They have the same units, but the realization and the forecast are two years apart, and so no theory would produce this function.
Panel B. Spine `ratio_v_floor_v`: $x_1/(c_1 + x_2)$	
$\ell = 1$ *(a theory binds)*	
$x_1 = Y_{i,t} - F_{t-1}[Y_{i,t}],\ \ x_2 = \text{disp}_{i,t-1}$	"The response to the latest forecast error shrinks when analysts disagree."
$x_1 = Y_{i,t} - F_{t-1}[Y_{i,t}],\ \ x_2 = \text{sd}\big(Y_{i,t-7}, \ldots, Y_{i,t}\big)$	"The response to the latest forecast error shrinks when recent fundamentals are volatile."
$\ell = 0$ *(nothing binds)*	
$x_1 = \text{ret}_{i,t-3},\ \ x_2 = \text{price}_{i,t-1}$	Damping a return by the price level: a return is already unitless, so dividing by price does not correspond to any precision or attention state — it is not clear what the expression is doing.
$x_1 = \text{coverage}_{i,t-6},\ \ x_2 = \text{ret}_{i,t-9}$	A stale coverage count damped by an old return: neither a revision-relevant signal nor a plausible damping state.

Notes. Each panel is one spine (a function shape found by the search); a fill assigns a panel column to every slot, written out as $x_1, x_2 = \ldots$ above. Within a panel, the $\ell = 1$ block lists fills to which the proposer attaches a theory (with the recovered one-line description), and the $\ell = 0$ block lists fills of the same spine to which nothing binds (with the reason). Bindability is a property of the fill, not the shape. Quantities in the notation of section 2 — $F_t[\cdot]$ the consensus, $Y_{i,t}$ realized EPS, every variable firm-indexed. Realized variables run through t; forecast-derived variables (consensus, dispersion, coverage) run through the prior period, matching the information available when the revision is made. Verdicts and descriptions are illustrative, hand-authored examples.

Table A8: Examples of labels for different fills.

Feature group	Count	Definition
Panel A: Features of the fill (131, the random forest's inputs)		
slot roles	10	one feature per role (consensus level, consensus revision, analyst count, dispersion, EPS level, forecast error, ML estimate, price, return, theory-predicted revision); the feature's value is the number of slots in the fill occupied by a variable of that role
slot units	4	one feature per unit (count, earnings per share, price, return); its value is the number of slots occupied by a variable carrying that unit
slot recency	4	one feature per recency bucket (lag 0, lags 1–3, lags 4+, no lag); its value is the number of slots whose variable's newest lag falls in that bucket
slot family type	3	one feature per kind of family (lag ladder; horizon ladder; machine/theory family, i.e. the $\hat{y}$ and theory-predicted-revision walks); its value is the number of slots occupied by a family of that kind
family hash buckets	16	one feature per bucket: every slot occupied by a family has that family's name hashed into a bucket, and feature b is the number of such slots landing in bucket b
column hash buckets	32	the same for slots occupied by a single column, with more buckets
role co-occurrence	55	one feature per unordered pair of roles (incl. a role with itself); its value is the number of pairs of slots in the fill whose roles form that pair
cross-slot structure	6	six scalars of the whole fill: the number of slots; the number of slots holding ordered families; the number holding unordered families; a 0/1 flag that every slot shares one unit; the number of distinct units; the number of distinct roles
units precheck flag	1	a single 0/1 flag: the fill combines units in a way no theory would (e.g. price minus count)
Panel B: Features of the spine (14, structure only)		
size and slots	5	five scalars of the whole spine: its node count; its number of slots; the number of its slots typed for columns, for ordered families, and for unordered families
operation counts	7	one feature per operation (the aggregates max, min, mean, trend; the gate; protected division; multiplication); its value is the number of times that operation occurs in the spine
polynomial shape	2	two scalars of the whole spine: its number of monomials, and its highest monomial degree
Total	**145**	

Notes. Panel A is the 131-feature vector the bindability random forest consumes: per-slot semantic features of the fill, aggregated to a fixed length, with the spine entering through the slot structure and the precheck. Panel B is the structure-only description of the spine itself. The forest is a calibrated random forest (sigmoid calibration, seed 0); label $\ell = 1$ (bindable) is the positive class. Source: the replication package.

Table A9: The features behind $\hat{p}_{\text{bind}}$: the fill features its random forest consumes (Panel A) and the structure-only spine features (Panel B).

	Examples	Positive rate ($\ell = 1$)	AUC	Accuracy at 0.50
Training set	5,253	51.8%	0.997	97.4%
Held-out set	60	50.0%	0.959	86.7%

Notes. The bindability gate: a calibrated random forest predicting whether the proposer and binder (expression → description → expression) attach a theory to a candidate fill. The positive class is $\ell = 1$ (a theory binds); the positive rate is the share of examples with that label. The held-out set is a randomly selected split of the generated examples. Source: the replication package.

Table A10: The bindability gate, on training and held-out examples.

	Recovered description	$\hat{R}^2$, fold $f5$	Status
1	**The forecaster continues the correction for its own habitual anchoring on the stale prior consensus: because consensus adjusts only partially toward new information, the unabsorbed part of past news keeps pushing earnings in the same direction, so the update leans the same way as the accumulated partial-adjustment gap, weighting the most recent observations most heavily while still drawing on a run of several forecast horizons; and the catch-up is applied more forcefully when the correction implied by the own past forecast error says the consensus has been running optimistic, since evidence of prior over-optimism makes the forecaster trust the unwinding of stale anchoring more.**	**0.529**	**picked**
2	The forecaster carries forward the sticky partial-adjustment correction toward the prior consensus, averaging it over the last few forecast horizons with most weight on the freshest reading because recent under-adjustment is the most reliable guide to how much catching-up is still owed, and it applies this correction more forcefully for firms whose annual consensus earnings expectation of about two years earlier stood comfortably above a modest positive level — for such established, clearly profitable firms a given fraction of unabsorbed news is worth more per share, while for marginal or near-zero-earnings firms only the baseline correction is made.	0.501	
3	The forecaster completes the catch-up implied by sticky, partial adjustment to the prior consensus, using the persistence of that under-adjustment across recent forecast horizons with recent horizons weighted most, and scales the correction inversely to the recent stock return because market strength signals that the news has already been priced and reflected while market weakness signals the consensus is still stale.	0.503	
4	The forecaster continues the unfinished portion of earlier partial adjustments: because consensus moves toward fresh information only fractionally each period, the direction of the recent anchored partial-adjustment corrections predicts further revision the same way, with a decay-weighted memory that leans on the most recent corrections but still draws on a run of several forecast horizons, and the continuation is applied only where the prior consensus for the current target formed about one horizon ago points to profit rather than loss, since for loss-making situations the anchor itself carries no reliable information to keep closing toward.	0.492	
5	The forecaster continues the incomplete adjustment embodied in the sticky partial-adjustment revision term, averaging it over several forecast horizons with the most recent readings weighted most heavily, and takes a larger step when the machine expectation of the nearest-horizon target has been drifting down over recent horizons than when it has been rising.	0.492	
6	The forecaster continues the incomplete adjustment embodied in the sticky partial-adjustment revision term — the part of the news that the prior consensus has not yet absorbed — averaging that signal over several forecast horizons with the most recent readings weighted most heavily, ignoring readings too small to be distinguishable from noise, and sizing the step by the direction of the machine expectation of the nearest-horizon target: when that statistical expectation has been drifting down over the last few horizons the catch-up is taken more aggressively, because a deteriorating fundamental path makes lagged consensus adjustment more likely to be genuine, while a rising statistical path makes the forecaster hold back.	0.493	
7	The forecaster continues the correction for anchoring on the stale prior consensus, treating the unabsorbed part of past news as still to be worked into earnings, and forms that catch-up from a recency-weighted history of such partial-adjustment gaps spanning a run of several forecast horizons rather than from the latest one alone.	0.491	
8	The forecaster finishes the adjustment that sticky anchoring on the prior consensus left undone, using a persistence-weighted blend of that correction across a run of several past forecast horizons rather than the latest reading alone, because repeated under-adjustment signals more reliable unfinished revision than a single one.	0.491	
9	The forecaster treats the under-adjustment implied by anchoring on the prior consensus as something that has not yet worked itself out, and finishes that adjustment in the current update, reading the sticky partial-adjustment correction not from the latest period alone but as a persistence-weighted blend of its readings over a run of several past forecast horizons, so that a correction that keeps reappearing carries more weight than a one-off; and the forecaster acts only once that accumulated correction is more than a negligible amount, leaving essentially closed gaps untouched rather than churning the number.	0.491	
10	The forecaster continues the incomplete adjustment embodied in the sticky partial-adjustment revision term, smoothing it across several forecast horizons with the most recent readings weighted most heavily, and stays put when that smoothed signal is too small to be distinguishable from noise.	0.491	
11	The forecaster moves the consensus further in the direction implied by the partial adjustment anchored on the prior consensus, because such under-adjustment to news tends to persist and to be completed in the following horizon.	0.469	
12	The forecaster carries forward the sticky partial-adjustment correction toward the prior consensus, smoothing it over the most recent few forecast horizons so that persistent incomplete adjustment, rather than a single noisy reading, drives the update.	0.469	
13	The forecaster completes the catch-up implied by sticky, partial adjustment to the prior consensus, drawing on how persistently that under-adjustment has run over recent forecast horizons, and acts only when the accumulated gap is material rather than chasing negligible amounts.	0.469	
14	The forecaster pushes the forecast further in the direction of the current gap between the prior consensus and where fuller information points, on the view that consensus adjusts only partially and the remaining adjustment is still to come.	0.278	
15	The forecaster continues the incomplete adjustment embedded in the prior consensus, revising further in the same direction because consensus updates absorb news only partially.	0.160	
16	The forecaster carries forward the sticky partial-adjustment correction toward the prior consensus, averaging it over the last few forecast horizons with most weight on the freshest reading, and applies it more strongly the higher the firm's earlier long-horizon annual consensus earnings level, since dollar-per-share catch-up scales with the firm's expected earnings power.	0.058	
17	The forecaster carries out the delayed catch-up implied by sticky, partial adjustment to the prior consensus: because the consensus moves only fractionally toward new information, the accumulated unadjusted gap — averaged over recent forecast horizons with the most recent horizons weighted most heavily — is added now, but only once that accumulated gap is large enough to be worth acting on, and its size is set inversely by the stock return of roughly the preceding two-thirds of a horizon, since a market that has already risen is taken to have absorbed the news while a market that has fallen leaves the consensus far more stale; in the weak-return state the response is more than proportional, so large accumulated gaps move the forecast disproportionately.	0.012	

Notes. The menu the researcher selected from in the final round: the Pareto frontier after the LLM projection, not the raw search frontier. Each frontier function that passed the gate ($\hat{p}_{\text{bind}} \geq 0.50$) was described in words by the LLM, bound back to a function, and refit on the fit folds; the table shows each candidate's description and the bound-back function's $\hat{R}^2$. Descriptions and the pick are verbatim from the original search run. $\hat{R}^2$ is re-scored on the current evaluation sample against the ML benchmark $\hat{m}_q$ rather than the forecast revision (coefficients fit on folds $f1$–$f4$, scored on held-out fold $f5$, $N = 57{,}000$ firm–quarters, the same rows for every candidate; candidates are zero-filled onto them). Rows are ordered by the $\hat{R}^2$ recorded during the original search run, on that run's own evaluation sample, so the order can differ slightly from the re-scored column. The top-ranked candidate was picked right away.

Table A11: The final round's selection: the menu after the LLM projection.

D The LLM prompts

The prompts behind the two LLM calls, the binder and the proposer, are shown in Figures A4 and A5. The descriptions of the contexts themselves that the prompts refer to — quarterly and annual — are shown in Figures A8 and A9. The prompts of the portability loop (Section 4.6) are in Figures A6 and A7: the naive LLM call, and the proposer prompt of the portability loop with the previous-attempts block it adds from round 2 on.

Panel A: the binder $\hat{B}$ (general core)

```
BINDING -- GENERAL CORE
(Assemble as: this core + one DOMAIN CONTEXT block. The context defines
the forecast target, the forecast horizon and its periods, the exact
default scaler term, and the variable vocabulary. Nothing in this core
is domain-specific. The prompt operates on ONE description per call;
batching is an implementation concern outside this prompt.)

TASK
You will be given ONE theory description: a single belief-updating
mechanism stated as an update rule. Write EXACTLY ONE concrete
expression implementing it as a bare adjustment term for the forecast
target defined in the DOMAIN CONTEXT -- a term to be ADDED to an
existing forecast as the current update. Binding is a FUNCTION of the
verbal description: one description in, one expression out. Return null
when the description cannot be faithfully implemented with the
vocabulary.

FAITHFULNESS
Implement the COMPLETE mechanism as described -- including any scaling,
gating, or conditioning quantities the description names (they are part
of the mechanism, not something to strip) -- and NOTHING the description
does not state. Do not add drivers, modulators, or transforms beyond
the description plus unit restoration. Every free quantity is a
WILDCARD: tunable constants (thresholds, gains, floors) are written c1,
c2, ...; aggregation windows are written k1, k2, ... -- all are
estimated downstream, so never write numeric guesses for them. Plain
numeric literals are allowed only where a specific number is structural
(e.g. the 0 in a positive-part gate).

GRAMMAR (expressions are machine-parsed; use EXACTLY this syntax)
- Infix arithmetic: +  -  *  /  with standard precedence and
  parentheses; unary minus allowed. '/' is protected division
  (evaluates to 1 when the denominator is ~0).
- gate(a, b): elementwise indicator, 1 if a >= b else 0 -- use for
  one-sided or regime clauses (e.g. the positive part of a signal x is
  gate(x, 0) * x).
- sgate(x, c1, c2): smooth threshold rising from 0 to 1 around c1 with
  softness c2 -- use when the response phases in gradually rather than
  switching on.
- dead(x, c1): dead zone -- zero response inside a band of half-width
  c1 around zero, linear beyond it.
- Unary functions: sign(x), abs(x), square(x) (= x*x), exp(x), log(x).
  log is PROTECTED: the argument is floored at exp(-10), so it is safe
  on zero/negative values. There is NO sqrt.
- Aggregators over the FAMILIES listed in the DOMAIN CONTEXT. Ordered
  families (histories, most-recent first) take a window wildcard:
    agg_mean(FAMILY, k=k1)  agg_max(FAMILY, k=k1)
    agg_min(FAMILY, k=k1)   agg_trend(FAMILY, k=k1)
  (agg_trend = per-period slope, oldest to newest), plus
  agg_std(FAMILY, k=k1) (dispersion) and agg_range(FAMILY, k=k1)
  (max minus min).
- agg_decay(FAMILY, c=c1): exponentially-weighted mean of the family,
  newest observations weighted most; the decay rate is a wildcard.
- Terminals: ONLY column and family names listed in the DOMAIN CONTEXT.
  Any other name is unparseable and the expression will be discarded.

UNIT RESTORATION
The expression must live in the target's units. The DEFAULT SCALER is
the lagged consensus forecast level -- the exact term is designated in
the DOMAIN CONTEXT; use exactly that term. If the mechanism's driving
signal is UNITLESS (a return, a like-unit ratio, a sign or gate
output), restore units by multiplying by the default scaler. If the
signal is in units that are NEITHER target units NOR unitless, first
normalize it unitless by dividing by a natural same-unit denominator of
your choosing, then apply the default scaler. Unit restoration is part
of faithful implementation -- it is NOT an extra mechanism -- and must
never be used to smuggle in additional drivers.

RESPONSE SHAPE (whole-signal only)
When the description states a shaped response, apply ONE shape device
to the WHOLE mechanism kernel, never to an individual component:
normalize the kernel unitless (divide by the default scaler), then
- convex: square(.) -- optionally reattach direction with sign(.) when
  the mechanism is directional;
- asymmetric: exp(.);
- one-sided: gate(., c1) times the kernel (respond only on one side of
  a threshold);
then restore units with the default scaler. Per-component shaping is
unit-incoherent and not allowed.

TIME TRANSLATION
Descriptions state frequency and lookback relative to the forecast
horizon. Frequency language selects the FAMILY: "within-horizon
observations" means the finer-frequency family for the role;
"horizon-frequency" means the family stepping once per horizon.
Lookback windows stay as k wildcards -- estimation fits them -- but when
the description pins a lookback ("about three forecast horizons"),
choose the family whose available depth covers it (periods per horizon
are stated in the DOMAIN CONTEXT).

OUTPUT FORMAT
Return ONLY a JSON object for the description:
{"expr": "<expression string>"}
Use {"expr": null} when nothing faithful can be written. No prose
outside the JSON object.
```

Figure A4: The prompt implementing the binder, which writes $\hat{\mathcal{F}}_{D,C}$.

