Can LLMs Price Optimally? Evidence from a Large-Scale Pricing Benchmark

Qiqi Hao[1], Enfal Arisoy[2], Michael Hamilton[3], Övünç Yılmaz[2],
Maxime C. Cohen[4], Titing Cui[5], Alejandro Lim[1], Xindi Shao[1]
[1]University of Pittsburgh, [2]University of Colorado, [3]Baruch College, [4]McGill University, [5]University of Tulsa
qih42@pitt.edu, ovunc.yilmaz@colorado.edu, michael.hamilton@baruch.cuny.edu, maxccohen@gmail.com, tic54@pitt.edu

Pricing is a critical part of many businesses' operations; however, setting effective prices is difficult, requiring practitioners to synthesize quantitative historical data with qualitative information about products and markets. Given the complexity of price-setting, there is growing interest in using AI technologies such as large language models (LLMs) to distill market information into profit-maximizing prices. However, their performance is difficult to evaluate in practice because pricing environments vary widely and revenue-maximizing prices are generally unobserved. To understand LLM pricing performance, we construct a large-scale benchmark of single-product pricing instances. Each instance is generated from a demand function with a known optimal price and contains historical prices, noisy demand observations, and contextual information. Using our benchmark, we evaluate six frontier LLMs in basic and advanced tiers at recommending next-period prices over all instances under different prompt variants, keeping demand functions and optimal prices hidden. We compare their recommendations across models as well as with regression-based and best-observed-price heuristics. On standard static instances, we find that basic models underperform the heuristics, incurring mean profit losses of 6.6–10.6%, compared with 4.3% for regression. By contrast, all three advanced models significantly outperform regression, with mean losses of 2.6–3.7%. On instances with more complex demand functions, seasonality, or competition, advanced models no longer consistently outperform the heuristics, and their pricing approaches and performance vary substantially across models and environments. We also find evidence that targeted prompt changes may improve performance for some models, suggesting a role for careful prompt design and human guidance in LLM-based pricing.

Key words: Large Language Models, Pricing Agents, Benchmarking, Artificial Intelligence
History: First Version: November 3, 2026. This Version: November 3, 2026

1. Introduction

Pricing is among the most consequential decisions a firm makes. Small changes in price can have disproportionately large effects on profit (Baker et al. 2014), yet pricing decisions are frequently made under uncertainty, with incomplete information about customer demand, and with limited opportunity for experimentation (den Boer 2015). Even when firms possess ample historical data, converting that data into effective prices requires estimating the price-demand relationship, optimizing the price over the uncertainty in that estimation, and understanding how the market of

customers and competitors will react to price changes (Cohen et al. 2022). Thus, pricing requires a complex mixture of both quantitative and qualitative reasoning about the market. To cope with this complexity, in practice, human price-setters often default to simple heuristics or ad hoc pricing strategies such as cost-plus pricing (sometimes referred to as *markup pricing*), break-even pricing, or target-return pricing, all of which may be inefficient (Rusetski 2014).

Given these inefficiencies, firms and researchers have shown sustained interest in developing systems to support pricing decisions (Spann et al. 2025). At the forefront of such systems are large language models (LLMs), which are ostensibly trained on a wealth of research in revenue management and pricing, case studies, and business reports, and whose flexible interface makes them easy to query and deploy. Indeed, today a pricing manager can describe their product, provide historical pricing and sales data, add information about their customer base and competitors, and ask an LLM to recommend a price to use going forward. Further, unlike a conventional pricing model built for a single environment, a general-purpose LLM can respond immediately to a wide variety of descriptions and data formats, and also explain its pricing recommendation in natural language (Cohen 2026).

These advantages have led to significant interest in systems that use LLMs either as interfaces to specialized pricing tools or as autonomous pricing agents (Ma et al. 2026). For example, the start-ups *7Learnings* and *Buynomics* both offer AI-powered predictive-pricing platforms that use LLM assistants to generate and communicate pricing recommendations (7Learnings 2026, Buynomics 2026). Similarly, *Salesforce* offers LLM-powered agents to assist customers with pricing and discount decisions through its Agentforce Sales program (Salesforce 2026). However, evaluating the true performance of these systems can be challenging.

First, any pricing recommendation (LLM-generated or not) is difficult to assess as the profit that would have been obtained at alternative prices is typically not observed in practice. A successful outcome at a specific price in a given period therefore does not reveal whether that price was optimal or near-optimal. Second, pricing problems differ immensely across products and market contexts. An agent's strong performance in one setting and market environment need not generalize to others (Han et al. 2026). Third, the capabilities and behavior of commercial LLMs can change as the models evolve, so their pricing performance must be reassessed over time (Fish et al. 2025).

Motivated by these challenges, we take a first step toward measuring the effectiveness of modern LLMs as pricing agents by constructing a large-scale benchmark of controlled pricing instances whose ground-truth optimal prices are known and computable. Each instance is generated from a hidden demand function and provides a history of prices and noisy demand realizations, together with *side information* about the product and market conditions such as seasonality and competition. The benchmark contains 432 instances spanning eight demand families, three market

environments, and multiple parameter configurations, history lengths, and noise levels, with additional variations in contextual framing and pricing task presentation. This design preserves the essential features of a pricing task in practice, including limited information, noisy demand observations, model uncertainty, and the need to synthesize data with contextual information, while making objective evaluation of pricing performance possible.

We then use our benchmark to evaluate six leading general-purpose LLMs from OpenAI (ChatGPT), Google (Gemini), and Anthropic (Claude) in two capability tiers, basic and advanced. Based on the history and side information in each instance, we ask each model to recommend a profit-maximizing price for the next period. We evaluate the expected profit generated by each recommendation relative to the true optimum and compare the LLMs performance with one another and with two heuristics: a *best-observed-price heuristic* which selects the historical price that generated the highest observed revenue, and a *regression heuristic* which fits a linear demand model to the historical data and optimizes the price under the fitted model. The heuristics act as a baseline level of performance on our pricing tasks. Our aim is to characterize where these models succeed and fail at pricing, understand differences in their performance, and examine how prompt design can improve their recommendations. To help explain these patterns, we examine the models' stated reasoning, focusing on how they select and fit demand models, incorporate contextual information, and calculate prices. By identifying common pricing biases and model-specific tendencies, we seek to develop prompt-design strategies and provide managerial guidance on how practitioners can obtain better performance from LLM pricing assistants.

1.1. Our Contributions

A summary of our main contributions is as follows:

1. We build a large-scale benchmark of single-product pricing instances to systematically evaluate the performance of LLMs as pricing agents. Each instance is a next-period pricing problem defined by a demand function, a market environment, and a finite history of prices and noisy demand observations. The benchmark contains 432 unique instances spanning eight demand families, three market environments (static, seasonal, and competitive), and multiple parameter sets, noise levels, and history lengths. We then use these instances to evaluate six LLMs in a controlled experiment, presenting each pricing instance as a prompt while keeping the underlying demand functions and optimal prices hidden. For a subset of these prompts, we vary role framing, product context, and distribution hints while holding the underlying pricing tasks fixed. We compare the expected payoff of each LLM pricing recommendation with the full-information optimal price. Including repeated queries, our evaluation of LLM pricing capability comprises 10,584 price recommendations across the benchmark.

2. We establish a baseline of performance by examining the six LLMs across four basic demand families in a static single-seller market. On these standard instances, we compare the LLMs' performance with that of a regression heuristic, which fits a linear demand model and optimizes the price according to the estimated regression. We find that all basic LLMs (Claude Haiku 4.5, Gemini 2.5 Flash, GPT-5 mini) significantly underperform the regression heuristic, whereas all advanced LLMs (Claude Fable 5.1, Gemini 3.1 Pro, GPT-6 Astra) significantly outperform it. Mean profit losses are 6.6–10.6% for basic LLMs and 2.6–3.7% for advanced LLMs, compared with 4.3% for regression; the advanced models reduce mean loss by 0.6–1.7 percentage points (all $p < 0.05$). While all models frequently report fitting a regression to guide their pricing decisions in their rationales, only the advanced LLMs consistently compute the OLS coefficients accurately. Among responses reporting a linear intercept and slope, mean absolute relative slope errors are 22.1% for basic LLMs and 5.3% for advanced LLMs. Moreover, the advanced models go beyond the regression: Claude Fable 5.1 often adjusts the regression-optimal price after checking observed revenues (its price departs by more than 2% from the optimum implied by its reported linear fit in 51% of qualifying responses, or 68 of 134), and Gemini 3.1 Pro and GPT-6 Astra sometimes recognize the shape of demand when the linear fit is inadequate.
3. Having established the limitations of basic LLMs on these standard instances, we turn our attention to the performance of advanced LLMs on instances with more complex demand functions, and market environments that exhibit seasonality in demand and competition. For more complex demand functions, advanced LLMs emphasize parameter estimation rather than learning more complex models, but differ in their use of post-hoc price adjustments. Gemini 3.1 Pro and GPT-6 Astra generally fit a linear demand model and choose its implied optimal price without further adjustment, while Claude Fable 5.1 continues to adjust the price after checking observed revenues. Interestingly, when the demand function has structural breaks (e.g., piecewise linear demand), advanced LLMs recognize the breakpoint and either price near the observed revenue peak or fit demand only above it, largely avoiding the regression heuristic's trap of recommending prices with zero or very low demand. Over piecewise demand functions, mean profit losses are 4.4–8.8% for advanced LLMs and 77.0% for the regression heuristic. In these settings, Claude Fable 5.1 and Gemini 3.1 Pro perform similarly to the best-observed-price heuristic (mean profit losses of 4.4% and 5.3%, compared with 4.0% for the heuristic; $p = 0.341$ and $p = 0.412$, respectively). Over instances which additionally include seasonality and/or competition, advanced LLMs rely more heavily on regression modeling and less on post-hoc price adjustments, incorporating competitor prices and seasonality into their models or restricting estimation to the relevant portion of the sample. Advanced models incur

mean profit losses of 2.4–3.5% under seasonality and 3.1–4.3% under competition, compared with 2.7% and 4.3%, respectively, for the regression heuristic.

4. Finally, we find evidence that minor prompt changes, including expert framing and demand-family hints, can improve the performance of some advanced LLMs. In particular, the use of expert role framing improves the performance of Gemini 3.1 Pro (including the role reduces mean profit loss by 0.4 percentage points, $p = 0.012$), which has the best overall performance in our benchmark. Furthermore, providing a demand-family hint can nudge models away from linear fits and improves the performance of both Gemini 3.1 Pro and GPT-6 Astra. Under the hint, the share of responses reporting a linear fit falls from 92% to 55% for Gemini 3.1 Pro and from 75% to 44% for GPT-6 Astra; mean profit loss falls by 0.6 and 0.8 percentage points, respectively ($p = 0.057$ and $p = 0.003$). These results suggest that human-in-the-loop interventions can improve on the pricing performance of advanced frontier models.

As LLM-powered pricing agents become increasingly common in the workplace, an essential operational question is not only whether they can make good decisions, but also when and where they may fail and require human-in-the-loop intervention (Yang et al. 2026a). Our paper addresses this question and finds that, while advanced LLMs are remarkably effective at pricing in many settings, they also exhibit several biases and failure modes that may reduce their performance in practice. We also find that model performance is highly heterogeneous, with different frontier models approaching complex pricing tasks in different ways, and also exhibiting different levels of sensitivity to how information is presented. This heterogeneity creates opportunities to improve performance through careful model selection, prompt design, and information curation. Our results therefore suggest a role for human oversight by managers who can correct, guide, choose and validate the models' pricing recommendations.

1.2. Literature Review

In the construction of benchmark and subsequent analysis, we primarily draw from and contribute to three streams of literature: pricing from demand data, LLMs as pricing agents and economic decision makers, and benchmarking LLMs at mathematical modeling and optimization tasks. In this subsection, we describe these streams and compare/contrast them with our work.

Pricing from demand data. A foundational question in revenue management and pricing is how sellers should set prices with incomplete information about demand (see den Boer (2015) for a survey). In particular, how should a seller use historical prices and the corresponding demand observations to choose a price for the next period? For the online version of this problem, a substantial literature develops policies that balance demand learning and revenue maximization (Besbes and Zeevi 2009, den Boer and Zwart 2014, Keskin and Zeevi 2014), including policies

based on simple, potentially misspecified demand models (Besbes and Zeevi 2015). Applied work also combines demand forecasting and price optimization for online retailers (Ferreira et al. 2016). For the offline version, related work studies pricing from limited historical purchase information (Allouah et al. 2023), and contextual demand estimation (Li et al. 2026), as well as pricing from inventory-censored sales (Bu et al. 2023, Tang et al. 2025). Unlike these papers, we do not propose new algorithms for learning to price from historical demand. Instead, we evaluate how effectively frontier LLMs, treated as black boxes, can utilize a finite history of demand observations and contextual information to recommend a price.

LLMs as pricing agents and economic decision makers. Beyond algorithmic approaches, an emerging literature studies LLMs as economic and operational decision makers. Most closely related to our work is Fish et al. (2025), which evaluates AI agents across a variety of economic tasks, including repeated pricing with feedback in unknown demand environments. Relative to this work, we focus exclusively on single-product pricing and systematically vary demand families, market environments, and prompt variants. Other work studies LLM-based demand estimation and pricing (Han et al. 2026), develops specialized LLM pricing systems (Ma et al. 2026), and examines the use and reliability of LLM-generated estimates of customer preferences and purchase behavior (Huang and Wang 2026, Wang et al. 2026, Oetzel and Maiberger 2026). Building on these papers, a related line of work examines market outcomes and the potential for collusion when LLM agents repeatedly compete on prices (Fish et al. 2024, Bichler et al. 2026, Giebel et al. 2026, Keppo et al. 2026). Unlike these works however, we only evaluate individual agents' pricing decisions, and primarily in monopoly settings.

There is also related work in supply chain and operations management examining LLM-based algorithm and inventory policy design (Baek 2026, Yang et al. 2026b). Baek et al. (2026) compare OR methods, LLM agents, and human-AI teams on inventory management tasks. We complement this literature by comparing one-period pricing recommendations from fixed histories and controlled prompt variants.

Benchmarks for mathematical modeling and optimization. Finally, our work draws on the broader literature on benchmarking LLMs' performance on quantitative tasks. Our benchmark design is informed by Reuel et al. (2024), who outline best practices for benchmarking LLM performance emphasizing reproducibility and statistical assessment. Related methodological work examines prompt sensitivity (Cohen and Hage-Youssef 2026, Zheng et al. 2024) and the faithfulness of AI generated explanations (Turpin et al. 2023), providing guidance for our prompt design and interpretation of model responses. Other AI benchmarks that we draw inspiration from include: NL4Opt who examine extracting optimization formulations from text (Ramamonjison et al. 2023),

ORLM's IndustryOR benchmark for industrial problems (Huang et al. 2025), OptiBench for end-to-end optimization (Yang et al. 2025), and PricingLogic which evaluates the application of overlapping fare rules (Liu et al. 2025). Our benchmark complements this literature by focusing on benchmarking AI performance at demand inference and price selection from limited observations.

1.3. Paper Outline

The remainder of the paper is organized as follows. In Section 2, we describe how we design pricing instances and the scope of our benchmark. In Section 3, we introduce our experiment setting and valuation methods. In Section 4, we evaluate the performance of six LLMs over our benchmark and provide statistical insights into their performance. Finally, in Section 5, we conclude and discuss avenues for future work. The full performance of all LLMs over the full benchmark of instances is publicly available at: https://enfalarisoy.github.io/aibenchmark/. All omitted details and additional empirics can be found in the Appendix.

2. Instance Design and Benchmark Construction

In this section, we construct a benchmark of single-product pricing instances. Each instance provides the seller with a finite history of prices and realized demand, together with relevant market information, and is governed by a hidden underlying demand function by which the histories are generated, the profit of candidate prices can be evaluated, and a true optimal price can be computed. The instances vary across demand functions, market environments, parameter sets, noise levels, and history lengths. We first define the pricing task in Section 2.1, and then, in Sections 2.2 to 2.4, we describe the demand families, market environments, and historical data variations across instances in the benchmark.

2.1. Single-Product Pricing Task

In this subsection, we describe the pricing task associated with each instance. Specifically, a seller chooses a price for a single product in the next period based on historical prices and realized demand. The seller does not know the underlying demand function and must use the available observations and market information to make its pricing decision. Each instance thus captures a fundamental challenge in data-driven pricing: historical sales provide information about customer responses to prices but generally do not identify the optimal price directly.

Formally, for each pricing instance i let $D_i(p, x)$ denote the expected demand at price p in market environment x. Given a feasible price set $\mathcal{P}_i$ (i.e., the support of the demand function) and the market condition x_i^+ for the next period, the full information optimal price satisfies:

$$p_i^* = \arg\max_{p \in \mathcal{P}_i} \Pi_i(p; x_i^+), \qquad \Pi_i(p; x_i^+) = pD_i(p, x_i^+), \tag{1}$$

where Π_i denotes the expected profit of instance i. Note, all demand families considered in our benchmark are such the optimal price p^* is the unique profit maximizer.

The seller does not observe the demand function $D_i(\cdot)$ or its parameters. Instead, each instance provides a history of prices and realized demand, $H_{i,T} = \{(p_{i,t}, D_{i,t}, x_{i,t})\}_{t=1}^{T}$, where $p_{i,t}$ denotes the posted price, $D_{i,t}$ is realized demand, and $x_{i,t}$ represents the relevant market condition in period t. The seller uses this information to choose a price, balancing price and sales volume while accounting for market conditions. The chosen price need not coincide with a historically observed price. The performance of chosen price is evaluated with respect to Eq. (1). Having defined a pricing instance, in the remainder of this section we will give a taxonomy of all pricing instances in our benchmark.

2.2. Demand Families

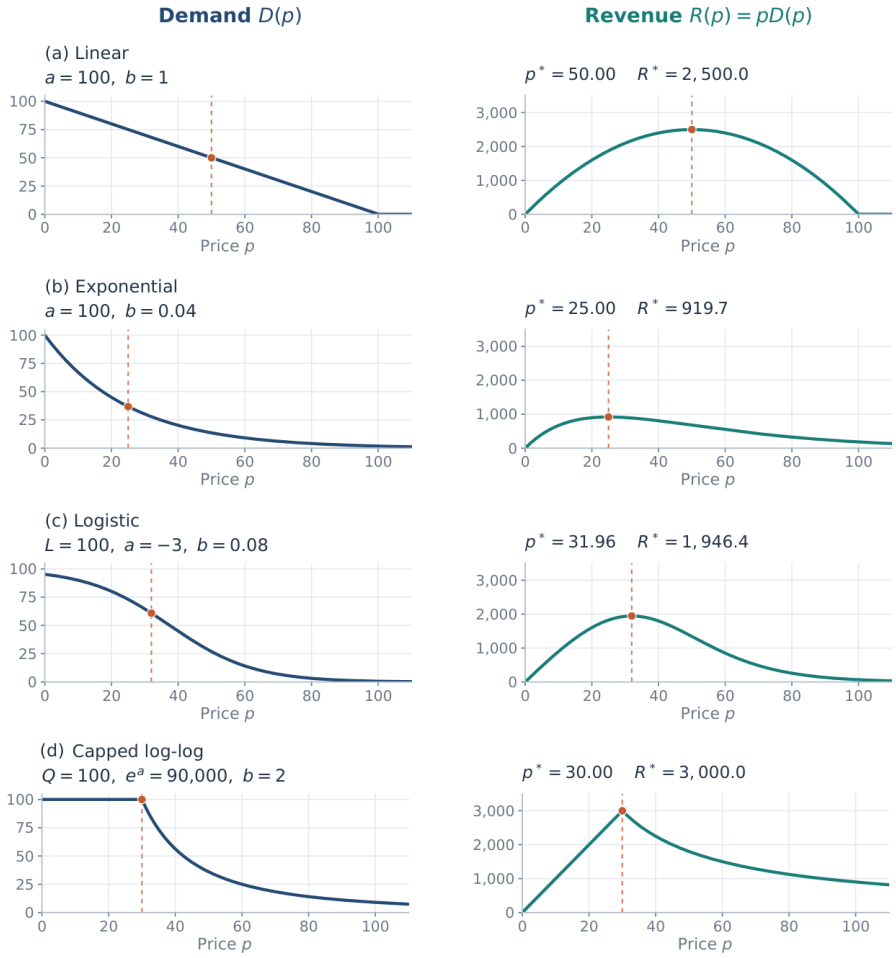
Each pricing instance is governed by a demand function with a specified parameter set, from which the histories are generated and the optimal price is computed. We refer to the collection of demand functions obtained by varying the parameters within a given functional form as a *demand family*. In our benchmark, we consider eight such demand families, ranging from standard parametric specifications to more complex relationships with structural breaks or heterogeneous customer responses. These different demand families generate substantially different revenue curves even when their demand patterns are similar over the observed price range. These families are summarized in Table 1 together with their optimal prices in a static single-seller market.

Table 1 Base demand families and the corresponding optimal prices in the static single-seller market.

Demand family	Expected demand function $D(p)$	Optimal price p^*
Linear	$[a - bp]^+$	$\frac{a}{2b}$
Exponential	ae^{-bp}	$\frac{1}{b}$
Logistic	$\frac{L}{1+e^{a+bp}}$	Unique solution to $bp^* = 1 + e^{-a-bp^*}$.
Weibull	$A\exp[-(p/\lambda)^k]$, $\lambda > 0$, $0 < k < 1$	$\lambda k^{-1/k}$
Piecewise linear	$\begin{cases}[a_1 - b_1 p]^+, & p < \bar{p}, \\ [a_2 - b_2 p]^+, & p \geq \bar{p}\end{cases}$	Compare valid branch candidates $a_j/(2b_j)$, $j = 1, 2$, with $\bar{p}$ and feasible boundaries.
Piecewise exponential	$\begin{cases}a_1 e^{-b_1 p}, & p < \bar{p}, \\ a_2 e^{-b_2 p}, & p \geq \bar{p}\end{cases}$	Compare valid branch candidates $1/b_j$, $j = 1, 2$, with $\bar{p}$ and feasible boundaries.
Capped log-log	$\min\{Q, e^a p^{-b}\}$, $b > 1$	$\left(\frac{e^a}{Q}\right)^{1/b}$
Mixed logit	$\frac{N}{K}\sum_{j=1}^{K} \frac{1}{1+e^{-(\alpha_j + \beta_j p)}}$, $K = 4$, $\beta_j < 0$	Global numerical maximization of $pD(p)$; multiple local maxima may occur.

Notes. Here, $[z]^+ = \max\{z, 0\}$, all scale parameters and b, b_1, b_2 are positive, and $D(0) = Q$ for capped log-log demand.

Figure 1 Demand and revenue curves of the basic demand families.

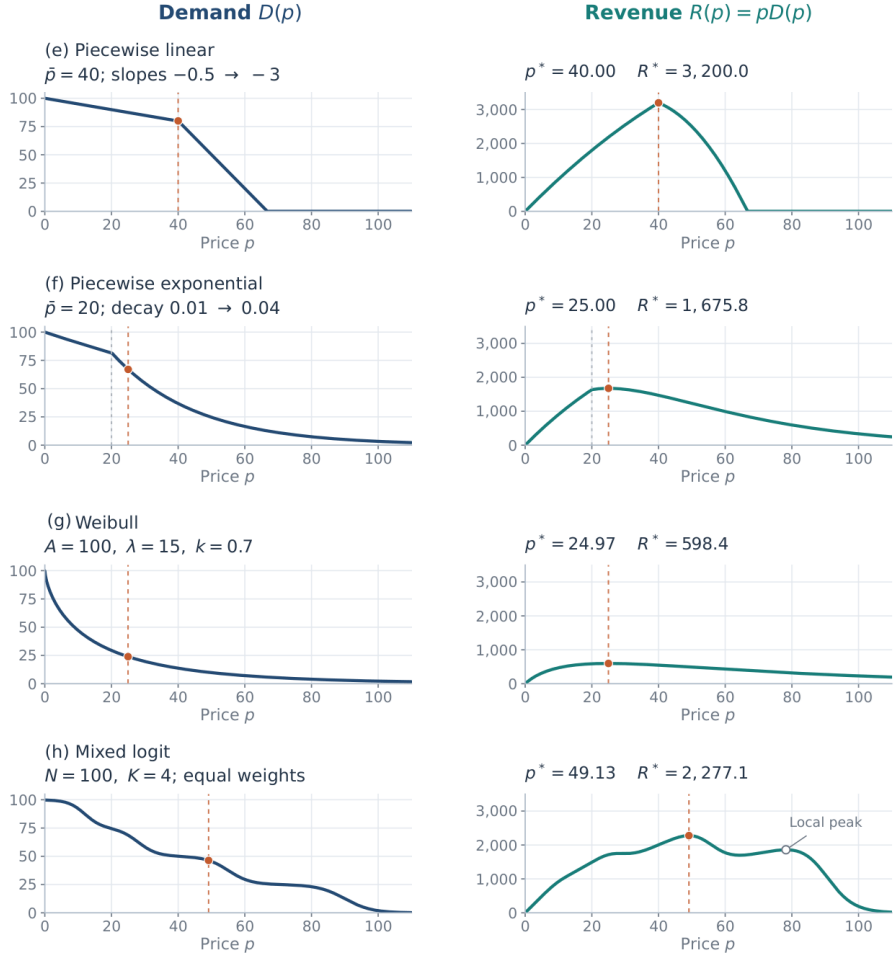


Note. Left panels show expected demand; right panels show expected revenue. Each row represents a different demand family, with parameters displayed in the panel. Orange markers and dashed vertical lines identify the global revenue-maximizing price.