Panel B: the proposer (general core)

```
INVERSE BINDING -- GENERAL CORE
(Assemble as: this core + one DOMAIN CONTEXT block. The context defines
the forecast target, the forecast horizon, the admissible economic
roles, and the variable vocabulary. Nothing in this core is
domain-specific. The prompt operates on ONE expression per call;
batching is an implementation concern outside this prompt.)

TASK
You will be given ONE candidate expression. The expression g is a bare
ADJUSTMENT TERM for the forecast target defined in the DOMAIN CONTEXT:
it is meant to be ADDED to an existing forecast as the current update --
it is not a level forecast, and no bridging to levels is involved.

The expression arrives with its FITTED numeric constants written out
(not wildcards): the sign and rough magnitude of each constant are part
of the evidence and should inform your reading of the mechanism.

INPUT
Alongside the expression you receive fitted_residual_coefficient: the
expression's own fitted coefficient against the residual target with
controls partialled out. A positive value means the target moves WITH
the expression; the magnitude is on the expression's own scale, so use
it for direction always, and for economic-significance judgment only in
comparison across interpretations of the same expression.

OUTPUT AS A CORRESPONDENCE -- mechanisms, each with a complexity ladder
Return one entry PER GENUINELY DISTINCT MECHANISM the expression could
implement (usually one; more only under real ambiguity of meaning --
never quota-fill). For each mechanism return a LADDER of descriptions,
richest first:
- rung 1: the RICHEST admissible description -- the full mechanism
  including any scaling, gating, or conditioning quantities the
  expression's structure genuinely supports;
- subsequent rungs (up to 3): progressively SIMPLER descriptions of the
  SAME mechanism, each dropping one modulator, conditioning clause, or
  response-shape refinement, ending at the raw kernel.
Every rung must independently satisfy ADMISSIBILITY. Do not invent
modulators the expression does not contain to enrich rung 1.

ADMISSIBILITY
An admissible description:
1. Describes a SINGLE mechanism (one update rule, not a menu). A named
   quantity MAY scale, gate, or condition the response -- stated as one
   coherent rule -- but two independent channels are two mechanisms:
   split them into separate entries.
2. Is written in plain English as an economic mechanism -- WHY the
   forecaster revises this way.
3. Does NOT narrate the algebra. Restating the formula in words is not
   a theory.
4. Contains NO numeric quantities, thresholds, coefficients, or
   variable names.
5. EVERY variable reference names a concrete economic role drawn from
   the DOMAIN CONTEXT's role list, and only that list. Role-free
   placeholders ("a signal", "a characteristic") are banned; if you
   cannot name every input's role, return fewer rungs or mechanisms.

RESPONSE SHAPE
A description MAY state how the response scales with its composite
signal: proportional (default), MORE THAN PROPORTIONAL (convex -- large
signals move the forecast disproportionately), ASYMMETRIC (one sign of
the signal moves the forecast more than the other), or ONE-SIDED (the
forecaster responds only when the signal is on one side of a
threshold). State the shape of the response to the WHOLE signal only --
never attribute curvature to an individual component. Shape is a
property of the one mechanism, not a second mechanism, and dropping it
is a valid ladder step.
Example of an admissible shaped description: "The forecaster revises
with A because <economic reason>, and the response is more than
proportional: large readings of the combined signal move the forecast
disproportionately."

TIME LANGUAGE
State observation frequency and lookback RELATIVE to the forecast
horizon defined in the DOMAIN CONTEXT, never in absolute calendar
units: e.g. "within-horizon observations", "a sustained run of about
three forecast horizons", "information about one horizon old". Naming
the horizon multiple is encouraged when the expression pins it; leave
it unstated when it does not.

NEAREST-MECHANISM FALLBACK
When the WHOLE expression does not implement a single admissible
mechanism, find the LARGEST sub-expression that does, and describe THAT
mechanism (never quoting the sub-expression itself). Return an empty
list only when no part supports any admissible mechanism. Lean toward
naming the nearest admissible mechanism -- but never at the cost of the
rules above.

NON-ADMISSIBLE EXAMPLES (form-level; A and B stand for roles from the
context list)
- "The forecaster shifts A by a fixed offset and caps its influence at
  a ceiling." [narrates algebra; no economic why]
- "The forecaster responds to a signal about the firm." [role-free
  placeholder -- the signal's role is never named]
- "The forecaster updates on A, and separately discounts when B is
  high, balancing the two." [two independent channels chained; split
  into two mechanisms]

OUTPUT FORMAT
Return ONLY a JSON object for the expression:
{"mechanisms": [
   {"theory_id": "<short_snake_case_slug>",
    "ladder": ["<richest description>", "<simpler>", "..."]},
   ...
]}
Use an empty "mechanisms" list only per the FALLBACK rule. No prose
outside the JSON object.
```

Notes. The prompts as sent in the July searches that produced the theories; text verbatim except that em-dashes, minus signs, and the element-of glyph are transliterated to ASCII for typesetting. Each core is assembled with one domain-context block (shown in the contexts figure) at call time.

Figure A5: The prompt implementing the proposer, which returns $\hat{D}_{\text{wide}}(f, C)$.

The naive LLM call: one call, no portability objective

```
INPUT

You are given a description of an empirical setting in which agents form forecasts, including the variables
      available in that setting, and a fitted forecasting rule: an algebraic expression in those variables,
      with numeric constants, that predicts the forecast described as Y_c.

OUTPUT

A theory of how forecasts are formed that would produce this rule in this setting. A theory has four parts:

0. A summary of the theory as a whole, in at most 30 words: the one mechanism by which forecasts are formed.
      It must state one core idea, something of which one can ask "does this apply here?", and it must
      capture what the formula does: a reader of the summary alone should expect the formula's shape.
1. A formula over abstract variables x1, x2, ... and named free parameters. Each abstract variable is either
      a single quantity (e.g. the most recent value of a leading indicator) or an indexed family of
      quantities (e.g. the daily outdoor temperatures over the past week, indexed by day, or the heights of
      the trees in a plot, indexed by tree). Use parameter names instead of the fitted numbers. You can use
      fixed numbers if they represent fixed structure (e.g. you might want a term like 1 + beta or 1 - beta).
      Use only these operations:
   - arithmetic: + - * / ^, and numbers as fixed structure; pdiv(a, b) = a / b, except 1 when |b| < 0.000001;
   - single-argument functions: abs, log, exp, sqrt, square, sign, where log(x) = the natural log of max(x,
      exp(-10));
   - indicator(condition), with conditions built from >, <, >=, <=, ==, !=;
   - gate(a, b) = 1 if a >= b else 0;
   - sgate(x, c, s) = 1 / (1 + exp(-(x - c) / |s|)), a smooth threshold around c with softness s (c and s are
      free parameters);
   - dead(x, c) = sign(x) * max(|x| - |c|, 0), no response inside a band of half-width |c| around zero (c is
      a free parameter);
   - over a family variable, whose members are in a fixed order (e.g. by age, most recent first, or by
      horizon, nearest first): mean_k, max_k, min_k, std_k (population standard deviation), range_k and
      trend_k over its first k members (trend_k = least-squares slope of the members against their position,
      running from the k-th member to the first, so for members ordered most recent first it is the slope
      over time; k is a free integer parameter), and decay(family, c) = the weighted sum of all members
      divided by the sum of the absolute weights, where the weight on the member at position p (p = 0 for the
      first member) is |c|^p if c >= 0, and 1 for p = 0 and -|c|^p for p >= 1 if c < 0 (c is a free
      parameter).
2. A description of each abstract variable. Start each description with a short mathematical notation for
      the variable, followed by a colon. Use braces {} in the notation only for the index of a family (e.g.
      T_{j}); a single variable's notation has no braces (e.g. N_t, F_(t-1)[Y_t]). For example, a single
      variable: "N_t: The number of consecutive periods, up to the most recent realization, in which the
      forecasted variable has moved in the same direction." A family: "T_{j}, j = 1..J: The outdoor
      temperature recorded on each of the J days before the forecast, indexed by how many days ago, most
      recent first."
    Each description must name one concept. Do not use "or" statements to join several concepts in one
      description, for example "the height of the tallest tree in the plot, or, if the plot has no trees, the
      height of the tallest shrub".
    Describe the concept precisely but in as few words as possible.
3. A description of the forecast the theory predicts (y*), in the same "notation: concept" form, e.g.
      "F_t[Y_(t+1)]: The agent's forecast, made at t, of next period's value of the forecasted variable."

OUTPUT FORMAT

Return only a JSON object:
{
  "summary": "<at most 30 words>",
  "y_description": "<notation>: <what it is>",
  "formula": "<expression in x1, x2, ... and parameter names>",
  "free_parameters": [{"name": "...", "constraint": "<e.g. between 0 and 1, positive, none>"}],
  "xstar": [{"symbol": "x1", "type": "single" | "family", "index": "<families only>", "description":
      "<notation>: <what it is>"}],
  "intended_binding": {"x1": "<expression in this setting's variables>"},
  "parameter_values": {"<name>": <number>}
}

CONSTRAINTS

- The theory must reproduce the fitted rule in this setting. Suppose a different researcher, who has never
      seen the fitted rule, reads only your theory and this setting's description, and binds each abstract
      variable: replaces it with the variable (or combination of variables) of this setting that matches its
      description. We call this step binding. For suitable values of the free parameters, the formula they
      end up with must equal the fitted rule you were given, up to algebraic rearrangement.
- How that researcher binds: they first check that your summary applies to this setting; then they bind each
      abstract variable on its own, seeing only its description (not the formula and not the other
      variables), to one variable of this setting or a simple combination of variables (such as a ratio, an
      indicator, or a unit conversion). A family is bound member by member.
- Every abstract variable and every free parameter must be needed to reproduce the rule.
- In general, lean towards fewer free parameters where possible.
```

Figure A6: The prompt of the naive LLM call: one call, with no portability objective.

The portability loop: the proposer prompt of every round

```
INPUT

You are given a description of an empirical setting in which agents form forecasts, including the variables
      available in that setting, and a fitted forecasting rule: an algebraic expression in those variables,
      with numeric constants, that predicts the forecast described as Y_c.

OUTPUT

A theory of how forecasts are formed that would produce this rule in this setting. A theory has four parts:

0. A summary of the theory as a whole, in at most 30 words: the one mechanism by which forecasts are formed.
      It must state one core idea, something of which one can ask "does this apply here?", and it must
      capture what the formula does: a reader of the summary alone should expect the formula's shape.
1. A formula over abstract variables x1, x2, ... and named free parameters. Each abstract variable is either
      a single quantity (e.g. the most recent value of a leading indicator) or an indexed family of
      quantities (e.g. the daily outdoor temperatures over the past week, indexed by day, or the heights of
      the trees in a plot, indexed by tree). Use parameter names instead of the fitted numbers. You can use
      fixed numbers if they represent fixed structure (e.g. you might want a term like 1 + beta or 1 - beta).
      Use only these operations:
   - arithmetic: + - * / ^, and numbers as fixed structure; pdiv(a, b) = a / b, except 1 when |b| < 0.000001;
   - single-argument functions: abs, log, exp, sqrt, square, sign, where log(x) = the natural log of max(x,
      exp(-10));
   - indicator(condition), with conditions built from >, <, >=, <=, ==, !=;
   - gate(a, b) = 1 if a >= b else 0;
   - sgate(x, c, s) = 1 / (1 + exp(-(x - c) / |s|)), a smooth threshold around c with softness s (c and s are
      free parameters);
   - dead(x, c) = sign(x) * max(|x| - |c|, 0), no response inside a band of half-width |c| around zero (c is
      a free parameter);
   - over a family variable, whose members are in a fixed order (e.g. by age, most recent first, or by
      horizon, nearest first): mean_k, max_k, min_k, std_k (population standard deviation), range_k and
      trend_k over its first k members (trend_k = least-squares slope of the members against their position,
      running from the k-th member to the first, so for members ordered most recent first it is the slope
      over time; k is a free integer parameter), and decay(family, c) = the weighted sum of all members
      divided by the sum of the absolute weights, where the weight on the member at position p (p = 0 for the
      first member) is |c|^p if c >= 0, and 1 for p = 0 and -|c|^p for p >= 1 if c < 0 (c is a free
      parameter).
2. A description of each abstract variable. Start each description with a short mathematical notation for
      the variable, followed by a colon. Use braces {} in the notation only for the index of a family (e.g.
      T_{j}); a single variable's notation has no braces (e.g. N_t, F_(t-1)[Y_t]). For example, a single
      variable: "N_t: The number of consecutive periods, up to the most recent realization, in which the
      forecasted variable has moved in the same direction." A family: "T_{j}, j = 1..J: The outdoor
      temperature recorded on each of the J days before the forecast, indexed by how many days ago, most
      recent first."
    Each description must name one concept. Do not use "or" statements to join several concepts in one
      description, for example "the height of the tallest tree in the plot, or, if the plot has no trees, the
      height of the tallest shrub".
    Describe the concept precisely but in as few words as possible.
3. A description of the forecast the theory predicts (y*), in the same "notation: concept" form, e.g.
      "F_t[Y_(t+1)]: The agent's forecast, made at t, of next period's value of the forecasted variable."

OUTPUT FORMAT

Return only a JSON object:
{
  "summary": "<at most 30 words>",
  "y_description": "<notation>: <what it is>",
  "formula": "<expression in x1, x2, ... and parameter names>",
  "free_parameters": [{"name": "...", "constraint": "<e.g. between 0 and 1, positive, none>"}],
  "xstar": [{"symbol": "x1", "type": "single" | "family", "index": "<families only>", "description":
      "<notation>: <what it is>"}],
  "intended_binding": {"x1": "<expression in this setting's variables>"},
  "parameter_values": {"<name>": <number>}
}

CONSTRAINTS

- The theory must reproduce the fitted rule in this setting. Suppose a different researcher, who has never
      seen the fitted rule, reads only your theory and this setting's description, and binds each abstract
      variable: replaces it with the variable (or combination of variables) of this setting that matches its
      description. We call this step binding. For suitable values of the free parameters, the formula they
      end up with must equal the fitted rule you were given, up to algebraic rearrangement.
- How that researcher binds: they first check that your summary applies to this setting; then they bind each
      abstract variable on its own, seeing only its description (not the formula and not the other
      variables), to one variable of this setting or a simple combination of variables (such as a ratio, an
      indicator, or a unit conversion). A family is bound member by member.
- Every abstract variable and every free parameter must be needed to reproduce the rule.
- In general, lean towards fewer free parameters where possible.

OBJECTIVE

The goal is to produce a theory of the maximum breadth: the number of other forecasting settings, unknown to
      you and varying widely in subject, data and timing, in which its summary applies and every one of its
      abstract variables can be bound to a variable of that setting. An abstract variable should therefore
      not be a narration of a specific variable in this setting; it should be a concept meaningful in as many
      forecasting or expectations settings as possible.
```

From round 2 on, appended: the header of the previous-attempts block, followed by each earlier proposal with its graded portability

```
PREVIOUS ATTEMPTS

You previously proposed the theories below for this same setting and fitted rule. Each is shown with its breadth: the number of settings, out of {n}, in which its summary applies and every one of its abstract variables can be
      bound (0 if the theory did not reproduce the fitted rule). Try to improve on them: propose a theory with greater breadth.
```

Notes. The prompt uses the term "breadth"; we later renamed the concept "portability". The two mean the same thing.

Figure A7: The prompt of the portability loop: the proposer prompt of every round, and the previous-attempts block added from round 2 on.