We classify the eight demand families into basic and advanced groups. The basic families represent commonly used specifications in the pricing literature and consists of linear, exponential, logistic, and capped log-log demand. Note, the linear and exponential demand provide standard benchmarks in which demand decreases with price. The logistic family introduces an S-shaped demand relationship, while capped log-log demand remains constant over an initial price range before declining. These demand and associated revenue functions are depicted in Fig. 1.

The advanced families introduce features that may complicate demand inference and price optimization consists of Weibull, piecewise-linear, piecewise-exponential, and mixed-logit demand. Weibull demand models slow demand decay at high prices, while piecewise-linear and piecewise-exponential demand allow structural breaks. Finally, the mixed-logit demand introduces heterogeneity in price sensitivity through four equally weighted logistic components and is inspired by a

Figure 2 **Demand and revenue curves of the advanced demand families.**



Note. Left panels show expected demand; right panels show expected revenue. Each row represents a different demand family. Orange markers identify the global revenue-maximizing price; gray dotted lines mark structural breakpoints. The labeled local peak in the mixed-logit panel is distinct from the global maximum.

true price and demand relations reported in the field experiment of Dubé and Misra (2023) (for details of how we arrive at this demand function, see Appendix A.1). These demand and associated revenue functions are depicted in Fig. 2.

Having defined the eight demand families which we use to generate instances, we now describe the different market environments each instance encodes.

2.3. Market Environments

In each instance, a demand function from one of the eight families is embedded in one of three market environments: static, seasonal, and competitive. We write $D_{i,t}(p)$ for expected demand at price p in period t. In the *static* environment, a single seller faces a time-invariant expected demand function $D_{i,t}(p) = d_i(p)$ that depends only on its posted price p. The seller observes historical

prices and realized demand and chooses a price for the next period. The pricing problem therefore concerns demand inference and price optimization without additional market effects.

The *seasonal* environment introduces observable demand fluctuations over time, while the *competitive* environment allows demand to depend on a rival's price. We introduce seasonality by allowing the level of demand to vary with the day type in each period. Specifically, let $x_{i,t}$ denote the day type in period t. We distinguish between weekdays and weekends and assume that expected demand takes the form:

$$D_{i,t}(p) = D_i(p; x_{i,t}) = s_{x_{i,t}} d_i(p), \qquad x_{i,t} \in \{\text{weekday}, \text{weekend}\}, \qquad s_{x_{i,t}} > 0,$$

where $d_i(p)$ is the underlying demand function and $s_{x_{i,t}}$ represents the seasonal demand multiplier in period t. The history records the day type for each observation, and the seller knows the day type for the next period. Observed demand can vary with both price and day type, so the seller must distinguish these effects when interpreting the historical data. As the seasonal adjustment scales demand proportionally at every price, it changes expected demand and revenue but leaves the optimal price unchanged across periods:

$$\arg\max_{p \in \mathcal{P}_i} (p - c_i) D_{i,t}(p) = \arg\max_{p \in \mathcal{P}_i} (p - c_i) d_i(p).$$

Note, higher demand during a particular season therefore does not by itself justify a higher price when the seasonal effect is purely multiplicative.

Finally, in the *competitive* environment, we consider a market with two sellers offering similar products. In this environment, demand for the focal seller is governed by an embedded demand function and depends on both its own price and the competitor's price in each period. Specifically, given the focal seller's price p and the competitor's price $q_{i,t}$ in period t, its expected demand is:

$$D_{i,t}(p) = D_i(p; q_{i,t}) = \left[d_i(p) + \gamma_i (q_{i,t} - p) \right]^+, \qquad \gamma_i \geq 0.$$

The parameter γ_i captures the sensitivity of demand to the difference between the two sellers' prices. Holding the focal seller's price fixed, its expected demand is weakly increasing in the competitor's price.

The focal seller observes its historical prices and realized demand, together with the competitor's historical prices. The competitor's next-period price, denoted by $q_i = q_{i,T+1}$, is disclosed before the focal seller chooses its own price. The corresponding full-information optimal price satisfies

$$p_i^*(q_i) \in \arg\max_{p \in \mathcal{P}_i} (p - c_i) \left[d_i(p) + \gamma_i (q_i - p) \right]^+.$$

To construct the competitive instances, we use the same base-demand parameters as in the corresponding static instances and calibrate the competitive effect relative to demand sensitivity at the

single-seller optimum. Specifically, when the base demand function is differentiable at its single-seller optimal price $p^*_{i,0}$, we set $\gamma_i = 0.2\left|d_i'(p^*_{i,0})\right|$. Thus, the competitive effect is calibrated to 20% of the base-demand sensitivity at the single-seller optimum. For demand functions that are not differentiable at this point, the calibration uses a consistent one-sided or numerical sensitivity. Note, the competitive effect can change the optimal price, creating a one-period best-response problem in which the seller must account for the competitor's disclosed price.

Finally, having defined the demand families and market environments, we complete our taxonomy of pricing instances in our benchmark by describing the different ways histories are generated in each instance.

2.4. Historical Data and Instance Design

For each instance, given a demand function and market environment, we generate a finite history of prices and realized demand to represent the information available to the seller. Historical prices are randomly selected to mimic previous price experimentation (in the competitive environment, both sellers' historical prices are randomized), and we vary the number of observations, which we refer to as the *length* of the history. We also apply idiosyncratic noise in the demand observed at each price and vary the *level* of noise which corresponds to how large the noise is relative to the demand observation. Each instance is thus further specified by the history length and noise level used to generate the fixed dataset provided to the seller.

Specifically, we vary the history length to study how the amount of available information affects pricing performance. For the static and competitive environments, we consider history lengths of $T \in \{5, 15, 45\}$. These choices correspond to short, medium, and long histories. For the seasonal environment, we use $T \in \{7, 21, 63\}$, so that each history contains complete weeks with a weekday-to-weekend ratio of $5:2$. Note, seasonal histories contain more observations than the corresponding static and competitive histories, which should be considered when interpreting comparisons across environments.

We also vary the noise level to study how demand uncertainty affects pricing performance. Given a posted price $p_{i,t}$ and market condition $x_{i,t}$, expected demand in period t is $D_{i,t}(p_{i,t}) = D_i(p_{i,t}; x_{i,t})$ where $x_{i,t}$ denotes the day type in the seasonal environment or the competitor's price $q_{i,t}$ in the competitive environment (no additional market condition is needed in the static environment). We generate realized demand in the histories according to:

$$D_{i,t} = D_{i,t}(p_{i,t})(1 + \varepsilon_{i,t}), \qquad \varepsilon_{i,t} \overset{\text{i.i.d.}}{\sim} \text{Uniform}(-\sigma_i, \sigma_i), \qquad \sigma_i \in \{0.1, 0.3\}, \tag{2}$$

where, σ_i corresponds to one of two noise levels allow realized demand to deviate from its conditional expectation by up to 10% and 30%, respectively. Since $\sigma_i < 1$, realized demand remains nonnegative, and $\mathbb{E}[D_{i,t} \mid p_{i,t}, x_{i,t}] = D_{i,t}(p_{i,t})$. The standard deviation of the multiplicative noise is $\sigma_i/\sqrt{3}$.

In summary, each instance represents a next-period pricing task for a single product, defined by a demand family and parameter set, a market environment, a history length, and a noise level. For each of the eight demand families, we construct three parameter sets, labeled a, b, and c (Table A.1), which vary the location of the optimal price and the shape of the revenue function. Combining these with the three market environments, three history lengths, and two noise levels yields $8 \times 3 \times 3 \times 3 \times 2 = 432$ distinct pricing instances, with 144 instances in each market environment. Within each instance, the historical dataset and next-period market conditions remain fixed across methods and prompt variants, enabling comparisons on the same numerical pricing problem.

In the next section, we use this benchmark to design a large-scale experiment, presenting each instance to a set of LLMs as a prompt requesting a next-period price and evaluating the expected revenue of the LLM-powered recommendations relative to the true optimum.

3. Experimental Setup for LLM Pricing Evaluation

Having defined the benchmark instances, in this section we describe how we will use our benchmark of instances to evaluate LLM pricing performance. In Section 3.1, we specify the models we give details for their run protocols. In Section 3.2, we describe the prompts by which we supply instances to LLMs. Finally, in Sections 3.3 and 3.4, we define the metrics used to score LLMs recommended pricing decisions and our heuristic baselines with which we compare them.

3.1. Models and Run Protocol

In our experiment, we evaluate six LLMs in two capability tiers. The basic tier includes Claude Haiku 4.5, GPT-5 mini, and Gemini 2.5 Flash; the advanced tier includes Claude Fable 5.1, GPT-6 Astra, and Gemini 3.1 Pro. The LLMs are given *prompts*, which an encoding of a pricing instance written with some prompt structure (in Section 3.2 we describe the prompt structure and variants). We organize the evaluation around the 72 baseline prompts for the four basic demand families in a static single-seller market (Section 4.1). Subsequent analyses examine changes in the market environment (Section 4.2, 144 prompts), the prompt variant (Section 4.4, 216 prompts), and the demand family (Section 4.3, 72 prompts). In total, each model is evaluated on 504 distinct prompts covering 288 unique instances, as summarized in Table 2. For each prompt query, we set the temperature to 0 where supported and disable reasoning or request the lowest supported reasoning level. Because providers differ in the settings they permit, we cannot impose identical decoding and reasoning settings across all models. The settings used for each model are reported in Table C.1.

To assess the stability of model performance, we query each model multiple times with each prompt. Each model output is a *response*, and repeated runs of the same prompt are *replications*. Due to the higher cost of query advanced models, we replicate the basic models five times per prompt and the advanced models only twice. This yields 2,520 and 1,008 responses per basic

Table 2 Evaluation scope by section.

				Responses per model	
Section	**Environment**	**Instances**	**Prompts**	**Basic**	**Advanced**
§4.1 Static single-seller market	Static	72	72	360	144
§4.2 Seasonality or competition	Seasonal, competitive	144	144	720	288
§4.4 Prompt variants	Static	72[a]	216	1,080	432
§4.3 Additional demand families	Static	72	72	360	144
Total		288	504	2,520	1,008

Note. Sections 4.2 to 4.3 each vary one aspect of the baseline setting in Section 4.1. A prompt is an instance written under one prompt variant, and a response is a model's output to a prompt. Basic models receive five replications per prompt and advanced models two. [a]The same instances as in Section 4.1; they are counted only once in the total.

and advanced model, respectively. For the prompts analyzed in Sections 4.1 to 4.4, we find no statistically significant differences in performance across replications for any model (for details, see Table C.2 in the Appendix).

3.2. Prompt Construction

We present each pricing instance to an LLM through a structured prompt describing the market environment and seller's objective, historical prices and realized demand, zero marginal cost, and relevant market conditions for the next period. The prompt also states that realized demand contains random variation. We instruct the LLM to recommend a profit-maximizing price and provide a brief explanation of its reasoning. To facilitate consistent evaluation, we request the LLM return a JSON object containing two fields: a numerical price and a reasoning field with fewer than 20 sentences. We use the price to evaluate economic performance and the explanation to examine the model's stated approach to the pricing problem.

We also investigate whether the way an instance is presented affects the resulting recommendation. We vary three prompt components: role framing, product context, and a distribution hint. These variations leave the demand function, historical observations, marginal cost, market conditions, and pricing objective unchanged, allowing us to assess their effects on the same numerical pricing problem.

Role framing (`off`/`on`). We first examine whether assigning an explicit expert role affects pricing decisions. Under `on`, the prompt begins with the instruction *You are a pricing expert*; under `off`, this instruction is omitted. This variant changes the role assigned to the LLM without providing additional observations or information about the underlying demand function.

Product context (`none`/`luxury`). We next examine whether qualitative product information influences pricing decisions. Under `luxury`, the prompt describes a premium product with distinctive

positioning and associated services. Under `none`, this description is omitted, leaving only the information needed for the numerical pricing task. Although premium positioning may convey information about customer preferences or price sensitivity in practice, the narrative is deliberately separated from the underlying demand relationship in our benchmark. This treatment therefore tests whether models adjust prices in response to product context when the numerical pricing problem is unchanged.

Distribution hint (`none`/`auto`). Finally, we examine how additional information about demand structure or market conditions affects pricing decisions. Under `auto`, the prompt-construction framework selects a hint according to the demand family and market environment; under `none`, it omits this information. For static instances with linear, exponential, logistic, or capped log-log demand, the hint lists these four functional forms as possible demand models without revealing which one generates the observations. For piecewise demand functions, it indicates that demand may contain a structural break. For Weibull demand, it highlights the possibility of substantial demand at relatively high prices. For empirically calibrated mixed-logit instances, it states that a pricing expert has estimated the demand curve from data.

The framework also supports hints tailored to the market environment. In seasonal instances, the hint draws attention to systematic differences between weekdays and weekends. In competitive instances, it describes how the competitor's price affects the focal seller's demand. The detailed prompt components and disclosures appear in Section B. Because these hints differ in the information they provide—possible demand forms, market conditions, or the source of demand estimates—their effects should be interpreted in relation to the corresponding disclosure.

For the reported prompt-variation experiment, we use a one-factor-at-a-time design on the 72 static instances with the four basic demand families. The baseline prompt sets role framing to `on`, product context to `none`, and the distribution hint to `none`. We construct three alternative variants, each changing one component: removing role framing, adding the luxury product description, or providing the distribution hint. In this experiment, the hint lists the four possible demand families without identifying the true one. Applying these variants to the same 72 instances yields 216 additional prompts, which, together with the baseline prompts for all 288 instances, give the 504 prompts summarized in Table 2. This design allows us to compare prompt components while holding the numerical pricing problem fixed; the results are presented in Section 4.4.

3.3. Evaluation Metrics

After collecting the models' responses to these prompts, we evaluate the quality of their pricing recommendations. Pricing quality can be assessed along several dimensions, including proximity to the optimal price and the absolute or relative revenue generated. We focus on expected revenue relative to the optimum and define *relative profit loss* as follows.

Definition 1 (Relative Profit Loss). Let p_i^* be the optimal price in Eq. (1), and write $\Pi_i(p)$ for the expected profit $\Pi_i(p; x_i^+)$. The relative profit loss of a recommended price p is

$$L_i(p) = 1 - \frac{\Pi_i(p)}{\Pi_i(p_i^*)}.$$

□

We score each LLM response by the loss associated with its recommended price and assign a loss of 1 to responses without a valid price. We use this measure to compare the LLMs with one another and with two heuristic baselines, described next.

3.4. Heuristic Baselines

We compare LLM performance with two simple, implementable heuristics that use the same historical observations. The first is the *best-observed-price* heuristic, which recommends the historical price that generated the highest observed revenue. This heuristic neither fits a demand model nor adjusts for contextual or market conditions. Comparisons with this heuristic assess whether LLMs can improve on simply repeating the most successful historical price.

The second is the *regression* heuristic, which fits a linear demand model to the historical data by ordinary least squares (OLS) and maximizes revenue under the fitted model, following the demand-estimation and price-optimization approach studied in the pricing literature (Keskin and Zeevi 2014, Besbes and Zeevi 2015). The model includes a weekend indicator in seasonal instances and the competitor's price in competitive instances. These terms are evaluated at the next period's day type and disclosed competitor price, respectively. Writing the resulting fitted demand as $\widehat{D}(p) = a + bp$, where a incorporates these covariate effects, the heuristic recommends $\hat{p} = -a/(2b)$ when $b < 0$. If $b \geq 0$ or the implied price is non-positive, we assign the heuristic a relative profit loss of 100% in that instance. This heuristic provides a baseline for simple model fitting, imposing a linear relationship between price and demand across all demand families. Improvements over this baseline may come from selecting a more suitable demand model or adjusting the price implied by the fitted model.

With the experimental design, evaluation metric, and heuristic baselines in place, we are now ready to evaluate LLM pricing performance across the benchmark.

4. How Well Do LLMs Price?

In this section, we evaluate six LLMs, three basic and three advanced, and compare their pricing performance against one another and against two benchmarks: the best-observed-price heuristic and the regression heuristic. Section 4.1 studies baseline pricing problems in which a single seller faces one of four standard demand families. Section 4.2 adds two sources of market complexity: seasonality, where demand varies between weekdays and weekends, and competition, where the seller must account for a rival's price. Section 4.3 extends the analysis to four additional demand families. Finally, Section 4.4 examines how prompt design affects pricing performance.

Table 3 Static Single-Seller Market: Relative Profit Loss of LLMs and Benchmarks

	Relative profit loss (%)				Share of responses (%)	
	Mean	Median	Min	Max	Near optimal	Severe loss
Basic LLMs						
Claude Haiku 4.5	9.10	4.63	0.00	58.52	54.4	8.6
Gemini 2.5 Flash	6.62	2.12	0.01	48.51	69.7	7.2
GPT-5 mini	10.61	6.19	0.34	45.92	55.0	13.1
Advanced LLMs						
Claude Fable 5.1	2.60	0.51	0.00	17.50	85.4	0.7
Gemini 3.1 Pro	3.51	0.25	0.00	46.76	83.3	2.8
GPT-6 Astra	3.69	0.22	0.00	46.77	79.9	2.8
Benchmarks						
Best-observed price	5.49	2.76	0.00	35.30	73.6	2.8
Regression heuristic	4.31	0.23	0.00	46.76	75.0	2.8

Note. Responses that return no valid price receive a loss of 100%; this occurred once, for Gemini 2.5 Flash.

4.1. Pricing in a Static Single-Seller Market

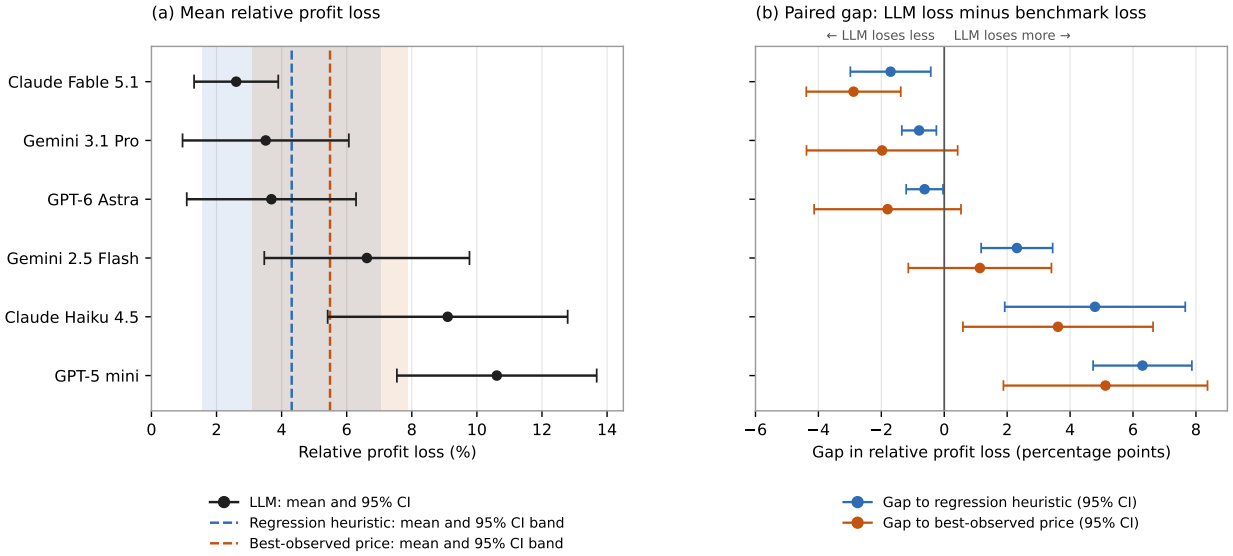
We first consider a single seller facing a static price-demand relationship. Table 3 summarizes performance across 72 instances, combining four demand families (linear, exponential, logistic, and capped log-log), three parameter sets, three history lengths (5, 15, and 45 price-demand observations), and two noise levels ($\sigma = 0.1$ and $\sigma = 0.3$). Using the baseline prompt, we query each basic LLM five times per instance and each advanced LLM twice. For each LLM, we first average losses across replications of each instance and then report the mean, median, minimum, and maximum across instances. Near-optimal and severe-loss rates are calculated across individual LLM responses and across the 72 instances for each heuristic.

On average, basic LLMs lose more profit than either heuristic, while advanced LLMs lose less. Mean losses range from 6.6 to 10.6% for basic LLMs and 2.6 to 3.7% for advanced LLMs, compared with 5.5% for the best-observed-price heuristic and 4.3% for the regression heuristic. The shares of near-optimal responses show a parallel pattern: 54.4–69.7% for basic LLMs and 79.9–85.4% for advanced LLMs, compared with 73.6–75.0% for the heuristics. Severe losses are also more frequent among basic LLMs (7.2–13.1% of responses) than either heuristic (2.8%). Advanced LLMs rarely produce severe losses (0.7–2.8% of responses), with rates at or below those of the heuristics.

We next compare each LLM with the two heuristics and test whether the differences in mean loss are statistically significant. Figure 3(a) plots mean losses with 95% confidence intervals, using 360 responses per basic LLM, 144 per advanced LLM, and 72 observations per heuristic. Standard errors are clustered by price history (36 clusters), and intervals use a t-distribution with 35 degrees of freedom.[1] For the paired comparisons, we subtract each heuristic's loss from the LLM's loss in

[1] The two noise-level variants of each history share the same posted prices but differ in realized demand. We group these variants and all replications in the same cluster.

Figure 3 Static Single-Seller Market: Mean Profit Loss and Paired Comparisons with Benchmarks



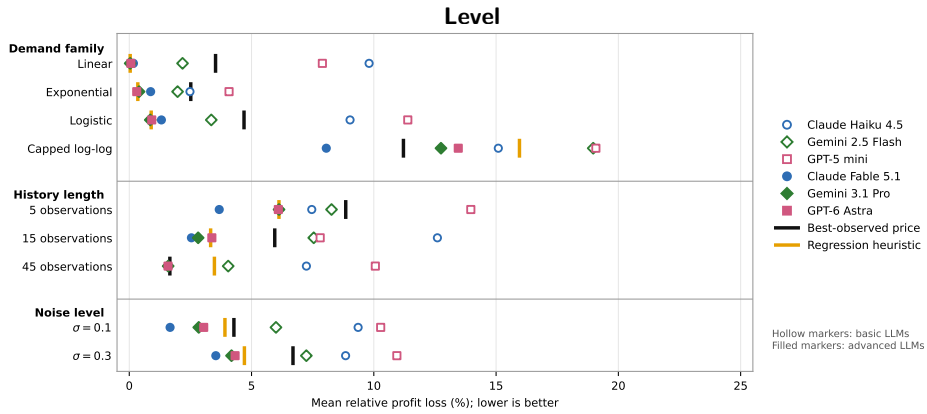
Note. In panel (a), black points and horizontal bars show LLMs' mean profit losses and 95% confidence intervals. Dashed vertical lines and shaded bands show the corresponding means and intervals for the regression heuristic (blue) and best-observed-price heuristic (orange). In panel (b), colored points and horizontal bars show mean paired loss differences and 95% confidence intervals. Negative differences favor the LLM; positive differences favor the heuristic.

the same instance for every response. We regress these differences on indicators for demand-family-by-parameter-set combinations, history length, and noise level. Each indicator is centered so that the intercept equals the mean paired difference, and standard errors are again clustered by price history. Figure 3(b) reports these estimates and their 95% confidence intervals.

The paired comparisons show a clear difference between basic and advanced LLMs. All three basic LLMs underperform the regression heuristic, with mean losses higher by 2.3 percentage points (pp) for Gemini 2.5 Flash ($p < 0.001$), 4.8 pp for Claude Haiku 4.5 ($p = 0.002$), and 6.3 pp for GPT-5 mini ($p < 0.001$). All three advanced LLMs outperform this heuristic, reducing mean loss by 1.7 pp for Claude Fable 5.1 ($p = 0.010$), 0.8 pp for Gemini 3.1 Pro ($p = 0.006$), and 0.6 pp for GPT-6 Astra ($p = 0.037$). Against the best-observed-price heuristic, Claude Haiku 4.5 and GPT-5 mini perform significantly worse, while only Claude Fable 5.1 performs significantly better; the remaining differences are not statistically significant at the 5% level.

Demand family, history length, and noise. We next examine how performance varies across demand families and with the number and noise level of historical observations. Figure 4 breaks down mean relative profit loss by demand family, history length, and noise level. Among advanced LLMs, differences in mean loss are small under linear, exponential, and logistic demand. The largest gaps arise under capped log-log demand, where Claude Fable 5.1's mean loss is 4.7 pp lower than Gemini 3.1 Pro's and 5.4 pp lower than GPT-6 Astra's. Neither difference is statistically

Figure 4 **Static Single-Seller Market: Mean Relative Profit Loss by Demand Family, History Length, and Noise Level**



Note. Colors identify models and heuristics. Hollow markers represent basic LLMs; filled markers represent advanced LLMs. Marker positions show mean relative profit loss within each subgroup; values farther left indicate lower loss. Each demand-family subgroup contains 18 instances; each history-length subgroup contains 24; each noise-level subgroup contains 36.