Panel A: the quarterly context

```
DOMAIN CONTEXT -- ANALYST QUARTERLY EPS FORECASTS
This context contains DEFINITIONS ONLY. Never add mechanism
examples, discovered-theory language, or example expressions to
this file.

TARGET
The adjustment term updates the analyst consensus EPS forecast for the
UPCOMING QUARTER. Target units: EPS (dollars per share).

FORECAST HORIZON AND PERIODS
The forecast horizon is ONE QUARTER. Monthly-walk families are observed
ON AVERAGE 3 times per horizon (roughly 3 periods per forecast
horizon; not always exactly 3); quarterly-walk
families step once per horizon; b-suffixes on plain columns give the
observation's age (b1qtr = one horizon old; b1mo = one month old, i.e.
one third of a horizon).

HORIZON vs AGE -- read every name as (target horizon, information
age): q1/q2/q3/y1/y2 in a name = the FORECAST HORIZON (which target
the number is about); the b-suffix = the INFORMATION AGE (how old
the observation is). Worked example: Fq3_b3qtr is the consensus formed three quarters ago for the
target that was then three quarters ahead -- that target is now the
quarter just being realized, so this is a stale far-horizon forecast
of the MOST RECENTLY REALIZED earnings, not of the current target.

DEFAULT UNIT SCALER
abs(Fq2_b1qtr) -- the magnitude of the LAGGED FORECAST of the current
target: the consensus for the current quarterly target as it stood one
quarter ago (rolling h=2 observed one quarter back lands on the
current target). Use exactly this scaler for unit restoration unless
the mechanism itself dictates a different specific forecast-level term.
Never estimate or vary its lag; never use an aggregator as the scaler.

ECONOMIC ROLES (the only admissible variable roles)
- prior consensus forecast (levels: Fq*, F_aq_*, and their walks)
- realized earnings (EPS_quarter*, EPS_annual*, and their walks)
- machine/statistical expectation of earnings (yhat_q_* columns, YHAT_Q)
- own past forecast error (fe_q*, fe_y*, and their walks)
- past forecast revision (forecast_revision_*, drev_*, and their walks)
- forecaster disagreement (stdev_*, range_*, meanmed_*, DISP families)
- analyst coverage / up-down revision counts (numest/numup/numdown)
- stock price (prc_t*)
- stock return (ret_t*)
- known behavioral-theory revision terms (theory_* columns; see vocabulary)
Note: the vocabulary deliberately includes ANNUAL-target families
(walks of annual consensus levels, revisions, and dispersion).
Cross-horizon information is legitimate: mechanisms may read
far-horizon signals when updating the quarterly forecast.

TWO INDEXING CONVENTIONS (read this first)
Two anchor clocks exist. The QUARTERLY anchor (the _aq_ system) is the
quarterly consensus cycle -- about ONE QUARTER back. Horizon labels
count fiscal period-ends FORWARD FROM THE ANCHOR: q1 = the first
quarterly period after the anchor (the MOST RECENTLY REALIZED
quarter), q2 = the second (the CURRENT target, realized soon), q3 =
the third; y1/y2 = the first and second annual fiscal periods ending
after the anchor. All observed on the quarterly-anchor cadence. The ANNUAL anchor -- about ONE YEAR back -- pins the annual
(_ay_) system; that system is excluded from this quarterly context
(FEATSET_Q).

1. FIXED-TARGET columns (names containing _aq_ or _ay_; monthly): the
   horizon tag hz names one FIXED fiscal period relative to the anchor,
   and b<M>mo moves ONLY the observation month -- b0mo IS the anchor
   month, b1mo one month before it. As N grows you see the consensus
   (or its revision, dispersion, counts) for the SAME fiscal period as
   it stood N months earlier.

2. ROLLING-HORIZON columns (no _aq_/_ay_; horizon-frequency): both the
   observation time AND the referenced target roll. b<N> counts whole
   horizons back from the anchor date and also shifts the referenced
   target: Fq<h>_b<N>qtr refers to the target that was h ahead AT THAT
   TIME, so its horizon relative to NOW is h-N (Fq3_b3qtr refers to the
   quarter now being realized). The same rolling logic applies to
   forecast_revision, fe, and realized-EPS lags.

[VOCABULARY REGENERATED FROM THE FEATSET RULE]
COLUMN VOCABULARY (complete inventory via placeholders)
The admissible set below EQUALS the black-box FEATSET_Q allow-list plus the machine/theory terms. Every
combination of placeholder values in range EXISTS and is a valid
plain terminal; nothing else does. Unpinned age N: use the most
recent element; the fitter may re-estimate the age within range.

Consensus forecast levels -- ROLLING-HORIZON (horizon-frequency):
- Fq<h>_b<N>qtr, h  in  {1,2,3}, N  in  1..12 -- consensus EPS forecast for the target <h> quarters ahead, observed N
      quarters ago
- Fy<h>_b<N>yr, h  in  {1,2}, N  in  1..8 -- annual consensus EPS forecast <h> years ahead, observed N years ago

Consensus forecast levels -- FIXED-TARGET (monthly):
- F_aq_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..12 -- monthly consensus EPS forecast level (quarterly-consensus
      set), FIXED horizon hz, observed N months before the anchor

Past forecast revisions:
- drev_aq_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..11 -- monthly revision of the quarterly-consensus-set forecast
      (FIXED-TARGET), N months before the anchor
- forecast_revision_q_b<N>qtr, N  in  1..12 -- quarterly consensus revision, N quarters old (ROLLING)
- forecast_revision_y_b<N>yr, N  in  1..8 -- annual consensus revision, N years old (ROLLING)

Forecaster disagreement -- FIXED-TARGET (monthly):
- stdev_aq_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..12 -- dispersion (std) of analyst forecasts
      (quarterly-consensus set)
- range_aq_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..12 -- high-minus-low range of analyst forecasts
      (quarterly-consensus set)
- meanmed_aq_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..12 -- mean-minus-median gap of analyst forecasts
      (quarterly-consensus set)

Analyst coverage / up-down counts -- FIXED-TARGET (monthly):
- numest_aq_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..12 -- number of analysts forecasting (quarterly-consensus set)
- numup_aq_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..12 -- analysts revising UP in the month (quarterly-consensus
      set)
- numdown_aq_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..12 -- analysts revising DOWN in the month
      (quarterly-consensus set)

Realized earnings, own errors, market:
- EPS_quarter_b<N>qtr, N  in  0..12 -- realized quarterly EPS, N quarters old (ROLLING)
- EPS_quarter_b<N>yr, N  in  1..8 -- realized quarterly EPS of the SAME fiscal quarter N YEARS earlier (year-over-year walk)
- EPS_annual_b<N>yr, N  in  0..8 -- realized annual EPS, N years old
- fe_q_b<N>qtr, N  in  1..12 -- own past quarterly forecast error, N quarters old (ROLLING)
- fe_y_b<N>yr, N  in  1..8 -- own past annual forecast error, N years old (ROLLING)
- ret_t_b<N>mo, N  in  1..12 -- monthly stock return, N months old
- prc_t_b<N>mo, N  in  1..12 -- stock price, N months old

Machine expectations and theory terms:
- yhat_q_h<j>_b<N>qtr, j  in  1..3, N  in  0..12 -- machine
  (forecast-blind) expectation of the quarterly target at the j-th
  nearest horizon (h-index = staleness). N = 0 is the CURRENT value
  (the columns named yhat_q_h<j>_b0qtr); N >= 1 are ROLLING row lags
  (columns yhat_q_h<j>_b<N>qtr).
- theory_<name>_q_b<N>qtr, name  in  {adaptive, fire, sticky,
  diagnostic, constant_gain}, N  in  0..12 -- the known behavioral-theory
  revision terms; N = 0 is the current value (columns
  theory_<name>_q_b0qtr), N >= 1 are ROLLING row lags.
  Glosses: adaptive = revises by a fraction of the own past forecast
  error; fire = full-information benchmark correction (the gap to the
  rational forecast); sticky = partial adjustment anchored to the
  prior consensus; diagnostic = overreaction to the latest news;
  constant_gain = fixed-gain learning from the own past error.

FAMILY VOCABULARY -- the complete list of aggregator families.
Aggregators read the k MOST RECENT elements of a family (the first k
in listed order). k is a wildcard (k1, k2, ...) fitted downstream,
2 <= k <= the family's size. agg_mean/agg_max/agg_min reduce those k
elements; agg_trend fits a per-period slope across them.

Consensus-level walks (FIXED-TARGET monthly):
- WALK__F_aq_q1: [F_aq_q1_b0mo, F_aq_q1_b1mo, ...] (13 elements)
- WALK__F_aq_q2: [F_aq_q2_b0mo, F_aq_q2_b1mo, ...] (13 elements)
- WALK__F_aq_q3: [F_aq_q3_b0mo, F_aq_q3_b1mo, ...] (13 elements)
- WALK__F_aq_y1: [F_aq_y1_b0mo, F_aq_y1_b1mo, ...] (13 elements)
- WALK__F_aq_y2: [F_aq_y2_b0mo, F_aq_y2_b1mo, ...] (13 elements)

Consensus-level walks (ROLLING-HORIZON):
- WALK__Fq1: [Fq1_b1qtr, Fq1_b2qtr, ...] (12 elements)
- WALK__Fq2: [Fq2_b1qtr, Fq2_b2qtr, ...] (12 elements)
- WALK__Fq3: [Fq3_b1qtr, Fq3_b2qtr, ...] (12 elements)
- WALK__Fy1: [Fy1_b1yr, Fy1_b2yr, ...] (8 elements)
- WALK__Fy2: [Fy2_b1yr, Fy2_b2yr, ...] (8 elements)

Revision walks (FIXED-TARGET monthly):
- WALK__drev_aq_q1: [drev_aq_q1_b0mo, drev_aq_q1_b1mo, ...] (12 elements)
- WALK__drev_aq_q2: [drev_aq_q2_b0mo, drev_aq_q2_b1mo, ...] (12 elements)
- WALK__drev_aq_q3: [drev_aq_q3_b0mo, drev_aq_q3_b1mo, ...] (12 elements)
- WALK__drev_aq_y1: [drev_aq_y1_b0mo, drev_aq_y1_b1mo, ...] (12 elements)
- WALK__drev_aq_y2: [drev_aq_y2_b0mo, drev_aq_y2_b1mo, ...] (12 elements)

Revision walks (ROLLING-HORIZON):
- WALK__forecast_revision_q: [forecast_revision_q_b1qtr, forecast_revision_q_b2qtr, ...] (12 elements)
- WALK__forecast_revision_y: [forecast_revision_y_b1yr, forecast_revision_y_b2yr, ...] (8 elements)

Disagreement walks (FIXED-TARGET monthly):
- WALK__meanmed_aq_q1: [meanmed_aq_q1_b0mo, meanmed_aq_q1_b1mo, ...] (13 elements)
- WALK__meanmed_aq_q2: [meanmed_aq_q2_b0mo, meanmed_aq_q2_b1mo, ...] (13 elements)
- WALK__meanmed_aq_q3: [meanmed_aq_q3_b0mo, meanmed_aq_q3_b1mo, ...] (13 elements)
- WALK__meanmed_aq_y1: [meanmed_aq_y1_b0mo, meanmed_aq_y1_b1mo, ...] (13 elements)
- WALK__meanmed_aq_y2: [meanmed_aq_y2_b0mo, meanmed_aq_y2_b1mo, ...] (13 elements)
- WALK__range_aq_q1: [range_aq_q1_b0mo, range_aq_q1_b1mo, ...] (13 elements)
- WALK__range_aq_q2: [range_aq_q2_b0mo, range_aq_q2_b1mo, ...] (13 elements)
- WALK__range_aq_q3: [range_aq_q3_b0mo, range_aq_q3_b1mo, ...] (13 elements)
- WALK__range_aq_y1: [range_aq_y1_b0mo, range_aq_y1_b1mo, ...] (13 elements)
- WALK__range_aq_y2: [range_aq_y2_b0mo, range_aq_y2_b1mo, ...] (13 elements)
- WALK__stdev_aq_q1: [stdev_aq_q1_b0mo, stdev_aq_q1_b1mo, ...] (13 elements)
- WALK__stdev_aq_q2: [stdev_aq_q2_b0mo, stdev_aq_q2_b1mo, ...] (13 elements)
- WALK__stdev_aq_q3: [stdev_aq_q3_b0mo, stdev_aq_q3_b1mo, ...] (13 elements)
- WALK__stdev_aq_y1: [stdev_aq_y1_b0mo, stdev_aq_y1_b1mo, ...] (13 elements)
- WALK__stdev_aq_y2: [stdev_aq_y2_b0mo, stdev_aq_y2_b1mo, ...] (13 elements)

Count walks (FIXED-TARGET monthly):
- WALK__numdown_aq_q1: [numdown_aq_q1_b0mo, numdown_aq_q1_b1mo, ...] (13 elements)
- WALK__numdown_aq_q2: [numdown_aq_q2_b0mo, numdown_aq_q2_b1mo, ...] (13 elements)
- WALK__numdown_aq_q3: [numdown_aq_q3_b0mo, numdown_aq_q3_b1mo, ...] (13 elements)
- WALK__numdown_aq_y1: [numdown_aq_y1_b0mo, numdown_aq_y1_b1mo, ...] (13 elements)
- WALK__numdown_aq_y2: [numdown_aq_y2_b0mo, numdown_aq_y2_b1mo, ...] (13 elements)
- WALK__numest_aq_q1: [numest_aq_q1_b0mo, numest_aq_q1_b1mo, ...] (13 elements)
- WALK__numest_aq_q2: [numest_aq_q2_b0mo, numest_aq_q2_b1mo, ...] (13 elements)
- WALK__numest_aq_q3: [numest_aq_q3_b0mo, numest_aq_q3_b1mo, ...] (13 elements)
- WALK__numest_aq_y1: [numest_aq_y1_b0mo, numest_aq_y1_b1mo, ...] (13 elements)
- WALK__numest_aq_y2: [numest_aq_y2_b0mo, numest_aq_y2_b1mo, ...] (13 elements)
- WALK__numup_aq_q1: [numup_aq_q1_b0mo, numup_aq_q1_b1mo, ...] (13 elements)
- WALK__numup_aq_q2: [numup_aq_q2_b0mo, numup_aq_q2_b1mo, ...] (13 elements)
- WALK__numup_aq_q3: [numup_aq_q3_b0mo, numup_aq_q3_b1mo, ...] (13 elements)
- WALK__numup_aq_y1: [numup_aq_y1_b0mo, numup_aq_y1_b1mo, ...] (13 elements)
- WALK__numup_aq_y2: [numup_aq_y2_b0mo, numup_aq_y2_b1mo, ...] (13 elements)

Realized-earnings and own-error walks:
- WALK__EPS_annual: [EPS_annual_b0yr, EPS_annual_b1yr, ...] (9 elements)
- WALK__EPS_quarter: [EPS_quarter_b0qtr, EPS_quarter_b1qtr, ...] (13 elements)
- WALK__EPS_quarter_yr: [EPS_quarter_b1yr, EPS_quarter_b2yr, ...] (8 elements)
- WALK__fe_q: [fe_q_b1qtr, fe_q_b2qtr, ...] (12 elements)
- WALK__fe_y: [fe_y_b1yr, fe_y_b2yr, ...] (8 elements)

Market walks:
- WALK__prc_t: [prc_t_b1mo, prc_t_b2mo, ...] (12 elements)
- WALK__ret_t: [ret_t_b1mo, ret_t_b2mo, ...] (12 elements)

Horizon ladders (SPLIT BY UNIT; aq anchor only):
- LADDER__aq__b<L>__q, L  in  0..12: [F_aq_q1_b<L>mo, F_aq_q2_b<L>mo, F_aq_q3_b<L>mo] (3 elements, nearest first)
- LADDER__aq__b<L>__y, L  in  0..12: [F_aq_y1_b<L>mo, F_aq_y2_b<L>mo] (2 elements)

Machine/theory walks and hat ladder:
- WALK__yhat_q_h<j>, j  in  1..3: [yhat_q_h<j>_b0qtr, yhat_q_h<j>_b1qtr, ...] (13 elements incl. current)
- WALK__theory_<name>_q: [theory_<name>_q_b0qtr, theory_<name>_q_b1qtr, ...] (13 elements incl. current)
- YHAT_Q: [yhat_q_h1_b0qtr, yhat_q_h2_b0qtr, yhat_q_h3_b0qtr] (3 elements, nearest horizon first)
```

Figure A8: The description of the quarterly context.