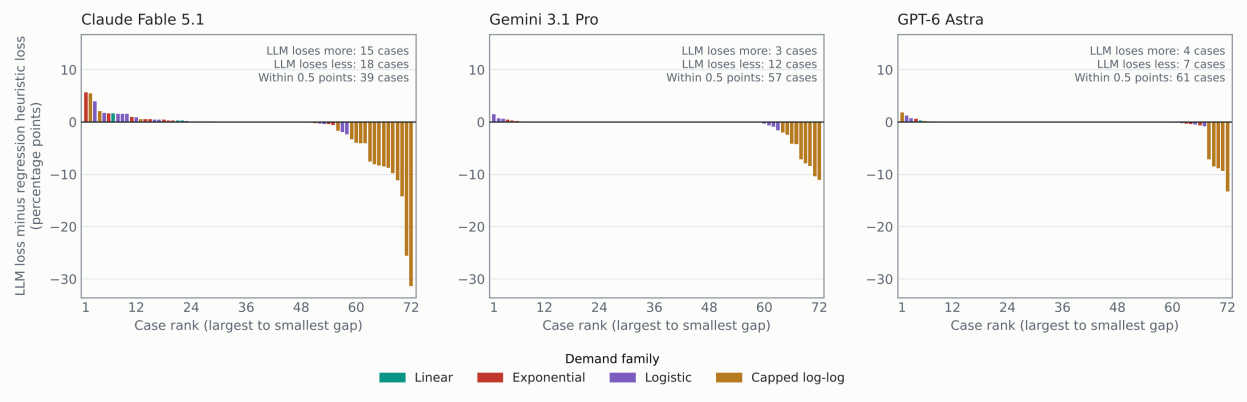
significant ($p = 0.165$ and $p = 0.104$, respectively), and we find no significant pairwise differences among advanced LLMs in any subgroup. However, Claude Fable 5.1 outperforms the regression heuristic under capped log-log demand, reducing mean loss by 7.9 pp ($p = 0.023$).

To examine how history length and noise affect performance, we estimate a separate instance-level regression for each LLM, with relative profit loss as the outcome and indicators for demand family, parameter set, history length, and noise level (Table D.1). Longer histories and lower noise should help models make better pricing decisions, but the estimated effects for basic LLMs show no consistent pattern. Advanced LLMs respond as expected: increasing history length from 5 to 45 observations reduces their losses by 2.1–4.5 pp (all $p < 0.10$), while increasing noise from $\sigma = 0.1$ to $\sigma = 0.3$ raises losses by 1.3–1.9 pp (all $p < 0.10$; $p < 0.05$ for Claude Fable 5.1 and Gemini 3.1 Pro).

Demand estimation and price selection. To understand these performance differences, we examine how the models estimate demand and select prices in their stated reasoning. Both basic and advanced LLMs frequently provide enough numerical information to identify the intercept and slope of a linear demand curve: this occurs in 78–100% of responses for basic LLMs and 75–93% for advanced LLMs. Among the 947 qualifying responses from basic LLMs, the mean absolute error in the reported slopes relative to the regression heuristic's estimates is 22.1%. Interestingly, 77% of the reported slopes are flatter than the regression heuristic's slopes. Among the 374 qualifying responses from advanced LLMs, this error is 5.3%.

Gemini 3.1 Pro reports a qualifying line in 92% of its responses (132 of 144). Among these responses, both coefficients are within 2% of the regression heuristic's estimates in 64% of responses (85 of 132), and the chosen price is within 2% of the optimum implied by the reported coefficients in 96% (127 of 132).

Figure 5 Advanced LLMs versus the Regression Heuristic: Profit Loss Differences Across Instances



Note. Each bar represents an instance, and colors identify demand families. Bar height is the LLM's profit loss minus the regression heuristic's loss, in percentage points; negative values favor the LLM. Instances are ranked separately within each model's panel, so horizontal positions need not represent the same instance across models.

GPT-6 Astra reports a qualifying line in 75% of its responses (108 of 144). Among these responses, both coefficients are within 2% of the regression heuristic's estimates in 87% of responses (94 of 108), and the chosen price is within 2% of the implied optimum in every case (108 of 108). Among the remaining 36 responses, it explicitly uses an exponential demand model in 18 and a logistic or S-shaped approximation in three, matching the true demand family. Another seven describe a low-price plateau followed by declining demand, consistent with the shape of capped log-log demand, although they do not specify a fitted function.

Claude Fable 5.1 refers to revenue observed at past prices in 133 of 144 responses and reports a qualifying line in 93% of responses (134 of 144). Among these responses, both coefficients are within 2% of the regression heuristic's estimates in only 19% of responses (26 of 134), while the chosen price departs from the implied optimum by more than 2% in 51% (68 of 134). In each of these 68 responses, the explanation describes checking observed revenues before selecting the final price. The remaining 10 responses all occur under logistic or capped log-log demand with long histories. In these responses, Claude Fable 5.1 recognizes the nonlinear shape correctly in eight.

We next compare each advanced LLM's loss with that of the regression heuristic across the 72 instances (Figure 5). Gemini 3.1 Pro and GPT-6 Astra have losses within 0.5 pp of the heuristic's in 79% (57 of 72) and 85% (61 of 72) of instances, respectively, compared with 54% (39 of 72) for Claude Fable 5.1. Claude Fable 5.1 differs from the heuristic more often in both directions, but its improvements are substantially larger than its losses. Its largest improvements, reaching 31 pp, occur under capped log-log demand. This family also accounts for the largest improvements by Gemini 3.1 Pro and GPT-6 Astra, although their gains are smaller.

Table 4 Seasonality and Duopoly: Relative Profit Loss of LLMs and Benchmarks

	Seasonality			Duopoly		
	Mean loss	**Near optimal**	**Severe loss**	**Mean loss**	**Near optimal**	**Severe loss**
Basic LLMs						
Claude Haiku 4.5	5.73	66.4	1.7	15.25	26.9	20.6
Gemini 2.5 Flash	5.10	70.0	4.7	18.00	31.4	25.3
GPT-5 mini	10.15	52.2	14.4	18.09	35.3	25.8
Advanced LLMs						
Claude Fable 5.1	2.38	85.4	0.7	3.57	73.6	0.7
Gemini 3.1 Pro	3.46	84.0	2.8	3.14	77.8	0.7
GPT-6 Astra	2.79	80.6	0.7	4.31	81.2	4.2
Benchmarks						
Best-observed price	6.52	66.7	6.9	6.13	55.6	0.0
Regression heuristic	2.69	77.8	0.0	4.25	81.9	4.2

Note. Responses that return no valid price receive a loss of 100%; this occurred once, for Gemini 2.5 Flash in the duopoly setting.

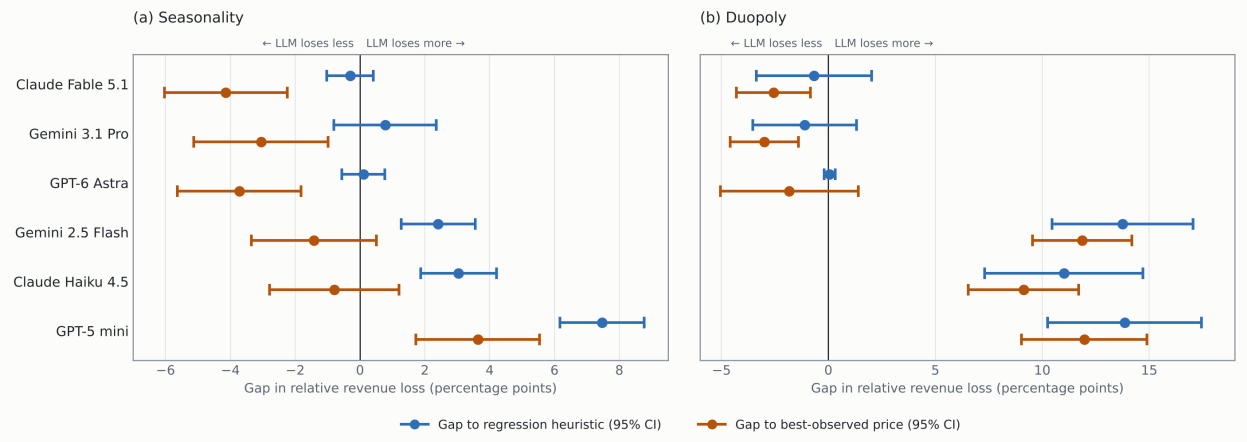
4.2. Pricing with Seasonality or Competition

We next add seasonality and competition to the static setting of Section 4.1. In the seasonal environment, demand varies between weekdays and weekends, while the profit-maximizing price remains unchanged. In the duopoly environment, demand depends on a competitor's price, and the agent must recommend a price given the competitor's disclosed price for the next period. We retain the four demand families, three parameter sets, two noise levels, and three history lengths, yielding 72 instances per environment and 144 instances in total. Seasonal histories contain 7, 21, or 63 observations to cover complete weeks, while duopoly histories contain 5, 15, or 45 observations. Using the same approach as in Section 4.1, we summarize performance in Table 4 and report comparisons with the two heuristics in Figure 6.

Under seasonality, basic LLMs have mean losses of 5.1–10.2%, compared with 2.4–3.5% for advanced LLMs, 6.5% for the best-observed-price heuristic, and 2.7% for the regression heuristic. All three basic LLMs significantly underperform the regression heuristic, with mean losses higher by 2.4 pp for Gemini 2.5 Flash, 3.0 pp for Claude Haiku 4.5, and 7.5 pp for GPT-5 mini (all $p < 0.001$). None of the advanced LLMs differs significantly from the regression heuristic. Against the best-observed-price heuristic, all three advanced LLMs perform significantly better, while only GPT-5 mini performs significantly worse.

Under competition, the performance gap between basic and advanced LLMs is larger. Basic LLMs incur mean losses of 15.3–18.1%, compared with 3.1–4.3% for advanced LLMs, 6.1% for the best-observed-price heuristic, and 4.3% for the regression heuristic. Severe losses occur in 20.6–25.8% of basic-LLM responses, compared with 0.7–4.2% of advanced-LLM responses. All three basic LLMs significantly underperform the regression heuristic, with mean losses higher by 11.0 pp for Claude

Figure 6 Seasonality or Competition: Paired Comparisons with Benchmarks



Note. Panel (a) shows seasonality and panel (b) duopoly. Points show mean paired differences in profit loss relative to the regression heuristic (blue) and best-observed-price heuristic (orange); horizontal bars show 95% confidence intervals. The vertical line marks zero difference. Negative values favor the LLM; positive values favor the heuristic.

Haiku 4.5 and 13.8 pp for both Gemini 2.5 Flash and GPT-5 mini (all $p < 0.001$). None of the advanced LLMs differs significantly from the regression heuristic. Against the best-observed-price heuristic, all three basic LLMs perform significantly worse, while Claude Fable 5.1 and Gemini 3.1 Pro perform significantly better; the difference for GPT-6 Astra is not statistically significant.

4.3. Additional Demand Families

We next consider four additional demand families: Weibull, empirically calibrated mixed logit, piecewise linear, and piecewise exponential. We use the same instance design and baseline prompt as in Section 4.1, yielding 72 instances in total. Using the same approach, we summarize performance in Table 5.

Weibull and mixed-logit demand. Across these two families, the regression heuristic has a mean loss of 0.3%, compared with 1.4% for the best-observed-price heuristic. Advanced LLMs have mean losses of 0.4–1.0% and produce near-optimal prices in 97.2–100%. Claude Fable 5.1 has a mean loss 0.6 pp higher than the regression heuristic ($p < 0.001$), whereas Gemini 3.1 Pro and GPT-6 Astra perform similarly to it. Against the best-observed-price heuristic, all three advanced LLMs perform significantly better, while Claude Haiku 4.5 and GPT-5 mini perform significantly worse; the difference for Gemini 2.5 Flash is not statistically significant.

Piecewise demand. Losses are larger under piecewise demand, particularly for the regression heuristic. Its fitted model recommends prices at which true demand is zero in 20 of the 36 instances, resulting in a mean loss of 77.0%. We therefore focus on comparisons with the best-observed-price heuristic, which has a mean loss of 4.0%. Advanced LLMs have mean losses of 4.4–8.8%, compared with 9.3–18.5% for basic LLMs. Claude Fable 5.1 and Gemini 3.1 Pro do not differ significantly

Table 5 Weibull, Mixed Logit, and Piecewise Demand: Relative Profit Loss of LLMs and Benchmarks

	Weibull and mixed logit			Piecewise linear and exponential		
	Mean loss	**Near optimal**	**Severe loss**	**Mean loss**	**Near optimal**	**Severe loss**
Basic LLMs						
Claude Haiku 4.5	3.87	77.8	0.6	9.28	61.1	13.9
Gemini 2.5 Flash	1.19	98.3	0.6	15.88	63.3	17.2
GPT-5 mini	2.93	85.0	1.1	18.54	48.9	21.1
Advanced LLMs						
Claude Fable 5.1	0.95	97.2	0.0	4.42	84.7	11.1
Gemini 3.1 Pro	0.39	100.0	0.0	5.25	77.8	5.6
GPT-6 Astra	0.39	100.0	0.0	8.83	69.4	9.7
Benchmarks						
Best-observed price	1.39	94.4	0.0	3.97	80.6	0.0
Regression heuristic	0.34	100.0	0.0	77.02	2.8	88.9

Note. Responses whose price falls in the zero-demand region receive a loss of 100%; this occurred in 14 responses, all under piecewise demand (GPT-5 mini 5, Gemini 2.5 Flash 8, GPT-6 Astra 1). The regression heuristic selects a price with zero demand, resulting in a loss of 100%, in 20 of the 36 piecewise instances.

from the best-observed-price heuristic ($p = 0.341$ and $p = 0.412$, respectively), while GPT-6 Astra ($p = 0.009$) and all three basic LLMs (all $p \leq 0.001$) perform significantly worse.

4.4. Prompt Variants

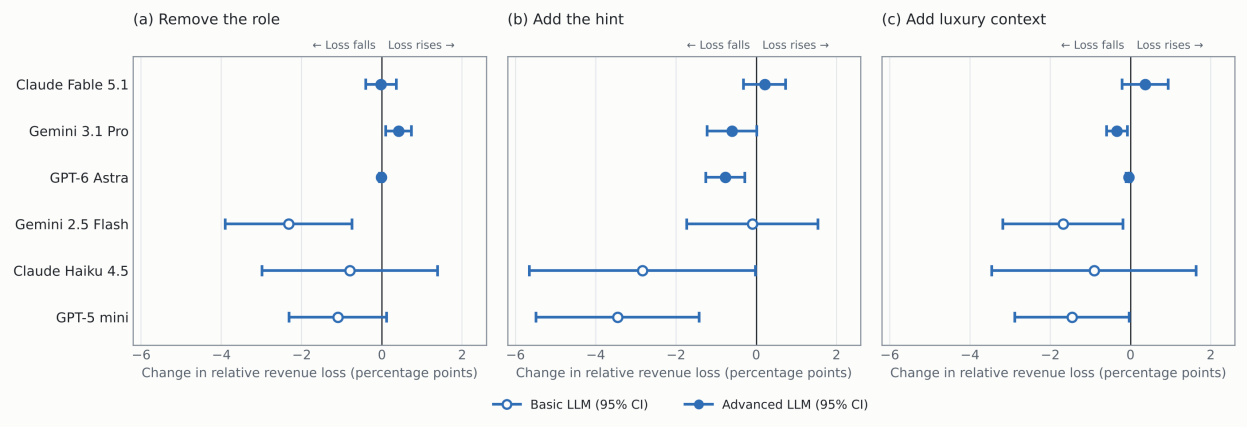
We finally examine how prompt design affects pricing performance. We consider three changes: removing the expert role, providing information about possible demand families, and describing the product as a luxury item. The underlying demand function, historical observations, and pricing objective remain unchanged. Figure 7 reports the estimated changes in relative profit loss, with 95% confidence intervals.

Removing the expert role. The baseline prompt assigns the model the role of a pricing expert. Removing this instruction has different effects across models. Gemini 2.5 Flash improves, with mean loss declining by 2.1 pp ($p = 0.001$), while Gemini 3.1 Pro performs worse, with mean loss increasing by 0.4 pp ($p = 0.012$). The changes for the other four models are not statistically significant.

Providing a demand-family hint. The hint lists linear, exponential, logistic, and capped log-log demand as possible demand families without identifying which family generated the observations. The hint improves performance for four of the six models. Mean loss declines by 2.2 pp for Claude Haiku 4.5 ($p = 0.001$), 1.4 pp for GPT-5 mini ($p = 0.044$), and 0.8 pp for GPT-6 Astra ($p = 0.003$). Gemini 3.1 Pro also improves by 0.6 pp, with the reduction significant at the 10% level ($p = 0.057$). The changes for Claude Fable 5.1 and Gemini 2.5 Flash are not statistically significant.

Luxury product. We describe the product as a luxury item while keeping the numerical pricing problem unchanged. This framing reduces mean loss by 1.5 pp for Gemini 2.5 Flash ($p = 0.011$),

Figure 7 Prompt Variants: Changes in Profit Loss Relative to the Baseline Prompt



Note. Points show changes in mean profit loss relative to the baseline prompt, in percentage points; horizontal bars show 95% confidence intervals. Hollow markers represent basic LLMs; filled markers represent advanced LLMs. The vertical line marks zero change. Negative values indicate lower loss under the prompt variant.

1.1 pp for GPT-5 mini ($p = 0.041$), and 0.3 pp for Gemini 3.1 Pro ($p = 0.012$). Claude Haiku 4.5 also improves by 0.9 pp, with the reduction significant at the 10% level ($p = 0.088$). The changes for Claude Fable 5.1 and GPT-6 Astra are not statistically significant.

5. Conclusions

LLMs' ability to parse both historical demand data and qualitative information about products and markets makes their application as pricing agents particularly natural. However, evaluating their performance at pricing in practice is difficult, as pricing environments can vary substantially and outcomes at alternative prices are generally unobserved. In this paper, we construct a large-scale benchmark of single-product pricing instances to enable systematic evaluation of LLM pricing performance. Using this benchmark, we evaluate six general-purpose LLMs in two capability tiers and compare their performance with that of regression-based and best-observed-price heuristics. On standard static instances, we find that basic models underperform the regression heuristic, whereas all three advanced models significantly outperform it. However, outside of standard instances, the performance of advanced LLMs varies significantly across models and environments, and these models do not consistently outperform the heuristics across the benchmark. Our analysis of their performance in these difficult instances highlights heterogeneity in the models' stated reasoning as well as differences in demand estimation and subsequent price adjustment. We also find that prompt-level framing and distribution hints can improve LLM performance, but their benefits depend on the model and setting.

For practitioners, our findings support the use of advanced LLMs as pricing assistants, together with targeted human guidance and validation, echoing Cohen (2026). We emphasize that the ability of LLMs to produce a price and a plausible explanation does not establish the quality of the

recommendation. Practitioners using LLMs as pricing agents should first assess their performance in settings representative of their application, compare recommendations with statistical benchmarks, and validate the effects of prompt changes. Our prompt experiments show that role framing improves Gemini 3.1 Pro's performance, while distribution hints reduce mean profit losses for both Gemini 3.1 Pro and GPT-6 Astra. More broadly, these findings show that LLMs are suggestible: changes in how a pricing task is presented can improve or worsen performance, making careful prompt design and validation important (see also Cohen and Hage-Youssef (2026)). Thus, careful human oversight should focus on the quality of demand estimates, the treatment of contextual information, and the consistency of recommended prices with the stated objective.

Finally, we believe our work opens several directions for future research. In particular, it would be interesting to examine more closely how prompt design impacts recommended prices, and to develop a wrapper that uses these insights to improve overall pricing performance. Given the substantial variation across frontier models' stated reasoning and price recommendations in hard instances, another direction is to investigate whether their price recommendations can be ranked and combined into a single recommendation that improves pricing performance, building on methods for ensembling model outputs (Jiang et al. 2023, Huang et al. 2026). Beyond revenue performance, it would also be interesting to evaluate LLM pricing under other objectives, such as customer welfare. Our preliminary results suggest that advanced LLMs systematically overprice in some settings. In such settings, an important question is whether guardrails or targeted price adjustments can improve both seller revenue and customer welfare by correcting these prices downwards. Related experiments could compare revenue-maximizing prompts with prompts that incorporate a customer welfare objective or constraint. These extensions would help establish when revenue and customer welfare can improve together, when they involve a trade-off, and how reliably LLMs respond to such guidance.

References

7Learnings. 2026. Price optimization. https://7learnings.com/price-optimization/. Accessed July 27, 2026. 2

Allouah, Amine, Achraf Bahamou, Omar Besbes. 2023. Optimal pricing with a single point. *Management Science* **69**(10) 5866–5882. doi:10.1287/mnsc.2023.4683. 6

Baek, Jackie. 2026. LLMs can design near-optimal OR algorithms. arXiv:2608.27296. URL https://arxiv.org/abs/2608.27296. Version 3, revised October 13, 2026. 6

Baek, Jackie, Yaopeng Fu, Will Ma, Tianyi Peng. 2026. AI agents for inventory control: Human-LLM-OR complementarity. URL https://arxiv.org/abs/2602.12631v2. Version 2. 6

Baker, Walter, Dieter Kiewell, Georg Winkler. 2014. Using big data to make better pricing decisions. *McKinsey & Company* **1**(1) 1–4. 1

Besbes, Omar, Assaf Zeevi. 2009. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations research* **57**(6) 1407–1420. 5

Besbes, Omar, Assaf Zeevi. 2015. On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science* **61**(4) 723–739. doi:10.1287/mnsc.2014.2031. 6, 16

Bichler, Martin, Philip Hierhager, Kassian Köck, Fabian Pieroth. 2026. How language model agents reshape price competition. URL https://arxiv.org/abs/2610.21711. 6

Bu, Jinzhi, David Simchi-Levi, Li Wang. 2023. Offline pricing and demand learning with censored data. *Management Science* **69**(2) 885–903. doi:10.1287/mnsc.2022.4382. 6

Buynomics. 2026. Identify pricing optimization potential with the decision guide. https://www.buynomics.com/decision-guide. Accessed July 27, 2026. 2

Cohen, Maxime C. 2026. *Pricing in the Age of AI*. The MIT Press. URL https://mitpress.mit.edu/9780262056779/pricing-in-the-age-of-ai/. 2, 24

Cohen, Maxime C, Paul-Emile Gras, Arthur Pentecoste, Renyu Zhang. 2022. *Demand prediction in retail: A practical guide to leverage data and predictive analytics*. Springer. 2

Cohen, Maxime C., Eddy Hage-Youssef. 2026. Confirmation bias in LLM pricing recommendations. SSRN working paper 7111078. doi:10.2139/ssrn.7111078. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=7111078. 6, 25

den Boer, Arnoud V. 2015. Dynamic pricing and learning: Historical origins, current research, and new directions. *Surveys in Operations Research and Management Science* **20**(1) 1–18. doi:10.1016/j.sorms.2015.03.001. 1, 5

den Boer, Arnoud V., Bert Zwart. 2014. Simultaneously learning and optimizing using controlled variance pricing. *Management Science* **60**(3) 770–783. doi:10.1287/mnsc.2013.1788. 5

Dubé, Jean-Pierre, Sanjog Misra. 2023. Personalized pricing and consumer welfare. *Journal of Political Economy* **131**(1) 131–189. doi:10.1086/720793. 10, 30

Ferreira, Kris Johnson, Bin Hong Alex Lee, David Simchi-Levi. 2016. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management* **18**(1) 69–88. doi:10.1287/msom.2015.0561. 6

Fish, Sara, Yannai A. Gonczarowski, Ran I. Shorrer. 2024. Algorithmic collusion by large language models. URL https://arxiv.org/abs/2404.00806v6. Version 6, revised August 31, 2026; accepted to EC 2026. 6

Fish, Sara, Julia Shephard, Minkai Li, Ran I. Shorrer, Yannai A. Gonczarowski. 2025. EconEvals: Benchmarks and litmus tests for economic decision-making by LLM agents. URL https://arxiv.org/abs/2503.18825v4. Version 4, revised February 18, 2026. 2, 6

Giebel, Marek, Anja Rösner, Benjamin Schröder. 2026. Artificial intelligence, communication, and strategic pricing: Evidence from large language models. SSRN working paper 6832662. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=6832662. 6

Han, Jinhui, Ming Hu, Zishi Zhang. 2026. LLM pricing agents as rote learners. SSRN working paper 7555398. doi:10.2139/ssrn.7555398. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=7555398. 2, 6

Huang, Chengpiao, Kaizheng Wang. 2026. LLM-powered virtual population for demand simulation and pricing. URL https://arxiv.org/abs/2606.16183. 6

Huang, Chenyu, Zhengyang Tang, Shixi Hu, Ruoqing Jiang, Xin Zheng, Dongdong Ge, Benyou Wang, Zizhuo Wang. 2025. ORLM: A customizable framework in training large models for automated optimization modeling. *Operations Research* **73**(6) 2986–3009. doi:10.1287/opre.2024.1233. 7