Panel B: the annual context

```
DOMAIN CONTEXT -- ANALYST ANNUAL EPS FORECASTS
This context contains DEFINITIONS ONLY. Never add mechanism
examples, discovered-theory language, or example expressions to
this file.

TARGET
The adjustment term updates the analyst consensus EPS forecast for the
UPCOMING YEAR. Target units: EPS (dollars per share).

FORECAST HORIZON AND PERIODS
The forecast horizon is ONE YEAR. Monthly-walk families are observed ON
AVERAGE 12 times per horizon; quarterly-walk families on average 4
times per horizon;
annual-frequency columns step once per horizon; b-suffixes give the
observation's age (b1yr = one horizon old; b1mo = one twelfth of a
horizon).

HORIZON vs AGE -- read every name as (target horizon, information
age): q1/q2/q3/y1/y2 in a name = the FORECAST HORIZON (which target
the number is about); the b-suffix = the INFORMATION AGE (how old
the observation is). Worked example: Fy2_b1yr is the consensus formed one year ago for the target
that was then two years ahead -- now roughly the current annual
target, seen through a stale far-horizon lens.

DEFAULT UNIT SCALER
abe(Fy2_b1yr) -- the magnitude of the LAGGED FORECAST of the current
target: the consensus for the current fiscal-year target as it stood
one year ago. Use exactly this scaler for unit restoration unless
the mechanism itself dictates a different specific forecast-level term.
Never estimate or vary its lag; never use an aggregator as the scaler.

ECONOMIC ROLES (the only admissible variable roles)
- prior consensus forecast (levels: Fy*, Fq*, F_ay_*, F_aq_*, walks)
- realized earnings (EPS_annual*, EPS_quarter*, and their walks)
- machine/statistical expectation of earnings (yhat_* families)
- own past forecast error (fe_y*, fe_q*, and their walks)
- past forecast revision (forecast_revision_y_*, drev_ay_*; the
  quarterly-anchored revision system is excluded by the ANNUAL
  INFORMATION RULE)
- forecaster disagreement (stdev_ay_*, range_ay_*, meanned_ay_* --
  the annual-anchored system; quarterly-anchored disagreement is
  excluded by the ANNUAL INFORMATION RULE)
- analyst coverage / up-down counts (numest/numup/numdown _ay_* --
  the annual-anchored system; quarterly-anchored counts are
  excluded by the ANNUAL INFORMATION RULE)
- stock price (prc_t*)
- stock return (ret_t*)
- known behavioral-theory revision terms (theory_* columns; see vocabulary)

ANNUAL INFORMATION RULE (leakage guard -- read before the grids)
The annual target's outcome accrues between the ANNUAL ANCHOR (one year
back) and the present. Any FORECAST-CONTAINING observation made inside
that window -- consensus levels, consensus revisions, forecaster
disagreement, analyst counts, own errors whose forecast was formed
inside the window -- partially embodies the outcome itself and is
LEAKAGE. Therefore ROLLING forecast-containing columns are admissible
only at ages of at least one year, and the quarterly-anchored monthly
system (_aq_ levels, drev, dispersion, counts and its ladders) plus ALL
quarterly consensus forecast columns are EXCLUDED from this context
entirely. The annual-anchored monthly (_ay_) system IS admissible: it
is pinned to the ANNUAL anchor and observed b<N>mo at or before that
anchor, so every _ay_ observation predates the outcome window. REALIZED
quantities (EPS), market data (returns, prices), and forecast-blind
machine expectations are admissible at any age.

TWO INDEXING CONVENTIONS (read this first)
The ANNUAL anchor sits about ONE YEAR back -- the start of the annual
outcome window. The _ay_ fixed-target system is pinned to it, with horizon
labels counting fiscal year-ends FORWARD FROM THE ANCHOR: y1 = the
first fiscal year after the anchor (the MOST RECENTLY REALIZED fiscal
year), y2 = the second (the CURRENT fiscal-year target). Observed
monthly back from the anchor. Quarterly-anchored systems (_aq_ and the quarterly forecast
columns) are excluded from this context by the FEATSET_Y rule.

1. FIXED-TARGET columns (names containing _aq_ or _ay_; monthly): the
   horizon tag hz names one FIXED fiscal period relative to the anchor,
   and b<N>mo moves ONLY the observation month -- b0mo IS the anchor
   month, b1mo one month before it. As N grows you see the consensus
   (or its revision, dispersion, counts) for the SAME fiscal period as
   it stood N months earlier.

2. ROLLING-HORIZON columns (no _aq_/_ay_; horizon-frequency): both the
   observation time AND the referenced target roll. b<N> counts whole
   horizons back from the anchor date and also shifts the referenced
   target: Fy<h>_b<N>yr refers to the target that was h ahead AT THAT
   TIME, so its horizon relative to NOW is h-N (Fy2_b1yr refers to the
   current fiscal-year target seen one year ago). The same rolling
   logic applies to forecast_revision_y, fe_y, and realized-EPS lags.

[VOCABULARY REGENERATED FROM THE FEATSET RULE]
COLUMN VOCABULARY (complete inventory via placeholders)
The admissible set below EQUALS the black-box FEATSET_Y allow-list plus the machine/theory terms. Every
combination of placeholder values in range EXISTS and is a valid
plain terminal; nothing else does. Unpinned age N: use the most
recent element; the fitter may re-estimate the age within range.

Consensus forecast levels -- ROLLING-HORIZON (horizon-frequency):
- Fy<h>_b<N>yr, h  in  {1,2}, N  in  1..8 -- annual consensus EPS forecast <h> years ahead, observed N years ago

Consensus forecast levels -- FIXED-TARGET (monthly):
- F_ay_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..12 -- monthly consensus EPS forecast level (annual-consensus set),
      FIXED horizon hz, observed N months before the anchor

Past forecast revisions:
- drev_ay_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..11 -- monthly revision of the annual-consensus-set forecast
      (FIXED-TARGET), N months before the anchor
- forecast_revision_y_b<N>yr, N  in  1..8 -- annual consensus revision, N years old (ROLLING)

Forecaster disagreement -- FIXED-TARGET (monthly):
- stdev_ay_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..12 -- dispersion (std) of analyst forecasts (annual-consensus
      set)
- range_ay_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..12 -- high-minus-low range of analyst forecasts
      (annual-consensus set)
- meanned_ay_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..12 -- mean-minus-median gap of analyst forecasts
      (annual-consensus set)

Analyst coverage / up-down counts -- FIXED-TARGET (monthly):
- numest_ay_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..12 -- number of analysts forecasting (annual-consensus set)
- numup_ay_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..12 -- analysts revising UP in the month (annual-consensus set)
- numdown_ay_<hz>_b<N>mo, hz  in  {q1,q2,q3,y1,y2}, N  in  0..12 -- analysts revising DOWN in the month (annual-consensus
      set)

Realized earnings, own errors, market:
- EPS_quarter_b<N>qtr, N  in  0..12 -- realized quarterly EPS, N quarters old (ROLLING)
- EPS_quarter_b<N>yr, N  in  1..8 -- realized quarterly EPS of the SAME fiscal quarter N YEARS earlier (year-over-year walk)
- EPS_annual_b<N>yr, N  in  0..8 -- realized annual EPS, N years old
- fe_y_b<N>yr, N  in  1..8 -- own past annual forecast error, N years old (ROLLING)
- ret_t_b<N>mo, N  in  1..12 -- monthly stock return, N months old
- prc_t_b<N>mo, N  in  1..12 -- stock price, N months old

Machine expectations and theory terms:
- yhat_y_b<j>_b<N>yr, j  in  1..2, N  in  0..8 -- ANNUAL machine
  (forecast-blind) expectation (h1 = current fiscal year, h2 = next);
  N = 0 is the current value (columns yhat_y_b<j>_b0yr), N >= 1 are
  ROLLING row lags.
- theory_<name>_y_b<N>yr, name  in  {adaptive, fire, sticky,
  diagnostic, constant_gain}, N  in  0..8 -- the known behavioral-theory
  revision terms; N = 0 is the current value (columns
  theory_<name>_y_b0yr), N >= 1 are ROLLING row lags.
  Glosses: adaptive = revises by a fraction of the own past forecast
  error; fire = full-information benchmark correction; sticky =
  partial adjustment anchored to the prior consensus; diagnostic =
  overreaction to the latest news; constant_gain = fixed-gain learning
  from the own past error.
(Quarterly machine hats yhat_q_* are FEATSET_Q-only and EXCLUDED here.)

FAMILY VOCABULARY -- the complete list of aggregator families.
Aggregators read the k MOST RECENT elements of a family (the first k
in listed order). k is a wildcard (k1, k2, ...) fitted downstream,
2 <= k <= the family's size. agg_mean/agg_max/agg_min reduce those k
elements; agg_trend fits a per-period slope across them.

Consensus-level walks (FIXED-TARGET monthly):
- WALK__F_ay_q1: [F_ay_q1_b0mo, F_ay_q1_b1mo, ...] (13 elements)
- WALK__F_ay_q2: [F_ay_q2_b0mo, F_ay_q2_b1mo, ...] (13 elements)
- WALK__F_ay_q3: [F_ay_q3_b0mo, F_ay_q3_b1mo, ...] (13 elements)
- WALK__F_ay_y1: [F_ay_y1_b0mo, F_ay_y1_b1mo, ...] (13 elements)
- WALK__F_ay_y2: [F_ay_y2_b0mo, F_ay_y2_b1mo, ...] (13 elements)

Consensus-level walks (ROLLING-HORIZON):
- WALK__Fy1: [Fy1_b1yr, Fy1_b2yr, ...] (8 elements)
- WALK__Fy2: [Fy2_b1yr, Fy2_b2yr, ...] (8 elements)

Revision walks (FIXED-TARGET monthly):
- WALK__drev_ay_q1: [drev_ay_q1_b0mo, drev_ay_q1_b1mo, ...] (12 elements)
- WALK__drev_ay_q2: [drev_ay_q2_b0mo, drev_ay_q2_b1mo, ...] (12 elements)
- WALK__drev_ay_q3: [drev_ay_q3_b0mo, drev_ay_q3_b1mo, ...] (12 elements)
- WALK__drev_ay_y1: [drev_ay_y1_b0mo, drev_ay_y1_b1mo, ...] (12 elements)
- WALK__drev_ay_y2: [drev_ay_y2_b0mo, drev_ay_y2_b1mo, ...] (12 elements)

Revision walks (ROLLING-HORIZON):
- WALK__forecast_revision_y: [forecast_revision_y_b1yr, forecast_revision_y_b2yr, ...] (8 elements)

Disagreement walks (FIXED-TARGET monthly):
- WALK__meanned_ay_q1: [meanned_ay_q1_b0mo, meanned_ay_q1_b1mo, ...] (13 elements)
- WALK__meanned_ay_q2: [meanned_ay_q2_b0mo, meanned_ay_q2_b1mo, ...] (13 elements)
- WALK__meanned_ay_q3: [meanned_ay_q3_b0mo, meanned_ay_q3_b1mo, ...] (13 elements)
- WALK__meanned_ay_y1: [meanned_ay_y1_b0mo, meanned_ay_y1_b1mo, ...] (13 elements)
- WALK__meanned_ay_y2: [meanned_ay_y2_b0mo, meanned_ay_y2_b1mo, ...] (13 elements)
- WALK__range_ay_q1: [range_ay_q1_b0mo, range_ay_q1_b1mo, ...] (13 elements)
- WALK__range_ay_q2: [range_ay_q2_b0mo, range_ay_q2_b1mo, ...] (13 elements)
- WALK__range_ay_q3: [range_ay_q3_b0mo, range_ay_q3_b1mo, ...] (13 elements)
- WALK__range_ay_y1: [range_ay_y1_b0mo, range_ay_y1_b1mo, ...] (13 elements)
- WALK__range_ay_y2: [range_ay_y2_b0mo, range_ay_y2_b1mo, ...] (13 elements)
- WALK__stdev_ay_q1: [stdev_ay_q1_b0mo, stdev_ay_q1_b1mo, ...] (13 elements)
- WALK__stdev_ay_q2: [stdev_ay_q2_b0mo, stdev_ay_q2_b1mo, ...] (13 elements)
- WALK__stdev_ay_q3: [stdev_ay_q3_b0mo, stdev_ay_q3_b1mo, ...] (13 elements)
- WALK__stdev_ay_y1: [stdev_ay_y1_b0mo, stdev_ay_y1_b1mo, ...] (13 elements)
- WALK__stdev_ay_y2: [stdev_ay_y2_b0mo, stdev_ay_y2_b1mo, ...] (13 elements)

Count walks (FIXED-TARGET monthly):
- WALK__numdown_ay_q1: [numdown_ay_q1_b0mo, numdown_ay_q1_b1mo, ...] (13 elements)
- WALK__numdown_ay_q2: [numdown_ay_q2_b0mo, numdown_ay_q2_b1mo, ...] (13 elements)
- WALK__numdown_ay_q3: [numdown_ay_q3_b0mo, numdown_ay_q3_b1mo, ...] (13 elements)
- WALK__numdown_ay_y1: [numdown_ay_y1_b0mo, numdown_ay_y1_b1mo, ...] (13 elements)
- WALK__numdown_ay_y2: [numdown_ay_y2_b0mo, numdown_ay_y2_b1mo, ...] (13 elements)
- WALK__numest_ay_q1: [numest_ay_q1_b0mo, numest_ay_q1_b1mo, ...] (13 elements)
- WALK__numest_ay_q2: [numest_ay_q2_b0mo, numest_ay_q2_b1mo, ...] (13 elements)
- WALK__numest_ay_q3: [numest_ay_q3_b0mo, numest_ay_q3_b1mo, ...] (13 elements)
- WALK__numest_ay_y1: [numest_ay_y1_b0mo, numest_ay_y1_b1mo, ...] (13 elements)
- WALK__numest_ay_y2: [numest_ay_y2_b0mo, numest_ay_y2_b1mo, ...] (13 elements)
- WALK__numup_ay_q1: [numup_ay_q1_b0mo, numup_ay_q1_b1mo, ...] (13 elements)
- WALK__numup_ay_q2: [numup_ay_q2_b0mo, numup_ay_q2_b1mo, ...] (13 elements)
- WALK__numup_ay_q3: [numup_ay_q3_b0mo, numup_ay_q3_b1mo, ...] (13 elements)
- WALK__numup_ay_y1: [numup_ay_y1_b0mo, numup_ay_y1_b1mo, ...] (13 elements)
- WALK__numup_ay_y2: [numup_ay_y2_b0mo, numup_ay_y2_b1mo, ...] (13 elements)

Realized-earnings and own-error walks:
- WALK__EPS_annual: [EPS_annual_b0yr, EPS_annual_b1yr, ...] (9 elements)
- WALK__EPS_quarter: [EPS_quarter_b0qtr, EPS_quarter_b1qtr, ...] (13 elements)
- WALK__EPS_quarter__yr: [EPS_quarter_b1yr, EPS_quarter_b2yr, ...] (8 elements)
- WALK__fe_y: [fe_y_b1yr, fe_y_b2yr, ...] (8 elements)

Market walks:
- WALK__prc_t: [prc_t_b1mo, prc_t_b2mo, ...] (12 elements)
- WALK__ret_t: [ret_t_b1mo, ret_t_b2mo, ...] (12 elements)

Horizon ladders (SPLIT BY UNIT; ay anchor only):
- LADDER__ay__b<L>_q, L  in  0..12: [F_ay_q1_b<L>mo, F_ay_q2_b<L>mo, F_ay_q3_b<L>mo] (3 elements, nearest first)
- LADDER__ay__b<L>_y, L  in  0..12: [F_ay_y1_b<L>mo, F_ay_y2_b<L>mo] (2 elements)

Machine/theory walks:
- WALK__theory_<name>_y: [theory_<name>_y_b0yr, theory_<name>_y_b1yr, ...] (9 elements incl. current)
- WALK__yhat_y_b<j> (9 elements incl. current)
```

Notes. The domain-context blocks handed to the binder and proposer prompts; text verbatim except that em-dashes, minus signs, and the element-of glyph are transliterated to ASCII for typesetting, and internal edit markers and a file reference are omitted.

Figure A9: The description of the annual context.

E Expectations contexts

This appendix lists the ten expectations contexts in $\mathcal{C}$ that the portability loop of Section 4.6 scores portability on. Each is described in words, with its forecast target and variables, and portability is graded against these descriptions; below is a short description of each and a paper that discusses it.

1. **Analyst earnings forecasts, quarterly.** The mean consensus of the analysts covering a U.S. firm forecasts the firm's earnings per share for the next quarter (Bouchaud et al., 2019).

2. **Analyst earnings forecasts, annual.** The same consensus forecasts the firm's earnings per share for the current fiscal year (Bouchaud et al., 2019).

3. **AR(1) forecasting experiment.** In a large randomized experiment, participants forecast stable AR(1) processes whose data-generating process the experimenter controls (Afrouzi et al., 2023).

4. **Grocery prices and inflation expectations.** Members of U.S. households in the Kilts–Nielsen Consumer Panel forecast inflation over the next 12 months, and the prices they see while grocery shopping may shape these forecasts (D'Acunto et al., 2021).