Huang, Jingkai, Will Ma, Zhengyuan Zhou. 2026. Optimal Bayesian stopping for efficient inference of consistent LLM answers. *Proceedings of the 43rd International Conference on Machine Learning*, *Proceedings of Machine Learning Research*, vol. 306. PMLR, 47306–47331. URL https://proceedings.mlr.press/v306/huang26ak.html. 25

Jiang, Dongfu, Xiang Ren, Bill Yuchen Lin. 2023. LLM-Blender: Ensembling large language models with pairwise ranking and generative fusion. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 14165–14178. doi:10.18653/v1/2023.acl-long.792. URL https://aclanthology.org/2023.acl-long.792/. 25

Keppo, Jussi, Yuze Li, Gerry Tsoukalas, Nuo Yuan. 2026. On the fragility of AI agent collusion. URL https://arxiv.org/abs/2603.20281. 6

Keskin, N. Bora, Assaf Zeevi. 2014. Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations Research* **62**(5) 1142–1167. doi:10.1287/opre.2014.1294. 5, 16

Li, Menglong, David Simchi-Levi, Renfei Tan, Chonghuan Wang, Michelle Xiao Wu. 2026. Contextual offline demand learning and pricing with separable models. *Management Science* **72**(8) 6743–6761. doi:10.1287/mnsc.2023.04026. 6

Liu, Yunuo, Dawei Zhu, Zena Al-Khalili, Dai Cheng, Yanjun Chen, Dietrich Klakow, Wei Zhang, Xiaoyu Shen. 2025. PricingLogic: Evaluating LLMs reasoning on complex tourism pricing tasks. *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 7725–7734. doi:10.18653/v1/2025.emnlp-main.393. URL https://aclanthology.org/2025.emnlp-main.393/. 7

Ma, Chennan, Yanning Zhang, Siqi Hong, Xiuchong Wang, Fei Xiao, Keping Yang. 2026. AIGP: An LLM-based framework for long-term value alignment in e-commerce pricing. *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*. ACM. doi:10.1145/3770855.3818316. URL https://arxiv.org/abs/2606.26787. 2, 6

Oetzel, Sebastian, Tobias Maiberger. 2026. How price sensitive is AI? benchmarking LLM-generated willingness to pay in pricing analytics. *Journal of Marketing Analytics* doi:10.1057/s41270-026-00542-7. URL https://link.springer.com/article/10.1057/s41270-026-00542-7. Published online September 28, 2026. 6

Ramamonjison, Rindranirina, Timothy T. Yu, Raymond Li, Haley Li, Giuseppe Carenini, Bissan Ghaddar, Shiqi He, Mahdi Mostajabdaveh, Amin Banitalebi-Dehkordi, Zirui Zhou, Yong Zhang. 2023. NL4Opt competition: Formulating optimization problems based on their natural language descriptions. *Proceedings of the NeurIPS 2022 Competitions Track*, *Proceedings of Machine Learning Research*, vol. 220. PMLR, 189–203. URL https://proceedings.mlr.press/v220/ramamonjison23a.html. 6

Reuel, Anka, Amelia Hardy, Chandler Smith, Max Lamparth, Malcolm Hardy, Mykel J. Kochenderfer. 2024. BetterBench: Assessing AI benchmarks, uncovering issues, and establishing best practices. *Advances in Neural Information Processing Systems*, vol. 37. 21763–21813. URL https://papers.nips.cc/paper_files/paper/2024/hash/26889e8359e7ef8a7f5d77457364ca55-Abstract-Datasets_and_Benchmarks_Track.html. 6

Rusetski, Alexander. 2014. Pricing by intuition: Managerial choices with limited information. *Journal of Business Research* **67**(8) 1733–1743. doi:10.1016/j.jbusres.2014.02.020. 2

Salesforce. 2026. Agentforce sales: AI sales agents. https://www.salesforce.com/sales/ai-sales-agent/. Accessed July 27, 2026. 2

Spann, Martin, Marco Bertini, Oded Koenigsberg, Robert Zeithammer, Diego Aparicio, Yuxin Chen, Fabrizio Fantini, Ginger Zhe Jin, Vicki G Morwitz, Peter Popkowski Leszczyc, et al. 2025. Algorithmic pricing: Implications for marketing strategy and regulation. *International Journal of Research in Marketing* . 2

Tang, Jingwen, Zhengling Qi, Ethan Fang, Cong Shi. 2025. Offline feature-based pricing under censored demand: A causal inference approach. *Manufacturing & Service Operations Management* **27**(2) 535–553. doi:10.1287/msom.2024.1061. 6

Turpin, Miles, Julian Michael, Ethan Perez, Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, vol. 36. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ed3fea9033a80fea1376299fa7863f4a-Abstract-Conference.html. 6

Wang, Mengxin, Dennis J. Zhang, Heng Zhang. 2026. Large language models for market research: A data-augmentation approach. *Marketing Science* **45**(4) 728–751. doi:10.1287/mksc.2025.0009. 6

Yang, Cathy, Kevin Bauer, Xitong Li, Oliver Hinz. 2026a. My advisor, her ai, and me: Evidence from a field experiment on human–ai collaboration and investment decisions. *Management Science* **72**(1) 242–264. 5

Yang, Fenghua, Preet Baxi, Yi Zhang, Stefanus Jasin, Yanzhe Lei, Mo Liu, Parshan Pakiman. 2026b. Automated design of inventory policy with large language models: An exploratory study. URL https://arxiv.org/abs/2609.08071. 6

Yang, Zhicheng, Yiwei Wang, Yinya Huang, Zhijiang Guo, Wei Shi, Xiongwei Han, Liang Feng, Linqi Song, Xiaodan Liang, Jing Tang. 2025. OptiBench meets ReSocratic: Measure and improve LLMs for optimization modeling. *International Conference on Learning Representations*. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/3deb687c44d3687ace0729e5db3b4efd-Abstract-Conference.html. 7

Zheng, Mingqian, Jiaxin Pei, Lajanugen Logeswaran, Moontae Lee, David Jurgens. 2024. When "a helpful assistant" is not really helpful: Personas in system prompts do not improve performances of large language models. *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, 15126–15154. doi:10.18653/v1/2024.findings-emnlp.888. URL https://aclanthology.org/2024.findings-emnlp.888/. 6

Appendix A: Additional Benchmark Implementation Details

In this section, we include some additional details about the governing price-demand relationships which underpin each instance in our benchmark that were omitted from Section 2.

A.1. Empirically Calibrated Demand Function

Dubé and Misra (2023) estimate demand for *ZipRecruiter*'s monthly subscription using a randomized field experiment in which 7,867 prospective customers were assigned to one of ten prices ranging from \$19 to \$399. Their individual-level demand model is a heterogeneous binary logit:

$$\Pr(Y_i = 1 \mid p, x_i) = \frac{\exp\left(\alpha(x_i) + \beta(x_i)p\right)}{1 + \exp\left(\alpha(x_i) + \beta(x_i)p\right)}, \tag{3}$$

where Y_i indicates whether customer i subscribes, p is the monthly price, and both the intercept $\alpha(x_i)$ and price sensitivity $\beta(x_i)$ depend on the customer's observable characteristics. Consequently, aggregate demand at a common price is naturally represented as an average of heterogeneous logistic demand curves.

To obtain a parsimonious demand function for our benchmark, we approximate the paper's high-dimensional mixture by four equally weighted representative customer types. Let $\ell(z) = (1+\exp(-z))^{-1}$ and define:

$$\begin{aligned} q_1(p) &= \ell(0.53 - 0.003p), \qquad q_2(p) = \ell(-1.47 - 0.005p), \\ q_3(p) &= \ell(-0.96 - 0.007p), \qquad q_4(p) = \ell(-0.60 - 0.009p). \end{aligned} \tag{4}$$

Our empirical demand function is the equally weighted average:

$$q(p) = \frac{1}{4}\sum_{i=1}^{4} q_i(p), \tag{5}$$

where $q(p)$ is the expected fraction of prospective customers who subscribe at price p. The four slopes span the central range of price sensitivities estimated in the paper and have mean -0.006, equal to the authors' reported mean Lasso price coefficient. Conditional on these slopes, we choose the intercepts by least squares to match the ten experimental acquisition rates reported in their Table 5. The resulting curve matches these rates with a root mean squared error of approximately 0.014, or 1.4 percentage points.

The four functions in Eq. (4) are therefore not four customer types separately reported by Dubé and Misra (2023); rather, they provide a transparent reduced-form approximation to the paper's empirically estimated heterogeneous-logit demand model. If M_t prospective customers reach the paywall in period t, the corresponding expected quantity demanded is:

$$D_t(p) = M_t q(p). \tag{6}$$

A.2. Parameter Configurations and Optimal Prices

Table A.1 reports the parameters of each demand family and configuration and the resulting optimal prices. Marginal cost is zero, so the optimal price maximizes $pD(p)$; seasonal instances share the static optimum because seasonality scales the whole curve.

The duopoly columns of Table A.1 report the cross-price coefficient γ, seller 1's disclosed price q, and the best response $p^*(q)$ that maximizes $p\,[D(p) + \gamma(q-p)]^+$.

Table A.1 Parameter configurations of the four basic demand families

Family	$D(p)$	Set	Parameters	Static and seasonal p^*	Duopoly γ	Duopoly q	Duopoly $p^*(q)$
Linear	$[a-bp]^+$	a	$a=300,\ b=15$	10.00	3.00	12.95	9.41
		b	$a=260,\ b=10$	13.00	2.00	5.66	11.30
		c	$a=280,\ b=10$	14.00	2.00	10.99	12.58
Exponential	ae^{-bp}	a	$a=500,\ b=0.100$	10.00	3.68	12.95	9.06
		b	$a=300,\ b=0.125$	8.00	2.76	3.48	6.45
		c	$a=480,\ b=0.091$	10.99	3.21	8.63	9.29
Logistic	$\frac{L}{1+e^{a+bp}}$	a	$L=500,\ a=-6.386,\ b=0.50$	10.00	8.00	12.95	9.73
		b	$L=400,\ a=-5.000,\ b=0.30$	13.09	4.56	5.69	12.10
		c	$L=450,\ a=-6.261,\ b=0.35$	14.00	5.12	10.99	13.33
Capped log-log	$\min\{Q, e^a p^{-b}\}$	a	$Q=64.8,\ a=8.7765,\ b=2.000$	10.00	2.59	12.95	10.00
		b	$Q=48.3,\ a=7.9395,\ b=1.500$	15.00	0.97	6.53	15.00
		c	$Q=58.7,\ a=8.4678,\ b=1.833$	11.00	1.96	8.64	11.00

Note. Marginal cost is zero in every instance. Seasonal instances use the static parameters on weekdays and scale the whole curve by $s_{\text{weekend}}=1.5$ on weekends ($s_{\text{weekday}}=1$), so p^* is the same on both day types. In the duopoly, demand is $[D(p)+\gamma(q-p)]^+$ with $\gamma=0.2\,|D'(p^*)|$ at the static optimum; q is seller 1's disclosed price for the target period (the same for all histories of a configuration), and $p^*(q)$ is the best response to it. Optimal prices are rounded to two decimals; under capped log-log demand, p^* is the kink of the curve.

Appendix B: Prompt Architecture and Examples

B.1. Prompt Templates for Single-Seller and Competitive Markets

Figures B.1 and B.2 present the prompt templates for the stationary monopoly and duopoly markets.

B.2. Section-by-Section Prompt Architecture

Table B.1 gives the content of all eleven prompt sections of Section 3.2, side by side for the single-seller and competitive families.

B.3. Details of the Distribution Hint

The `auto` level of the distribution hint does not inject a single fixed sentence: it dispatches on the market family and, within the static family, on the demand family. Table B.2 records which disclosure each instance receives. Because the disclosed content differs qualitatively across market families, we report the hint's realized level alongside its nominal level and do not pool the `auto` cells across market families when interpreting its effect.

Appendix C: Additional Experiment Setup Details

In this section, we give additional details about our LLM pricing performance experiment which were omitted form Section 3.

C.1. Decoding Configuration

Table C.1 reports each model's decoding settings, the closest available approximation to deterministic, minimal-reasoning decoding.

Table B.1 Prompt architecture: section presence and content across benchmark families. Entries spanning both columns are shared across families. Quoted text is included verbatim in the prompt. † = optional or variant-dependent component; "—" indicates the section is not included in that family. In the competitive family the agent represents seller 2 and best-responds to seller 1's disclosed price.