5. **Inflation expectations from lived experience.** U.S. consumers in the Michigan Survey of Consumers, 1953–2007, report the inflation they expect over the next year (Malmendier and Nagel, 2016).

6. **New Zealand firms' inflation expectations.** General managers of New Zealand firms, surveyed from 2013Q4 to 2016Q4, forecast aggregate inflation (Coibion et al., 2018b).

7. **Daily lottery.** Before each daily draw of a three-digit number from 000 to 999, players bet on the numbers of their choice; the bets reveal what players expect to be drawn (Clotfelter and Cook, 1993).

8. **Election prediction market.** A real-money market trades winner-take-all contracts on each of two presidential candidates, and the price of a contract is the market's probability that its candidate wins (Wolfers and Zitzewitz, 2004).

9. **Movie opening weekends.** Forecasts of a film's domestic opening-weekend box office gross, made before release and updated as the weekend's first numbers come in (Court et al., 2018).

10. **Penalty-shootout goalkeeper.** A goalkeeper facing five kicks in a shootout commits to a side before each kick, so the side chosen reveals where the goalkeeper expects the kick to go (Palacios-Huerta, 2003).

F Results

F.1 Where else sticky (unless there's bad news) expectations applies

Table A12: Settings outside analyst earnings where sticky (unless there's bad news) expectations bind, and what the theory predicts there.

Setting and data	$F_{(t-1)}[Y_{(t+1)}]$, **previous forecast of the current target**	G_j, **objective model prediction minus the earlier forecast**	e_t, **most recent forecast error**	**Prediction**
College students' grades Berea Panel Study (Stinebrickner and Stinebrickner, 2012, 2014)	the student's forecast, at the end of the first year, of the GPA for the fall of the second year	a GPA prediction from high-school GPA, ACT score and college grades so far (administrative records), made at the start of the fall, minus the student's earlier forecast	the spring semester's GPA minus the student's forecast of it	after a semester with grades below their own forecast, students move further toward the record-based prediction
Farmers' yields NASS Agricultural Yield Survey, the same farmers every month, May–November (USDA National Agricultural Statistics Service, 2026); Egelkraut et al. (2006)	the farmer's expected yield per acre for this season, reported in last month's survey	the field-measured yield from NASS's Objective Yield Survey for the same crop and state, minus the farmer's earlier expected yield	last season's harvested yield minus the farmer's forecast of it	after a harvest below their own forecast, farmers move their expected yield further toward the field measurements
Macroeconomic forecasts Survey of Professional Forecasters microdata (Croushore, 1993); Bordalo et al. (2020)	the forecaster's forecast of a quarter's real GDP growth, made one quarter earlier	a statistical forecast of the same quarter from data available then, minus the forecaster's earlier forecast	the latest released GDP growth minus the forecaster's forecast of it	after GDP growth comes in below their own forecast, forecasters close more of the gap to the statistical forecast

Notes. Each column is one variable of D_T in Figure 15, with what it binds to in the setting. The time structure of farmers' yields differs from analyst earnings: farmers revise their forecast of this season's yield every month from May to November, but yield is realized once a year, so the most recent forecast error is last season's yield minus the farmer's forecast of it. The same time structure difference also applies to the college students' grades. The theory's words do not require the time structure of the earnings setting; they only say that a more negative most recent forecast error leads to stronger updating.

F.2 How the search found sticky (unless there's bad news) expectations

This appendix gives the full account of how the search found the theory of Section 5.2.

Our procedure from Section 4 has a few steps: after fitting the black box $\hat{m}$ in Section 2, we run symbolic regression with the bindability proxy $\hat{p}_{\text{bind}} \geq 0.50$ (Figure A3). The progress across the generations of symbolic regression is shown in Figure A10. Note that the starting R^2 is already fairly high because we encode the predictions of standard sticky expectations (the orange dotted line). The search appears to converge after roughly 200 generations.

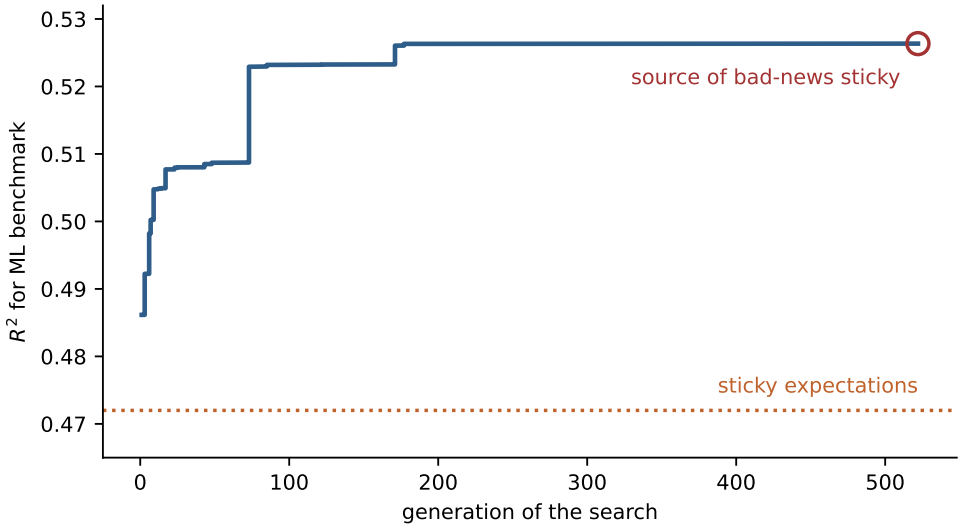
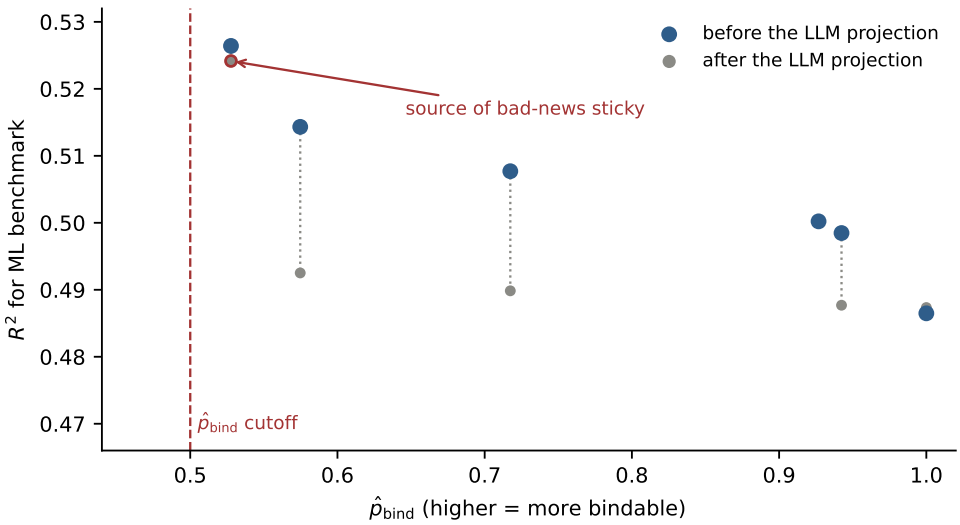


Figure A10: The search climbing generation by generation. Best R^2 against $\hat{m}$ found so far, in the final round of symbolic regression; dotted line: plain sticky expectations (R^2 0.472).

Next, the search takes the best functions from Figure A10 (in particular, those with the best trade-off between $\widehat{R^2}_{\hat{m}}$ and $\hat{p}_{\text{bind}}$) and runs an LLM "projection", where one LLM proposes a description D_T to explain some $\hat{f}$ and a second maps D_T into a bindable function (the after-search step of Figure A3). This helps correct for misspecification in $\hat{p}_{\text{bind}}$ that lets unbindable functions through (Figure A11). We note that in general, the lower $\hat{p}_{\text{bind}}$ is, the more R^2 is lost in the LLM projection (Figure A13 in Appendix F.4): the search often increases R^2 by letting unbindable components through. In Figure A11 something rare happens: the function with the highest $\widehat{R^2}_{\hat{m}}$ and the lowest $\hat{p}_{\text{bind}}$ roughly keeps its R^2 after the LLM projection (from 0.526 to 0.524), and it becomes sticky (unless there's bad news) expectations. This is because its pre-projection function is already almost sticky (unless there's bad news) expectations, with a soft gate in place of the hard one, while the other pre-projection functions combine operations such as dead zones that lose some of their R^2 once the math is cleaned up into a bindable function; Table A13 in Appendix F.4 shows each pre-projection function and what the LLM projection returns for it.[46] So if one worries that the LLM projection throws away the R^2 symbolic regression finds, at least in this round of search it does not. Finally, the search takes the function from Figure A11 and puts it through our LLM portability loop to maximize portability. We note that our LLM portability loop from Section 4.6 makes four generalizations from the specific implication for analyst forecasts: earnings become the target being forecast, a quarter becomes one period, the consensus forecast becomes the agents' forecast, and the machine-learning forecast $\hat{\mathbb{E}}$ becomes an objective (forecast-blind) model prediction.

[46] A dead zone sets small values to zero: $\text{dz}(x, c) = \text{sign}(x)\max(|x| - |c|, 0)$, which is zero when $|x| \leq |c|$ and moves one for one with x beyond that.



Notes. Output of the final round of symbolic regression. Blue: each function before the LLM projection (the search's best trade-offs of $\widehat{R^2}_{\hat{m}}$ against $\hat{p}_{\text{bind}}$), with $\widehat{R^2}_{\hat{m}}$ the out-of-fold R^2 against $\hat{m}$ on folds 1–4 (rotation). Grey: the LLM projection outputs many functions for each pre-projection function; the grey points are the best-predicting ones, with constants refit on folds 1–4 and $\widehat{R^2}_{\hat{m}}$ their in-sample R^2 there. Dotted lines join each pre-projection function to its LLM-projection output; the dashed line is the $\hat{p}_{\text{bind}} = 0.50$ cutoff; the ringed point is the function that became the bad-news theory.

Figure A11: The search's best functions before and after the LLM projection.

The theory as the algorithm returned it. Figure 15 rewrites the theory's description in plainer words. As the algorithm returned it, the summary reads "Sticky updating: agents move their previous forecast of the target partway toward fresh objective signals, using decaying weights on recent signal gaps, and update more strongly after a large disappointing surprise." and the variables are described as follows: $F_t[Y_{(t+1)}]$, "The agent's current forecast, made at t, of the current target period's value of the forecasted variable."; $F_{(t-1)}[Y_{(t+1)}]$, "The agent's previous forecast of the current target, made one period earlier."; G_j, "For the target of j periods ago, the objective (forecast-blind) model prediction of it made then, minus the agent's forecast of it made one period before that; most recent first."; and e_t, "The agent's most recent forecast error, the latest realized value minus the agent's forecast of it."

F.3 History-dependent sticky expectations

The first theory produced by our induction procedure is something we call *history-dependent sticky expectations*. We note that this is a special case of our sticky (unless there's bad news) expectations of Section 5.2, with $\delta = 0$. Figure A12 shows the theory $(\mathcal{F}_T, D_T)$, what each of its variables binds to in our context, and its fitted values, in the same format as Figure 15.

Theory $T = (\mathcal{F}_T, D_T)$

Summary. "Agents keep their previous forecast of the target and add a fixed fraction of an exponentially weighted history of gaps between an objective model estimate and their own earlier forecasts."

$$\mathcal{F}_T = \left\{ F_{(t-1)}[Y_{(t+1)}] + \beta \, \frac{G_0 + \gamma\, G_1 + \gamma^2\, G_2 + \cdots}{1 + \gamma + \gamma^2 + \cdots} \;:\; \beta \in [0,1],\ \gamma \in [0,1] \right\}$$

The theory's words (D_T)	**Bound in quarterly analyst EPS**
$F_t[Y_{(t+1)}]$ "The agent's forecast, made at t, of the forecasted variable's value for the current target period."	$F_t[Y_{i,t+1}]$, the consensus EPS forecast for next quarter
$F_{(t-1)}[Y_{(t+1)}]$ "The agent's forecast of the current target period's value, made one period earlier."	$F_{t-1}[Y_{i,t+1}]$, the consensus for the same quarter, made one quarter earlier
G_j "For the period that was the target j periods ago, the objective model-based estimate of its value made then, minus the agent's forecast of it made one period before that."	$g_{i,t-j}$ $(j = 0, \ldots, 12)$, the ML forecast minus the earlier consensus, j quarters back: $\hat{\mathbb{E}}_{t-j}\left[Y_{i,t-j+1}\right] - F_{t-j-1}[Y_{i,t-j+1}]$

Parameter	Role	Estimate on quarterly analyst EPS
β	Share of the information gap closed every period	0.391 [0.354, 0.429]
γ	Weight on older information gaps	0.239 [0.182, 0.301]

Figure A12: History-dependent sticky expectations: the theory, what its variables bind to in quarterly analyst EPS forecasts, and its fitted values. The estimates are fit directly on the forecast revision $\Delta F_{i,t}$, on fold 5, with 95% confidence intervals in brackets.

F.4 Fit lost in the LLM projection

This section pools the Pareto-frontier functions of all ten quarterly searches with $\hat{p}_{\text{bind}} \geq 0.5$ and shows, for each, how much of its R^2 survives the LLM projection of Section 4.5. In that step each frontier function goes through the proposer and binder projection (Figure A3): the proposer writes descriptions of the function, the binder writes a class of functions for each description, and the constants of each class are refit, giving one bindable function per description. We keep the best-fitting of these bindable functions and report its R^2 as a share of the frontier function's R^2. Table A13 shows the functions behind Figure A11: each frontier function of the final round and the best function the LLM projection returned for it. Functions with a lower $\hat{p}_{\text{bind}}$ keep a smaller share of their fit, so the best-fitting function keeping its fit, as in Figure A11, is rare.[47]

[47] All R^2 here are on folds 1–4, because all of this happens within the search process, before fold 5 is used: R^2 before the LLM projection is out-of-fold within folds 1–4 and R^2 after is the refit's in-sample R^2 on folds 1–4; from round 2 on, both are the R^2 added over the earlier rounds' theories.

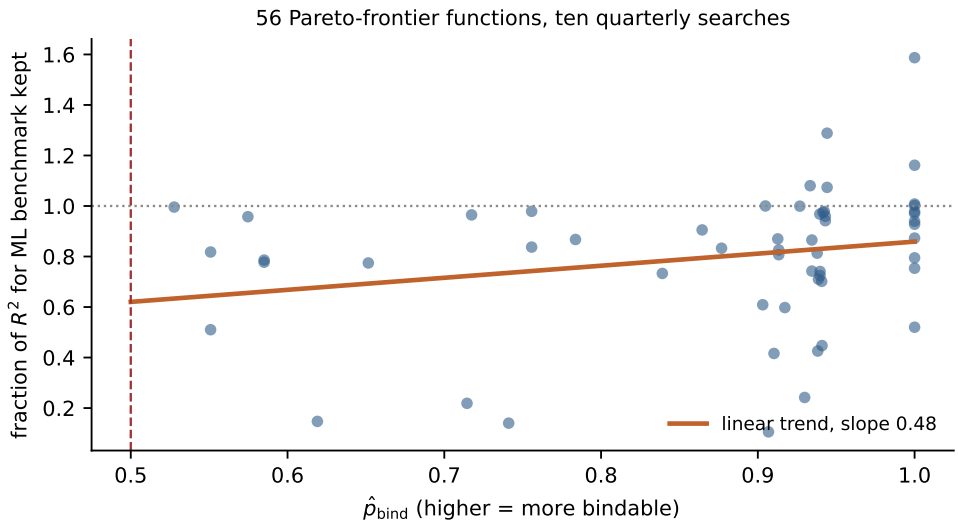


Figure A13: Fraction of fit kept after the LLM projection (R^2 after over R^2 before, best returned function) against $\hat{p}_{\text{bind}}$, Pareto-frontier functions from all ten rounds of symbolic regression on quarterly forecasts with $\hat{p}_{\text{bind}} \geq 0.5$, with a linear trend.

Table A13: The functions behind Figure A11: each Pareto-frontier function of the final round of symbolic regression and the best-fitting function the LLM projection returns for it.

$\hat{p}_{\text{bind}}$	R^2 **before** $\rightarrow$ **after**		**Pareto-frontier function, then after the LLM projection**
1.00	0.487	$\rightarrow$	*frontier:* $\text{dz}\big(\text{dz}(\text{decay}_{-0.21}(G),\, 0.0026),\, 0.0013\big)$
	0.487		*projected:* $0.46\,\text{dz}\big(\text{decay}_{0.21}(G),\, 0.0039\big)$
0.94	0.498	$\rightarrow$	*frontier:* $\text{decay}_{-0.19}(G)\,\big(-0.22 -$
	0.488		$0.34\,\sigma(\log \text{sign}(F_{t-1,-3}[Y_{t+2}]);\, -1.93, 4.71)\big)$
			projected: $0.46\,\text{decay}_{-0.21}(G)\,\mathbf{1}\{F_{t-1}[Y_{t+1}] \geq -1.68\}$
0.93	0.500	$\rightarrow$	*frontier:* $\text{decay}_{0.20}(G)\,\big(0.58 + 0.84\,\sigma(e^{A_{-2}};\, 1.13, 1.52)\big)$
	0.500		*projected:* $\text{decay}_{0.20}(G)\,\big(1.60 + 0.73\,\mathbf{1}\{A_{-2} \geq 0.5\}\big)$
0.72	0.508	$\rightarrow$	*frontier:* $\text{dz}\big(\text{decay}_{0.21}(G),\, 0.0053\big)\,/\,\big(4.93 + \text{sign}(\text{trend}_3(\hat{\mathbb{E}}))\big)$
	0.490		*projected:* $\text{dz}\big(\text{decay}_{-0.21}(G),\, 0.0041\big)\,\big(18.4 - \text{trend}_4(\hat{\mathbb{E}})/|F_{t-1}[Y_{t+1}]|\big)$
0.57	0.514	$\rightarrow$	*frontier:* $\text{dz}\big(\text{decay}_{-0.21}(G),\, 0.0030\big)\,/\,\big(e^{r_{t,2}} - 0.30\big)$
	0.493		*projected:* $\text{decay}_{-0.22}(G)\,/\,(1.36 + r_{t,1})$
0.53	0.526	$\rightarrow$	*frontier:* $\text{decay}_{0.23}(G)\,\big(-0.79 + 0.36\,\sigma(e_t;\, -0.110, 0.038)\big)$
	0.524		*projected:* $\text{decay}_{0.23}(G)\,\big(1.02 + 0.73\,\mathbf{1}\{-0.106 - e_t \geq 0.0006\}\big)$

Notes. G is the family of gaps G_j, $j = 0, 1, \ldots$ (the ML forecast minus the earlier consensus, j quarters back). $\text{decay}_c(G) = \sum_{j\geq 0} c^j G_j / \sum_{j\geq 0} c^j$, the decay-weighted average with weight c (a negative c gives alternating signs). The dead zone $\text{dz}(x, c) = \text{sign}(x)\max(|x| - |c|, 0)$ is zero when $|x| \leq |c|$ and moves one for one with x beyond that. The soft gate $\sigma(x;\, c, s) = 1/(1 + e^{-(x-c)/|s|})$ is a logistic step at c with softness s; $\mathbf{1}\{\cdot\}$ is the hard gate. log is protected, $\log(\max(x, e^{-10}))$, so $\log \text{sign}(x)$ is 0 when $x > 0$ and -10 otherwise. e_t is the latest forecast error; $F_{t-1,-3}[Y_{t+2}]$ is the consensus for the quarter after next, three months before the previous snapshot; A_{-2} is the annual consensus for the next fiscal year, two years earlier; $\text{trend}_k(\hat{\mathbb{E}})$ is the least-squares slope of the ML forecast over the last k quarters; $r_{t,\ell}$ is the stock's return ℓ months before the statement date. Constants rounded; the outer slope and intercept are

fit in the regression and not shown. R^2 for the ML benchmark: before is out-of-fold on the search sample, after is the refit's in-sample R^2, both on folds 1–4.

F.5 The ten induced theories

Round	What it captures	Fitted expression (constants symbolic)	R^2 vs. $\Delta F_{i,t}$ alone	R^2 vs. $\Delta F_{i,t}$ cum.	R^2 vs. $\hat{m}$ alone	R^2 vs. $\hat{m}$ cum.
1	history-dependent sticky expectations	$\beta\ \mathrm{decay}_{\gamma}\left(g_{i,t}\right)$	0.191	0.191	0.491	0.491
2	history-dependent rational	$\beta\ \mathrm{decay}_{\gamma}\left(\hat{\mathbb{E}}_t[Y_{i,t+1}] - \hat{\mathbb{E}}_{t-1}[Y_{i,t+1}]\right)$	0.118	—	0.303	—
2	rational-update volatility	$\beta\,\mathrm{std}\Big(\left\{\hat{\mathbb{E}}_{t-j}[Y_{i,t-j+1}] - \hat{\mathbb{E}}_{t-j-1}[Y_{i,t-j+1}]\right\}_{j=0}^{k}\Big)$	0.012	0.240	0.047	0.632
3	worst recent return	$\beta\Big(\min_{\ell\in\{1,\ldots,k\}} r_{i,t,\ell}\Big)\ \left\lvert F_{t-1}[Y_{i,t+1}]\right\rvert$	0.053	—	0.181	—
3	stale forecast level	$\beta\, F_{i,t-1,-3}\left[Y_{i,t}\right]$	0.016	0.253	0.051	0.685
4	saturating error correction	$\beta\left\lvert F_{t-1}[Y_{i,t+1}]\right\rvert\ \sigma\Big(\left(e_{i,t}/\left\lvert F_{t-1}[Y_{i,t+1}]\right\rvert - c_1\right)/c_2\Big)$	0.030	—	0.050	—
4	fixed-target revision momentum	$\beta\ \frac{1}{k}\sum_{j=0}^{k-1}\left(F_{i,t-1,-j}[Y_{i,t+1}] - F_{i,t-1,-(j+1)}[Y_{i,t+1}]\right)$	0.024	0.266	0.063	0.723
5	price trend	$\beta\, s_{i,t,1:4}\left\lvert F_{t-1}[Y_{i,t+1}]\right\rvert$	0.053	0.268	0.161	0.730
6	rolling-target revision momentum	$\beta\ \mathrm{decay}_{\gamma}\left(\Delta F_{i,t-1}\right)$	0.074	0.272	0.182	0.739
7	sticky (unless there's bad news) expectations	$\beta\left(1+\delta\,\mathbf{1}\{e_{i,t}\le-\kappa\}\right)\ \mathrm{decay}_{\gamma}\left(g_{i,t}\right)$	0.198	0.275	0.532	0.748

One row per function returned by function search, in the order the rounds found them; functions sharing a round number were found in the same round. 'What it captures' is a short description that we (as researchers) provide. 'alone' is the function fit by itself — internal constants and outer weights refit on folds 1–4 — scored on held-out fold 5; 'cum.' is the joint fit of all functions through that round, same folds, shown once per round on its last row. All R^2 on held-out fold 5, computed on the $N = 57{,}000$ fold-5 firm–quarters where $\hat{m}$ is defined, against the forecast-revision target and the machine benchmark $\hat{m}$ respectively. Notation: $g_{i,t} = \hat{\mathbb{E}}_t[Y_{i,t+1}] - F_{t-1}[Y_{i,t+1}]$, the gap, with $\hat{\mathbb{E}}_s[\,\cdot\,]$ the machine estimate of the rational forecast; $\mathrm{decay}_{\gamma}(x_{i,t}) = \sum_{k\ge0}\gamma^k x_{i,t-k} \big/ \sum_{k\ge0}\gamma^k$, the decay-weighted average (at $\gamma = 0$ it is $x_{i,t}$); $e_{i,t} = Y_{i,t} - F_{t-1}[Y_{i,t}]$, the latest past forecast error; $F_{i,t-1,-m}[Y_{i,t+1}]$ is the consensus for $Y_{i,t+1}$ held m months before the $t-1$ snapshot; $s_{i,t,1:4}$ is the least-squares slope of the price level over the four months before F_t, scaled by the last pre-F_t price; $r_{i,t,\ell}$ is the stock's monthly return ℓ months before the month-t statement date; integer constants such as k are fitted constants of the search; $\sigma(x) = e^x/(1+e^x)$, with $\sigma((z-c_1)/c_2)$ turning on around c_1 with softness c_2.

Table A14: The functions returned by function search.

Table A15: The ten induced theories as the portability loop returns them.

Round	D_T **(verbatim)**	$\mathcal{F}_T$ **(formula and free parameters)**
1 history-dependent sticky expectations	Summary: "Agents keep their previous forecast of the target and add a fixed fraction of an exponentially weighted history of gaps between an objective model estimate and their own earlier forecasts." y^*. $F_t[Y_{(t+1)}]$: "The agent's forecast, made at t, of the forecasted variable's value for the current target period." x_1. $F_{(t-1)}[Y_{(t+1)}]$: "The agent's forecast of the current target period's value, made one period earlier." x_2. G_j: "For the period that was the target j periods ago, the objective model-based estimate of its value made then, minus the agent's forecast of it made one period before that."	$x_1 + \text{gain} \cdot \text{decay}(x_2, \rho)$ gain: between 0 and 1; ρ: between 0 and 1
2 history-dependent rational	Summary: "Agents keep their previous forecast of the same target and add a fixed fraction of an exponentially weighted average of recent revisions in an objective, forecast-blind model prediction." y^*. $F_t[Y_{(t+1)}]$: "The agents' current forecast, made at t, of the target variable's value for the period now being forecast." x_1. $F_{(t-1)}[Y_{(t+1)}]$: "The agents' previous forecast of the same target value, made one period earlier." x_2. $R_j, j = 0, \ldots, J$: "For the target that was being forecast j periods ago, an objective forecast-blind model's prediction of it made then, minus the model's prediction of it made one period earlier. Ordered most recent first."	$x_1 + \kappa \cdot \text{decay}(x_2, \rho)$ κ: positive; ρ: between 0 and 1
2 rational-update volatility	Summary: "Agents keep their previous forecast for the same target and shade it down in proportion to how volatile recent objective news about the forecasted variable has been." y^*. $F_t[Y_{(t+1)}]$: "The agent's current forecast of the forecasted variable for the current target period." x_1. $F_{(t-1)}[Y_{(t+1)}]$: "The agent's forecast of the same target period, made one period earlier." x_2. N_j: "The one-period revision of an objective, forecast-blind model prediction of the target that was current j periods ago, made over the period ending j periods ago."	$x_1 - \beta \cdot \text{std}_k(x_2)$ β: positive; k: positive integer

Round	D_T **(verbatim)**	$\mathcal{F}_T$ **(formula and free parameters)**
3 worst recent return	Summary: "Agents keep their previous forecast for the same target and rescale it by the worst recent percentage change in a market price tied to the forecasted variable, reacting to downside news." y^*. $F_t[Y_{(t+1)}]$: "The agent's current forecast of the forecasted variable for the current target period." x_1. $F_{(t-1)}[Y_{(t+1)}]$: "The agent's previous forecast of the same target, made one forecasting period earlier." x_2. $R_j, j = 1, \ldots, J$: "The percentage return of the market price linked to the forecasted variable over each of the J most recent sub-periods, most recent first."	$x_1 + \beta \cdot \min_k(x_2) \cdot \text{abs}(x_1)$ β: positive; k: positive integer
3 stale forecast level	Summary: "Agents keep their earlier forecast for the same target and shade it down by a small fixed fraction of their earlier forecast for the period just realized." y^*. $F_t[Y_{(T)}]$: "The agents' current forecast of the forecasted variable's value for the current target period T." x_1. $F_{(t-1)}[Y_{(T)}]$: "The agents' forecast of the current target period's value, made one forecast period earlier." x_2. $F_{(t-1)}[Y_{(T-1)}]$: "The agents' earlier forecast of the most recently realized period's value, made about one period before that period ended."	$x_1 - \beta \cdot x_2$ β: none
4 saturating error correction	Summary: "Agents keep their previous forecast of the same target and nudge it upward by at most a fixed fraction of its size, the nudge rising smoothly with the latest surprise relative to that forecast." y^*. $F_t[Y_{(t+1)}]$: "The agents' current forecast of the forecasted variable for the upcoming target period." x_1. $F_{(t-1)}[Y_{(t+1)}]$: "The agents' forecast of the same target period, made one forecasting period earlier." x_2. e_t: "The most recently realized forecast error, the realized value minus the agents' earlier forecast of it."	$x_1 + \gamma \cdot \text{sgate}(\text{pdiv}(x_2, \text{abs}(x_1)), c, s) \cdot \text{abs}(x_1)$ γ: positive; c: none; s: positive

Round	D_T **(verbatim)**	$\mathcal{F}_T$ **(formula and free parameters)**
4 fixed-target revision momentum	Summary: "The forecast anchors on the earlier forecast of the same target and adds a fixed fraction of the average of its most recent revisions, a partial momentum adjustment." y^*. $F_t[Y_T]$: "The current consensus forecast of the target-period value of the forecasted variable." x_1. $F_{(t-1)}[Y_T]$: "The forecast of the same target period made one forecasting cycle earlier." x_2. R_j: "The successive recent revisions of the forecast for the same target period, indexed by how many revision intervals ago, most recent first."	$x_1 + \beta \cdot \text{mean}_k(x_2)$ β: none; k: positive integer
5 price trend	Summary: "Forecasters carry forward their previous forecast of the same target and scale it up or down in proportion to the recent percentage trend in a related market price." y^*. $F_t[Y_{(t+1)}]$: "The agents' current forecast of the forecasted variable for the current target period." x_1. $F_{(t-1)}[Y_{(t+1)}]$: "The agents' forecast of the same target period, made one forecasting period earlier." x_2. $P_j, j = 1, \ldots, J$: "The observed values of a market price tied to the forecasted variable, j sub-periods before the forecast, most recent first." x_3. P_1: "The most recently observed value of the market price tied to the forecasted variable."	$x_1 + \beta \cdot \text{pdiv}(\text{trend}_k(x_2), x_3) \cdot \text{abs}(x_1)$ β: none; k: positive integer
6 rolling-target revision momentum	Summary: "Agents begin from their earlier forecast of the same target and add a fraction of a geometrically weighted average of their own recent forecast revisions, so revisions carry momentum." y^*. $F_t[Y_{(t+1)}]$: "The agents' current forecast, made at t, of the forecasted variable's value one period ahead." x_1. $F_{(t-1)}[Y_{(t+1)}]$: "The agents' previous forecast, made one period earlier, of the same target value now being forecast." x_2. $R_j, j = 1, \ldots, J$: "The agents' past forecast revisions, each the change in their forecast over one period, indexed by how many periods ago, most recent first."	$x_1 + \beta \cdot \text{decay}(x_2, \lambda)$ β: none; λ: between 0 and 1

Round	D_T **(verbatim)**	$\mathcal{F}_T$ **(formula and free parameters)**
7 sticky (unless there's bad news) expectations	Summary: "Sticky updating: agents move their previous forecast of the target partway toward fresh objective signals, using decaying weights on recent signal gaps, and update more strongly after a large disappointing surprise." y^*. $F_t[Y_{(t+1)}]$: "The agent's current forecast, made at t, of the current target period's value of the forecasted variable." x_1. $F_{(t-1)}[Y_{(t+1)}]$: "The agent's previous forecast of the current target, made one period earlier." x_2. $G_j, j = 0, \ldots, J$: "For the target of j periods ago, the objective (forecast-blind) model prediction of it made then, minus the agent's forecast of it made one period before that; most recent first." x_3. e_t: "The agent's most recent forecast error, the latest realized value minus the agent's forecast of it."	$x_1 + \text{decay}(x_2, \rho) \cdot (\text{gain} + \text{extra_gain} \cdot \mathbf{1}(x_3 \leq \theta))$ ρ: between 0 and 1; gain: positive; extra_gain: none; θ: none

Notes. One row per induced function, in the order and with the names of Table A14. Each row is the most portable passing proposal of the portability loop (Section 4.6) for that function in the quarterly analyst context. D_T is copied verbatim; the formula and free parameters are as returned, typeset as math, with the operators defined in Table A6; in particular, $\text{decay}(x, \rho) = \sum_{j \geq 0} \rho^j x_j / \sum_{j \geq 0} \rho^j$ is the decay-weighted average of a family x whose most recent member is x_0; $\text{std}_k(x)$ is the population standard deviation of its k most recent members $x_0, \ldots, x_{k-1}$; $\text{pdiv}(a, b) = a/b$, set to 1 when $|b| < 10^{-6}$; and $\text{sgate}(x, c, s) = 1/\left(1 + e^{-(x-c)/|s|}\right)$, a logistic step that turns on around c with softness $|s|$.