	Section	Single-seller benchmark	Competitive benchmark
(1)†	Role	You are a pricing expert.	
(2)	Market setup	The market has a single seller. Each period, the seller posts a price and observes the resulting demand, recorded as total units sold.	The market has two sellers, seller 1 and seller 2, offering similar products. Each period, seller 1 posts a price first, then seller 2 posts a price and observes its resulting demand, recorded as total units sold. You represent seller 2.
(3)†	Product context	Identical product description, restated symmetrically for both sellers in the competitive family: The product is a premium personal care item positioned in the luxury segment of an online retail platform … Full text in Figs. B.1 and B.2.	
(4)	Data header	The following are T periods of observations where the seller posted prices at random.	The following are T periods of observations where both sellers posted prices at random.
(5)	Historical data	T rows with columns Day, Price (p_t), and Demand (D_t); an Environment (x_t) column† is included only in the seasonality family.	T rows with columns Period, Seller 1 price ($p_{1,t}$), Seller 2 price ($p_{2,t}$), and Seller 2 Demand ($D_{2,t}$).
(6)	Task description	Your task is to recommend the profit-maximizing price for the seller for period T+1. Note that the optimal price is not restricted to the historical data provided.	Prepended with: "Seller 1's price for period T+1 is given as $p_{1,T+1}$." Then: Your task is to recommend the profit-maximizing price for seller 2 for period T+1. Note that the optimal price is not restricted to the historical data provided.
(7)	Marginal cost	The seller's marginal cost is c per unit.	Seller 2's marginal cost is c per unit.
(8)	Noise hint	Demand observations contain random variation.	
(9)†	Distribution hint	Content depends on the demand family: the four candidate functional forms; or a structural-break, fat-tail, or estimated-from-data disclosure (Section B.3).	Seller 2's demand also includes a cross-price effect from seller 1's price: this effect is larger when seller 1 charges more, and smaller when seller 1 charges less.
(10)†	Seasonality hint	Demand varies systematically across different environments (e.g., weekday vs. weekend).	—
(11)	Closing instruction	Briefly explain your reasoning. Your final output must be a JSON dictionary with two keys: reasoning and price. Limit your reasoning to under 20 sentences. You must directly think within the dictionary so that I can easily parse. Do not wrap the JSON in markdown ""json"" markers. Followed by the example {"reasoning": "...", "price": 12.34}	

Figure B.1 Prompt template for single-seller markets.

[Role: You are a pricing expert.]

Market Setup: The market has a single seller. Each period, the seller posts a price and observes the resulting demand, recorded as total units sold.

[Product Context: The product is a premium personal care item positioned in the luxury segment of an online retail platform. The seller has established a reputation for quality craftsmanship and exclusive materials over the past five years. Customer reviews highlight the product's superior finish and longevity, with an average rating of 4.8 out of 5. The seller uses a white-glove delivery service with guaranteed next-day shipping in premium packaging. A members-only loyalty program is available by invitation, with low but highly engaged participation. The product is produced in limited quantities and inventory is not a binding constraint.]

Data Header: The following are T periods of observations where the seller posted prices at random.

Historical Data Table: Historical data (T days):
| Day | Price (p_t) | Environment (x_t) | Demand (D_t) |
followed by T rows of observations.

Task Description: Your task is to recommend the profit-maximizing price for the seller for day $T+1$(environment: x_{T+1}). Note that the optimal price is not restricted to the historical data provided.

Marginal Cost: The seller's marginal cost is c per unit.

Noise Hint: Demand observations contain random variation.

[Distribution Hint: The demand follows one of four functional forms: linear, exponential, logistic, or a capped log-log curve.]

[Seasonality Hint: Demand varies systematically across different environments (e.g., weekday vs. weekend).]

⟨Closing Instruction: Briefly explain your reasoning. Your final output must be a JSON dictionary with two keys: "reasoning" and "price". Limit your reasoning to under 20 sentences. You must directly think within the dictionary so that I can easily parse. Do not wrap the JSON in markdown ```json``` markers.

Example format:
{
"reasoning": "Your reasoning here",
"price": 12.34
} ⟩

Note. The Environment column, the next period's environment in the task description, and the seasonality hint appear only in seasonality instances, where the seasonality hint takes the place of the distribution hint. Required components appear in blue, optional components in orange, and the closing instruction in green.

C.2. Replication Stability Checks

Table C.2 reports mean relative profit loss by replication together with tests of equality across replications.

Figure B.2 Prompt template for competitive (two-seller) markets.

[Role: You are a pricing expert.]

Market Setup: The market has two sellers, seller 1 and seller 2, offering similar products. Each period, seller 1 posts a price first, then seller 2 posts a price and observes its resulting demand, recorded as total units sold. You represent seller 2.

[Product Context: Both products are premium personal care items positioned in the luxury segment of an online retail platform. Both sellers have established a reputation for quality craftsmanship and exclusive materials over the past five years. Customer reviews highlight each product's superior finish and longevity, with an average rating of 4.8 out of 5. Both sellers use a white-glove delivery service with guaranteed next-day shipping in premium packaging. A members-only loyalty program is available for each, by invitation, with low but highly engaged participation. Both products are produced in limited quantities and inventory is not a binding constraint for either seller.]

Data Header: The following are T periods of observations where both sellers posted prices at random.

Historical Data Table: Historical data (T periods):
| Period | Seller 1 price | Seller 2 price | Seller 2 Demand |
followed by T rows of observations.

Task Description: Seller 1's price for period $T + 1$ is given as $\$p_{1,T+1}$. Your task is to recommend the profit-maximizing price for seller 2 for period $T + 1$. Note that the optimal price is not restricted to the historical data provided.

Marginal Cost: Seller 2's marginal cost is c per unit.

Noise Hint: Demand observations contain random variation.

[Distribution Hint: Seller 2's demand also includes a cross-price effect from seller 1's price: this effect is larger when seller 1 charges more, and smaller when seller 1 charges less.]

⟨Closing Instruction: Briefly explain your reasoning. Your final output must be a JSON dictionary with two keys: "reasoning" and "price". Limit your reasoning to under 20 sentences. You must directly think within the dictionary so that I can easily parse. Do not wrap the JSON in markdown ```json``` markers.

Example format:
{
"reasoning": "Your reasoning here",
"price": 12.34
} ⟩

Note. The agent represents seller 2 and best-responds to seller 1's disclosed price. Required components appear in blue, optional components in orange, and the closing instruction in green.

Appendix D: Additional Results

Table D.1 reports the separate regression for each LLM summarized in Section 4.1.

Table B.2 Realized content of the `auto` distribution hint. Counts are instances per market family (of 144). Under the `none` level no disclosure is made in any family.

Market	Applies to	Inst.	Disclosure
Static	linear, exponential, logistic, capped log–log	72	Names the four candidate functional forms, without identifying which one generated the data
Static	piecewise-linear, piecewise-exponential	36	States that the demand curve may exhibit a structural break at some price threshold
Static	Weibull	18	States that demand decays with a fat tail, remaining substantial at high prices
Static	mixed logit	18	States that the curve was estimated from scatter-plot data by a pricing expert
Seasonality	all eight families	144	States that demand varies systematically across environments (weekday vs. weekend); *no* functional-form disclosure is made
Duopoly	all eight families	144	States that the focal seller's demand includes a cross-price effect that is larger when the rival charges more; *no* functional-form disclosure is made

Table C.1 Temperature and Reasoning Settings by Model.

Model	Temperature	Reasoning
Claude Haiku 4.5	0	Off
GPT-5 mini	Not accepted	Effort set to minimal
Gemini 2.5 Flash	0	Thinking budget set to 0
Claude Fable 5.1	Not accepted	Effort set to low
GPT-6 Astra	Not accepted	Effort set to low
Gemini 3.1 Pro	0	Thinking budget set to 1

Note. Temperature 0 selects the most likely token at each step, which minimizes randomness in the output. Reasoning (called thinking by Anthropic and Google) is the hidden deliberation a model performs before answering.

Table C.2 Mean relative profit loss across replications.

	Mean relative profit loss (%)					
Model	**Rep. 1**	**Rep. 2**	**Rep. 3**	**Rep. 4**	**Rep. 5**	p
Basic LLMs						
Claude Haiku 4.5	8.72	8.71	8.87	8.60	9.14	0.469
Gemini 2.5 Flash	8.09	7.43	7.57	7.37	7.44	0.494
GPT-5 mini	10.30	10.17	10.49	12.08	10.86	0.155
Advanced LLMs						
Claude Fable 5.1	2.75	2.89	–	–	–	0.349
Gemini 3.1 Pro	3.33	3.38	–	–	–	0.734
GPT-6 Astra	3.47	3.54	–	–	–	0.472

Note. Means and tests use 432 prompts: the static analysis (72), the seasonal and duopoly analyses (72 each), and the three prompt variants (72 each); the difficult-demand prompts are not included. The p-values test H_0: mean relative profit loss is equal across replications. For basic LLMs, the test is a repeated-measures ANOVA with replications as conditions and prompts as subjects (F with 4 and 1,724 degrees of freedom); for advanced LLMs, it is a paired t test between the two replications (431 degrees of freedom). Responses that return no valid price receive a loss of 100%.

Table D.1 Static Single-Seller Market: Regression of Each LLM's Relative Profit Loss on Demand Family, History Length, and Noise Level

	Basic LLMs			Advanced LLMs		
	Claude Haiku 4.5	GPT-5 mini	Gemini 2.5 Flash	Claude Fable 5.1	GPT-6 Astra	Gemini 3.1 Pro
Demand family and parameter set (base: linear, set a)						
Linear, set b	−8.94	−10.31**	3.30	−0.24	0.06	0.03
	(9.60)	(3.75)	(3.05)	(0.60)	(1.70)	(1.74)
Linear, set c	−12.96	−11.79*	−1.21	−0.28	0.09	0.00
	(9.60)	(4.69)	(2.62)	(0.68)	(1.71)	(1.75)
Exponential, set a	−15.64	−11.39*	0.55	−0.18	0.21	0.27
	(9.81)	(4.32)	(2.72)	(0.69)	(1.62)	(1.71)
Exponential, set b	−12.73	−10.89*	−0.33	1.80*	0.50	0.72
	(9.64)	(4.05)	(2.69)	(0.81)	(1.53)	(1.61)
Exponential, set c	−15.52	−11.27*	1.26	−0.03	0.16	0.14
	(9.73)	(4.28)	(3.01)	(0.64)	(1.70)	(1.73)
Logistic, set a	3.67	1.00	0.65	0.96	1.61	1.25
	(14.63)	(5.99)	(2.58)	(0.75)	(1.69)	(1.70)
Logistic, set b	−13.73	−7.03	−0.39	−0.01	0.27	0.27
	(10.26)	(5.12)	(2.53)	(0.53)	(1.56)	(1.61)
Logistic, set c	−14.18	−5.59	5.35	1.96	0.83	0.97
	(9.85)	(5.50)	(5.25)	(1.46)	(1.88)	(2.03)
Capped log-log, set a	−3.64	3.87	17.15***	6.26**	8.77***	8.27***
	(12.19)	(5.58)	(4.59)	(2.24)	(2.35)	(1.94)
Capped log-log, set b	−4.74	1.78	19.01*	7.71***	18.57*	16.58+
	(9.83)	(8.67)	(8.80)	(0.87)	(8.41)	(9.21)
Capped log-log, set c	2.32	5.80	16.30***	9.17***	12.97***	13.31***
	(10.00)	(4.07)	(3.38)	(1.96)	(2.81)	(2.57)
History length (base: short, 5 observations)						
Medium (15)	5.14	−6.16*	−0.73	−1.14+	−2.72	−3.31
	(3.81)	(2.95)	(2.69)	(0.67)	(2.16)	(2.32)
Long (45)	−0.22	−3.90	−4.22+	−2.09*	−4.51*	−4.53+
	(2.77)	(2.71)	(2.40)	(0.90)	(2.11)	(2.26)
Noise (base: $\sigma = 0.1$)						
$\sigma = 0.3$	−0.51	0.66	1.24	1.87*	1.28+	1.34*
	(1.75)	(1.98)	(1.50)	(0.70)	(0.66)	(0.59)
Constant (base cell)	15.73+	18.29***	2.51	0.48	1.78	1.97
	(8.58)	(4.06)	(2.81)	(0.51)	(1.69)	(1.80)
Observations	72	72	72	72	72	72
Price histories (clusters)	36	36	36	36	36	36
R^2	0.392	0.444	0.574	0.670	0.655	0.611

Note. Separate OLS for each LLM of its instance-level relative profit loss (percentage points; each instance's loss is the mean over its replications, five for the basic LLMs and two for the advanced) on demand family × parameter set, history length and noise. Baseline prompt (expert role, no product context, no distribution hint); 72 static instances. The constant is the loss in the base cell (linear, set a, short history, $\sigma = 0.1$). Standard errors clustered by price history in parentheses; p-values from $t(G-1)$. $^{+}p < 0.10$, $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.