Constant	vs. $\Delta F_{i,t}$ alone
Theory 1: $\beta\ \text{decay}_{\gamma}\left(g_{i,t}\right)$	
β	0.3914
γ	0.2392
intercept	−0.01633
Theory 2: $\beta\ \text{decay}_{\gamma}\left(\hat{\mathbb{E}}_t[Y_{i,t+1}] - \hat{\mathbb{E}}_{t-1}[Y_{i,t+1}]\right)$	
β	0.5273
γ	0.2958
intercept	−0.03068
Theory 3: $\beta\,\text{std}\Big(\big\{\hat{\mathbb{E}}_{t-j}[Y_{i,t-j+1}] - \hat{\mathbb{E}}_{t-j-1}[Y_{i,t-j+1}]\big\}_{j=0}^{k}\Big)$	
β	−0.2183
k	4
intercept	−0.01616
Theory 4: $\beta\Big(\min_{\ell\in\{1,\dots,k\}} r_{i,t,\ell}\Big)\,\big\lvert F_{t-1}[Y_{i,t+1}]\big\rvert$	
β	0.5215
k	3
intercept	−0.01104
Theory 5: $\beta\, F_{i,t-1,-3}\big[Y_{i,t}\big]$	
β	−0.03494
intercept	−0.01798

Constant	vs. $\Delta F_{i,t}$ alone
Theory 6: $\beta\,\big\lvert F_{t-1}[Y_{i,t+1}]\big\rvert\,\sigma\Big(\big(e_{i,t}/\big\lvert F_{t-1}[Y_{i,t+1}]\big\rvert - c_1\big)/c_2\Big)$	
β	0.4808
c_1	0.869
c_2	0.25
intercept	−0.04246
Theory 7: $\beta\,\frac{1}{k}\sum_{j=0}^{k-1}\left(F_{i,t-1,-j}[Y_{i,t+1}] - F_{i,t-1,-(j+1)}[Y_{i,t+1}]\right)$	
β	0.4209
k	2
intercept	−0.0272
Theory 8: $\beta\, s_{i,t,1:4}\,\big\lvert F_{t-1}[Y_{i,t+1}]\big\rvert$	
β	0.532
intercept	−0.02836
Theory 9: $\beta\ \text{decay}_{\gamma}\left(\Delta F_{i,t-1}\right)$	
β	0.3926
γ	0.3045
intercept	−0.02034
Theory 10: $\beta\left(1+\delta\,\mathbf{1}\{e_{i,t}\le-\kappa\}\right)\ \text{decay}_{\gamma}\left(g_{i,t}\right)$	
β	0.3141
δ	0.541
γ	0.2453
κ	0.05
intercept	−0.01346

Notes. Companion to Table A14. One block per induced quarterly theory, in round order, headed by the theory's expression in the notation of Table A14. Rows are that notation's symbols (β, γ, δ, κ, ...) plus the fit's intercept; each symbol's value is the identified combination of the underlying fit constants — e.g. a scale pair collapsed in the display enters only through β, which shows the product of the outer evaluation weight and the expression's internal scale (only the product is identified), and δ and κ are the ratio and threshold combinations of theory 10's raw constants. The column is the theory alone, against the forecast-revision target $\Delta F_{i,t}$: each theory's solo refit on fold $f5$, all constants free. All fits use the N = 57,000 fold-$f5$ firm–quarters. Values are points only, target ΔF only. Theory 6's decay constant, as returned by the search, is 0 (its aggregate reduces to the latest forecast error, as the header equation shows); the refits treat it as free like any other constant and move it slightly (within its interval of zero).

Table A16: The fitted constants of the ten theories, each fit alone, against $\Delta F_{i,t}$ on the held-out fold 5.

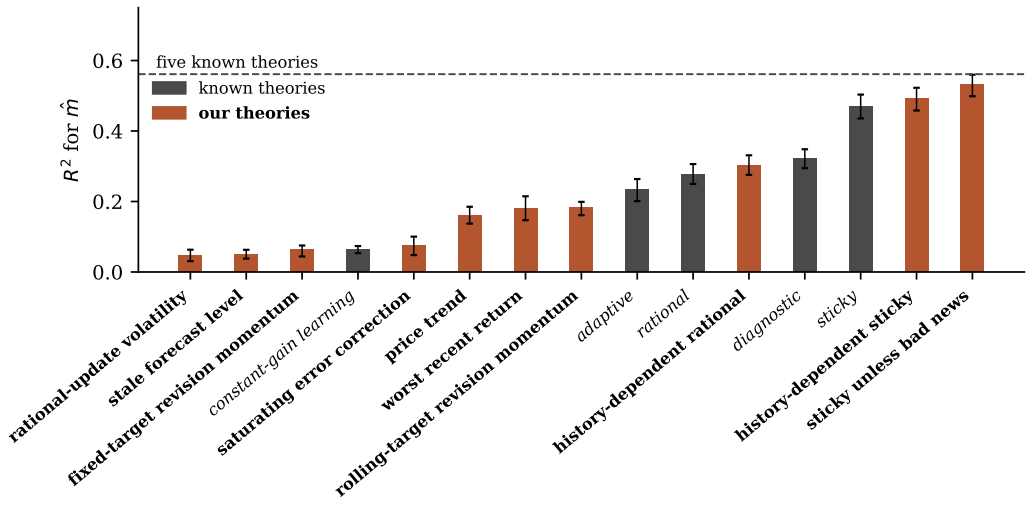
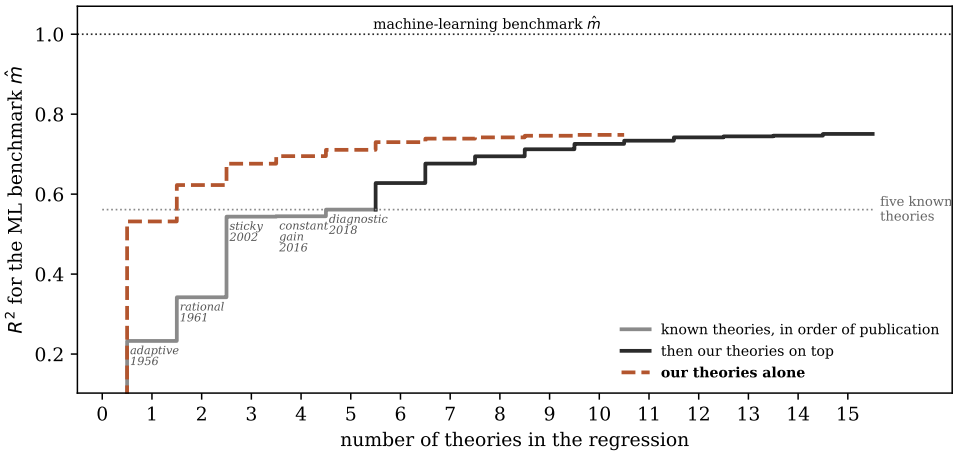


Figure A14: Every theory fit alone, with the ML benchmark $\hat{m}$ as the target: coefficients fit to $\hat{m}$ on folds 1–4, R^2 against $\hat{m}$ on fold 5.



Notes. As Figure 14, but each linear regression is fit to $\hat{m}$ on folds 1–4 and scored against $\hat{m}$ on fold 5, the constants inside our theories are refit jointly against $\hat{m}$, and both greedy orders are re-chosen against $\hat{m}$.

Figure A15: Figure 14 with the ML benchmark $\hat{m}$ as the target.

	Quarterly				Annual			
	Alone		On known		Alone		On known	
Step	Theory	Cum. R^2	Theory	Cum. R^2	Theory	Cum. R^2	Theory	Cum. R^2
1	sticky (unless there's bad news) expectations	0.198	worst recent return	0.233	sticky (unless there's bad news) expectations	0.637	price trend	0.651
2	history-dependent rational	0.234	rolling-target revision momentum	0.255	saturating error correction	0.641	worst recent return	0.654
3	worst recent return	0.247	stale forecast level	0.262	worst recent return	0.650	saturating error correction	0.649
4	rolling-target revision momentum	0.255	saturating error correction	0.267	history-dependent rational	0.649	sticky (unless there's bad news) expectations	0.656
5	saturating error correction	0.258	sticky (unless there's bad news) expectations	0.269	price trend	0.651	history-dependent sticky expectations	0.659
6	stale forecast level	0.269	history-dependent rational	0.270	history-dependent sticky expectations	0.654	rolling-target revision momentum	0.662
7	price trend	0.271	price trend	0.272	rolling-target revision momentum	0.659	stale forecast level	0.662
8	history-dependent sticky expectations	0.272	fixed-target revision momentum	0.274	stale forecast level	0.660	rational-update volatility	0.664
9	fixed-target revision momentum	0.274	history-dependent sticky expectations	0.274	rational-update volatility	0.662	fixed-target revision momentum	0.664
10	rational-update volatility	0.275	rational-update volatility	0.274	fixed-target revision momentum	0.661	history-dependent rational	0.664

Notes. The order in which our theories enter Figures 14 (quarterly) and 19 (annual), target the forecast revision. "Alone": our theories with no known theory, chosen by pooled out-of-fold R^2 within folds 1–4. "On known": our theories added on top of the five known theories, chosen by in-sample R^2 on folds 1–4 of [the theories so far + the five known]. At every step the constants inside our theories are refit jointly on folds 1–4. Cumulative R^2 is scored on held-out fold 5 (N = 57,000 firm–quarters; 10,634 firm–years). The five known theories together reach 0.218 (quarterly) and 0.643 (annual).

Table A17: The greedy order under each target, chosen on the folds 1–4 rotation.

G Annual forecasts

Transfer is formalized in the following procedure:

Procedure: Evaluating transfer

Given theories $T_1, \ldots, T_n$ and a new context C' with target y' and data W'.

Step 1. Bind each theory into the new context,

$$B_{T_1,C'},\ \ldots,\ B_{T_n,C'}.$$

Step 2. Report the R^2 of the collection in the new context: the R^2 of one linear regression of y' on all n bound theories (as in Section 5.1), estimated on W'.

We will carry out the bindings by hand rather than use an LLM binder. This is because there are only 10 theories, unlike inside the induction procedure, where the binder and the portability loop need to be called countless times. The re-bound equations are in Table A19. Summary statistics for the annual panel are in Table A18.

Panel A: Sample

Firms	13,281
Firm–years	110,963
Years per firm (median)	5
Sample period	1984–2024

Panel B: Variables

	N	Mean	SD	P10	P50	P90
Forecast $F_t[Y_{i,t+1}]$	110,963	1.55	2.31	−0.39	1.16	3.85
Realized outcome $Y_{i,t+1}$	110,963	1.38	2.59	−0.81	1.02	3.88
Forecast error $Y_{i,t+1} - F_t[Y_{i,t+1}]$	110,963	−0.18	1.13	−0.93	−0.04	0.44
Forecast revision $\Delta F_{i,t}$	75,524	−0.22	0.93	−1.00	−0.10	0.37
Analysts per consensus	110,963	8.1	7.3	1	6	19

Notes. Annual earnings per share in split-adjusted dollars per share; values are not deflated by the share price. The forecast is the consensus one fiscal year ahead, recorded at the statement date of the annual cycle. N varies with each variable's coverage: realized earnings cover every firm–year of the panel; the forecast and analyst count require the corresponding consensus statistics at the statement date; the forecast revision additionally requires a valid link to the prior year's consensus snapshot, hence its smaller N. Consensus-based rows inherit the monthly panel's per-month 1%/99% winsorization; analyst counts are raw. The annual counterpart of Table A2.

Table A18: Summary statistics for the annual panel. Full panel: 110,963 firm-years.

Theory	Bound expression, annual
1	$\beta\ \text{decay}_\gamma\left(g_{i,\tau}\right)$
2	$\beta\ \text{decay}_\gamma\left(\hat{\mathbb{E}}_\tau[Y_{i,\tau+1}] - \hat{\mathbb{E}}_{\tau-1}[Y_{i,\tau+1}]\right)$
3	$\beta\,\text{std}\Big(\left\{\hat{\mathbb{E}}_{\tau-j}[Y_{i,\tau-j+1}] - \hat{\mathbb{E}}_{\tau-j-1}[Y_{i,\tau-j+1}]\right\}_{j=0}^{k}\Big)$
4	$\beta\Big(\min_{\ell\in\{1,\ldots,k\}} r_{i,\tau,\ell}\Big)\left\lvert F_{\tau-1}[Y_{i,\tau+1}]\right\rvert$
5	$\beta\,F_{i,\tau-1,-12}\left[Y_{i,\tau}\right]^\dagger$
6	$\beta\left\lvert F_{\tau-1}[Y_{i,\tau+1}]\right\rvert\,\sigma\Big(\left(e_{i,\tau}/\left\lvert F_{\tau-1}[Y_{i,\tau+1}]\right\rvert - c_1\right)/c_2\Big)$
7	$\beta\left(F_{i,\tau-1,0}[Y_{i,\tau+1}] - F_{i,\tau-1,-1}[Y_{i,\tau+1}]\right)^\dagger$
8	$\beta\,s_{i,\tau,1:12}\left\lvert F_{\tau-1}[Y_{i,\tau+1}]\right\rvert$
9	$\beta\ \text{decay}_\gamma\left(\Delta F_{i,\tau-1}\right)$
10	$\beta\left(1+\delta\,\mathbf{1}\{e_{i,\tau}\le-\kappa\}\right)\ \text{decay}_\gamma\left(g_{i,\tau}\right)$

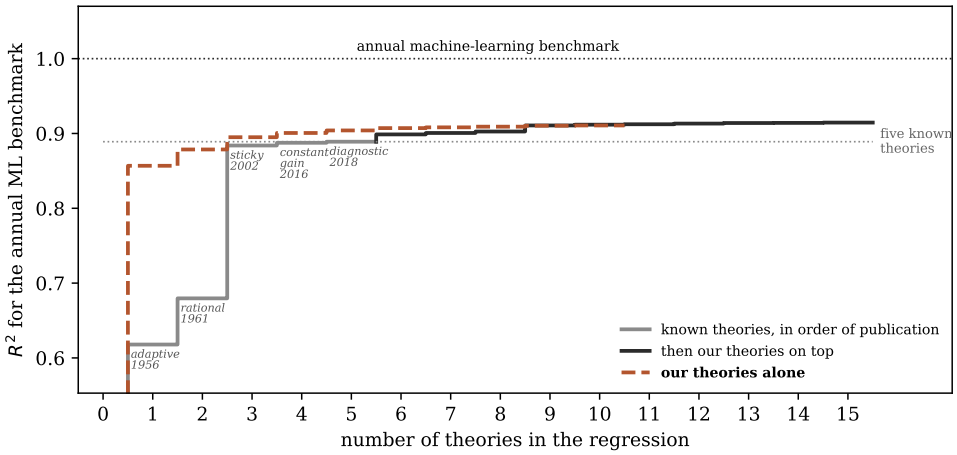
Notes. One row per function returned by function search, in the order of Table A14, bound into the annual context; the column shows the bound annual expression (the quarterly forms are in Table A14). All ten functions bound — no row lacks an annual analog. † marks the 2 bindings whose horizon scaling we corrected on a second pass (lags and horizons must scale with the forecast cycle). All ten were bound by hand from the theories' words. Expressions are shown in equation form; $\text{decay}_\gamma(x_{i,t}) = \sum_{k\ge0}\gamma^k x_{i,t-k}\big/\sum_{k\ge0}\gamma^k$ denotes the decay-weighted average; τ indexes annual periods, with month-denominated lags and windows rescaled where the binding changed them. Fixed-target revision momentum uses the one available monthly revision of the current fiscal-year forecast ($k=1$). The decay weight γ of rolling-target revision momentum is restricted to $[0,1]$ when refit, as its words require. Fit is reported in Figures 18 and 19, not here.

Table A19: The theories bound in the annual context. Constants fit on folds 1–4 of the annual panel.

Category	Variable	Lag clock	Most recent	Lag depth	Columns
Announced earnings	quarterly EPS	quarterly	latest announced by F_t	13 quarters	37
Announced earnings	quarterly EPS	annual	at F_{t-1}	8 years	22
Announced earnings	annual EPS	annual	latest announced by F_t	9 years	25
Consensus forecasts, monthly snapshots	forecast level, horizons $q1, q2, q3, y1, y2$	monthly	at F_{t-1}	13 months	65
Consensus forecasts, monthly snapshots	month-over-month revision, horizons $q1, q2, q3, y1, y2$	monthly	into F_{t-1}	12 months	60
Consensus forecasts, annual snapshots	forecast level, horizons $y1, y2$	annual	at F_{t-1}	8 years	16
Consensus forecasts, annual snapshots	forecast error, realized minus consensus	annual	latest announced before F_t	8 years	8
Consensus forecasts, annual snapshots	snapshot-over-snapshot revision	annual	at F_{t-1}	8 years	8
Fundamentals	the 99 accounting ratios and market variables of Zhang, Zhu, and Linnainmaa (2025): the WRDS financial-ratio suite, CRSP market variables, and per-share fundamentals recovered from the ratios	monthly	the month before F_t	12 months	3,366
Fundamentals	the same 99	quarterly	the quarter before F_t	12 quarters	3,366
Fundamentals	the same 99	annual	at F_{t-1}	8 years	2,178
Disagreement	forecast standard deviation, per horizon	monthly	at F_{t-1}	13 months	65
Disagreement	forecast high minus low, per horizon	monthly	at F_{t-1}	13 months	65
Disagreement	forecast mean minus median, per horizon	monthly	at F_{t-1}	13 months	65
Disagreement	analysts in the consensus, per horizon	monthly	at F_{t-1}	13 months	65
Disagreement	analysts revising up, per horizon	monthly	at F_{t-1}	13 months	65
Disagreement	analysts revising down, per horizon	monthly	at F_{t-1}	13 months	65
Coverage history	length of the current coverage run, the gap in coverage before it, number of earlier coverage runs, coverage breaks in the past year	—	—	lagless	5
Total					9,546

Notes. In this table t indexes fiscal years: rows are the annual-cycle quarters (the quarter after each fiscal-year report); F_t is recorded on this cycle's statement date and F_{t-1} on the previous cycle's, roughly one year back. The predicted revision is the annual one, from F_{t-1} to F_t, so consensus and disagreement readings stop at F_{t-1} — anything later is a partial sum of the target — and the quarterly-snapshot consensus block does not enter at all. Quarter steps walk the firm's quarterly statement dates ("the quarter before F_t"); fundamentals run up to the month before F_t (month-end-dated, hence the one-month delay); announced earnings are announcement-dated, entering the moment announced, up to F_t itself — including the past forecast errors, whose consensus component sits at the year's opening statement. Source: the replication package.

Table A20: The raw feature set, annual target.



Notes. As Figure 19, but every regression, including the joint refit of the constants inside our theories, is fit against the annual ML benchmark, and both greedy orders are re-chosen against it.

Figure A16: Figure 19 with the annual ML benchmark as the target.

References

Hassan Afrouzi, Spencer Y. Kwon, Augustin Landier, Yueran Ma, and David Thesmar. Overreaction in expectations: Evidence and theory. *Quarterly Journal of Economics*, 138(3):1713–1764, 2023. URL https://doi.org/10.1093/qje/qjad009.

Isaiah Andrews, Drew Fudenberg, Lihua Lei, Annie Liang, and Chaofeng Wu. The transfer performance of economic models. Working paper, 2025. URL https://arxiv.org/abs/2202.04796.

Gary S. Becker. Crime and punishment: An economic approach. *Journal of Political Economy*, 76 (2):169–217, 1968. doi: 10.1086/259394.

Gary S. Becker. A theory of marriage: Part I. *Journal of Political Economy*, 81(4):813–846, 1973.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

Francesco Bianchi, Sydney C. Ludvigson, and Sai Ma. Belief distortions and macroeconomic fluctuations. *American Economic Review*, 112(7):2269–2315, 2022. URL https://doi.org/10.1257/aer.20201713.

Pedro Bordalo, Nicola Gennaioli, and Andrei Shleifer. Diagnostic expectations and credit cycles. *Journal of Finance*, 73(1):199–227, 2018. URL https://doi.org/10.1111/jofi.12586.

Pedro Bordalo, Nicola Gennaioli, Rafael La Porta, and Andrei Shleifer. Diagnostic expectations and stock returns. *Journal of Finance*, 74(6):2839–2874, 2019. URL https://doi.org/10.1111/jofi.12833.

Pedro Bordalo, Nicola Gennaioli, Yueran Ma, and Andrei Shleifer. Overreaction in macroeconomic expectations. *American Economic Review*, 110(9):2748–2782, 2020. URL https://doi.org/10.1257/aer.20181219.

Jean-Philippe Bouchaud, Philipp Krüger, Augustin Landier, and David Thesmar. Sticky expectations and the profitability anomaly. *Journal of Finance*, 74(2):639–674, 2019. URL https://doi.org/10.1111/jofi.12734.

Phillip Cagan. The monetary dynamics of hyperinflation. In Milton Friedman, editor, *Studies in the Quantity Theory of Money*, pages 25–117. University of Chicago Press, 1956.

Colin F. Camerer. Artificial intelligence and behavioral economics. In Ajay Agrawal, Joshua Gans, and Avi Goldfarb, editors, *The Economics of Artificial Intelligence: An Agenda*, pages 587–608. University of Chicago Press, 2019.

Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997. URL https://link.springer.com/article/10.1023/A:1007379606734.

Charles T. Clotfelter and Philip J. Cook. The "gambler's fallacy" in lottery play. *Management Science*, 39(12):1521–1525, 1993. doi: 10.1287/mnsc.39.12.1521.

Olivier Coibion and Yuriy Gorodnichenko. What can survey forecasts tell us about information rigidities? *Journal of Political Economy*, 120(1):116–159, 2012. URL https://doi.org/10.1086/665662.

Olivier Coibion and Yuriy Gorodnichenko. Information rigidity and the expectations formation process: A simple framework and new facts. *American Economic Review*, 105(8):2644–2678, 2015. URL https://doi.org/10.1257/aer.20110306.

Olivier Coibion, Yuriy Gorodnichenko, and Rupal Kamdar. The formation of expectations, inflation, and the Phillips curve. *Journal of Economic Literature*, 56(4):1447–1491, 2018a. URL https://doi.org/10.1257/jel.20171300.

Olivier Coibion, Yuriy Gorodnichenko, and Saten Kumar. How do firms form their expectations? New survey evidence. *American Economic Review*, 108(9):2671–2713, 2018b. doi: 10.1257/aer.20151299.

David Court, Benjamin Gillen, Jordi McKenzie, and Charles R. Plott. Two information aggregation mechanisms for predicting the opening weekend box office revenues of films. *Economic Theory*, 65(1):25–54, 2018. doi: 10.1007/s00199-017-1036-1.

Miles Cranmer. Interpretable machine learning for science with PySR and SymbolicRegression.jl, 2023. arXiv:2305.01582.

Miles Cranmer, Alvaro Sanchez-Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. In *Advances in Neural Information Processing Systems*, volume 33, 2020. arXiv:2006.11287.

Dean Croushore. Introducing: The survey of professional forecasters. *Federal Reserve Bank of Philadelphia Business Review*, pages 3–15, November/December 1993.

Francesco D'Acunto, Ulrike Malmendier, Juan Ospina, and Michael Weber. Exposure to grocery prices and inflation expectations. *Journal of Political Economy*, 129(5):1615–1639, 2021. doi: 10.1086/713192.

Ricardo De la O and Sean Myers. Subjective cash flow and discount rate expectations. *Journal of Finance*, 76(3):1339–1387, 2021. URL https://doi.org/10.1111/jofi.13016.

Tim de Silva and David Thesmar. Noise in expectations: Evidence from analyst forecasts. *Review of Financial Studies*, 37(5):1494–1537, 2024. URL https://doi.org/10.1093/rfs/hhad091.

Pierre Duhem. *La Théorie Physique: Son Objet et Sa Structure*. Chevalier et Rivière, Paris, 1906.

Thorsten M. Egelkraut, Bruce J. Sherrick, Philip Garcia, and Joost M. E. Pennings. Farmers' subjective perceptions of yield and yield risk. In *Proceedings of the NCR-134 Conference on Applied Commodity Price Analysis, Forecasting, and Market Risk Management*, St. Louis, MO, 2006. URL https://ageconsearch.umn.edu/record/18991.

Keaton Ellis, Shachar Kariv, and Erkut Y. Ozbay. Scaling up: Individual-level transfer performance of models. Working paper, University of California, Berkeley, 2024.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

Larry G. Epstein and Martin Schneider. Ambiguity, information quality, and asset pricing. *Journal of Finance*, 63(1):197–228, 2008. URL https://doi.org/10.1111/j.1540-6261.2008.01314.x.

Drew Fudenberg and Annie Liang. Predicting and understanding initial play. *American Economic Review*, 109(12):4112–4141, 2019. URL https://www.aeaweb.org/articles?id=10.1257/aer.20180654.

Drew Fudenberg and Indira Puri. Evaluating and extending theories of choice under risk. Working paper, MIT, 2023.

Drew Fudenberg, Jon Kleinberg, Annie Liang, and Sendhil Mullainathan. Measuring the completeness of economic models. *Journal of Political Economy*, 130(4):956–990, 2022. URL https://www.journals.uchicago.edu/doi/abs/10.1086/718371.

Drew Fudenberg, Wayne Gao, and Annie Liang. How flexible is that functional form? Quantifying the restrictiveness of theories. *Review of Economics and Statistics*, 108(1):194–209, 2026. URL https://direct.mit.edu/rest/article-abstract/108/1/194/118340/.

Andreas Fuster, David Laibson, and Brock Mendel. Natural expectations and macroeconomic fluctuations. *Journal of Economic Perspectives*, 24(4):67–84, 2010. URL https://doi.org/10.1257/jep.24.4.67.

Nicola Gennaioli, Yueran Ma, and Andrei Shleifer. Expectations and investment. In *NBER Macroeconomics Annual 2015, Volume 30*, pages 379–431. University of Chicago Press, 2016. URL https://doi.org/10.1086/685965.

Nelson Goodman. *Fact, Fiction, and Forecast*. Harvard University Press, Cambridge, MA, 1955.

Robin Greenwood and Andrei Shleifer. Expectations of returns and expected returns. *Review of Financial Studies*, 27(3):714–746, 2014. URL https://doi.org/10.1093/rfs/hht082.

David W. Harless and Colin F. Camerer. The predictive utility of generalized expected utility theories. *Econometrica*, 62(6):1251–1289, 1994. URL https://www.econometricsociety.org/publications/econometrica/1994/11/01/predictive-utility-generalized-expected-utility-theories.

Toshihiko Hirasawa, Michihiro Kandori, and Akira Matsushita. Using big data and machine learning to uncover how players choose mixed strategies. Working paper, University of Tokyo Market Design Center, UTMD-084, 2025.

David Hume. *An Enquiry Concerning Human Understanding*. A. Millar, London, 1748. First published as *Philosophical Essays Concerning Human Understanding*; retitled in 1758.

John Jumper et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596 (7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979. URL https://www.jstor.org/stable/1914185.

John R. Koza. Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems. Technical Report STAN-CS-90-1314, Department of Computer Science, Stanford University, June 1990.

John R. Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, 4(2):87–112, 1994. doi: 10.1007/BF00175355.

Remi Lam et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677): 1416–1421, 2023. doi: 10.1126/science.adi2336.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. doi: 10.1038/nature14539.

Pablo Lemos, Niall Jeffrey, Miles Cranmer, Shirley Ho, and Peter Battaglia. Rediscovering orbital mechanics with machine learning. *Machine Learning: Science and Technology*, 4(4):045002, 2023. doi: 10.1088/2632-2153/acfa63.

Annie Liang. Using machine learning to generate, clarify, and improve economic models, 2025. arXiv:2508.19136.

Robert E. Lucas. Expectations and the neutrality of money. *Journal of Economic Theory*, 4(2): 103–124, 1972. URL `https://doi.org/10.1016/0022-0531(72)90142-1`.

Robert E. Lucas, Jr. Asset prices in an exchange economy. *Econometrica*, 46(6):1429–1445, 1978.

Jens Ludwig and Sendhil Mullainathan. Machine learning as a tool for hypothesis generation. *Quarterly Journal of Economics*, 139(2):751–827, 2024. URL `https://academic.oup.com/qje/article-abstract/139/2/751/7515309`.

Ulrike Malmendier and Stefan Nagel. Learning from inflation experiences. *Quarterly Journal of Economics*, 131(1):53–87, 2016. doi: 10.1093/qje/qjv037.

N. Gregory Mankiw and Ricardo Reis. Sticky information versus sticky prices: A proposal to replace the new Keynesian Phillips curve. *Quarterly Journal of Economics*, 117(4):1295–1328, 2002. URL `https://doi.org/10.1162/003355302320935034`.

Benjamin S. Manning, Kehang Zhu, and John J. Horton. Automated social science: Language models as scientist and subjects. Working Paper 32381, National Bureau of Economic Research, 2024.

Lloyd A. Metzler. The nature and stability of inventory cycles. *Review of Economic Statistics*, 23 (3):113–129, 1941.

Jan Mossin. Aspects of rational insurance purchasing. *Journal of Political Economy*, 76:553–568, 1968.

Sendhil Mullainathan and Ashesh Rambachan. Science in the age of algorithms. In Ajay K. Agrawal, Erik Brynjolfsson, and Anton Korinek, editors, *The Economics of Transformative AI*. University of Chicago Press, 2025. NBER conference volume chapter, forthcoming.

Sendhil Mullainathan and Ashesh Rambachan. From predictive algorithms to the automatic generation of anomalies. *Econometrica*, forthcoming. URL `https://www.econometricsociety.org/publications/econometrica/forthcoming-papers/0000/00/00/From-Predictive-Algorithms-to-the-Automatic-Generation-of-Anomalies/file/22958-3.pdf`.

John F. Muth. Rational expectations and the theory of price movements. *Econometrica*, 29(3): 315–335, 1961. URL `https://doi.org/10.2307/1909635`.

Ignacio Palacios-Huerta. Professionals play minimax. *Review of Economic Studies*, 70(2):395–415, 2003. doi: 10.1111/1467-937X.00249.

Alexander Peysakhovich and Jeffrey Naecker. Using methods from machine learning to evaluate behavioral models of choice under risk and ambiguity. *Journal of Economic Behavior & Organization*, 133:373–384, 2017. URL https://www.sciencedirect.com/science/article/abs/pii/S0167268116301846.

Willard V. O. Quine. Two dogmas of empiricism. *Philosophical Review*, 60(1):20–43, 1951.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009. URL https://doi.org/10.1126/science.1165893.

Reinhard Selten. Properties of a measure of predictive success. *Mathematical Social Sciences*, 21(2):153–167, 1991. URL https://www.sciencedirect.com/science/article/abs/pii/0165489691900764.

Christopher A. Sims. Implications of rational inattention. *Journal of Monetary Economics*, 50(3): 665–690, 2003. URL https://doi.org/10.1016/S0304-3932(03)00029-1.

Ralph Stinebrickner and Todd R. Stinebrickner. Academic performance and college dropout: Using longitudinal expectations data to estimate a learning model. *Journal of Labor Economics*, 32(3): 601–644, 2014. doi: 10.1086/675308.

Todd Stinebrickner and Ralph Stinebrickner. Learning about academic ability and the college dropout decision. *Journal of Labor Economics*, 30(4):707–748, 2012. doi: 10.1086/666525.

Silviu-Marian Udrescu and Max Tegmark. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020. URL https://doi.org/10.1126/sciadv.aay2631.

USDA National Agricultural Statistics Service. Agricultural yield survey. https://www.nass.usda.gov/Surveys/Guide_to_NASS_Surveys/Agricultural_Yield/, 2026. Accessed October 28, 2026.

Jules H. van Binsbergen, Xiao Han, and Alejandro Lopez-Lira. Man versus machine learning: The term structure of earnings expectations and conditional biases. *Review of Financial Studies*, 36 (6):2361–2396, 2023. URL https://doi.org/10.1093/rfs/hhac085.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS 2017)*, 2017.

Justin Wolfers and Eric Zitzewitz. Prediction markets. *Journal of Economic Perspectives*, 18(2): 107–126, 2004.

Michael Woodford. Imperfect common knowledge and the effects of monetary policy. In Philippe Aghion, Roman Frydman, Joseph Stiglitz, and Michael Woodford, editors, *Knowledge, Information, and Expectations in Modern Macroeconomics: In Honor of Edmund S. Phelps*. Princeton University Press, 2003.

Yingguang Zhang, Yandi Zhu, and Juhani T. Linnainmaa. Man versus machine learning revisited. *Review of Financial Studies*, 38(12):3768–3790, 2025. URL https://doi.org/10.1093/rfs/hhaf066.