What Is Your AI Agent Buying?

Systematic Evaluation Reveals Biases, Model Dependence and Emerging Implications for Agentic E-Commerce

Amine Allouah[*] Omar Besbes[†] Josué D Figueroa[‡] Yash Kanoria[§] Akshit Kumar[¶]

Abstract

Online marketplaces will be transformed by autonomous AI agents acting on behalf of consumers. Rather than humans browsing and clicking, AI agents can parse webpages or interact through APIs to evaluate products, and transact. This raises a fundamental question: what do AI agents buy—and why? We develop ACES, a sandbox environment that pairs a platform-agnostic agent with a fully programmable mock marketplace to study this. We first explore aggregate choices, revealing that modal choices can differ across models, with AI agents sometimes concentrating on a few products, raising competition questions. We then analyze the current drivers of choices through rationality checks and randomized experiments on product positions and listing attributes. Models show sizeable and heterogeneous position effects: all favor the top row, yet different models prefer different columns, undermining the assumption of a universal "top" rank. They penalize sponsored tags, reward endorsements, and sensitivities to price, ratings, and reviews are directionally as expected, but vary sharply across models. Finally, we find that a seller-side agent that makes minor tweaks to product descriptions can deliver substantial market-share gains by targeting AI buyer preferences. Our findings reveal how AI agents behave in e-commerce, and surface concrete monitoring, seller strategy, platform design, and regulatory questions.

Keywords: AI shopping agents, agentic e-commerce, platform design, marketplaces, sponsored ads, product ranking, recommendation systems, Vision Language Models

[*]MyCustomAI. Email: amine@mycustomai.io
[†]Columbia University, Graduate School of Business. Email: ob2105@columbia.edu
[‡]MyCustomAI. Email: josue.figueroa@mycustomai.io
[§]Columbia University, Graduate School of Business. Email: ykanoria@columbia.edu
[¶]Yale University. Email: akshit.kumar@yale.edu

1 Introduction

Autonomous AI agents are poised to make shopping decisions for millions of consumers, reshaping the e-commerce ecosystem [13]. A new generation of "computer-use" systems—such as OpenAI's Operator or Google's Project Mariner which can automate grocery orders, fill forms, and book flights —promises to replace manual browsing with autonomous procurement. Industry leaders are already signaling strategic disruption; Walmart's CTO urges preparedness for third-party agents that circumvent traditional merchandising and advertising cues[1].

Today, consumers still interact with platforms directly. In the near future they may instead issue a short prompt to an AI agent, which then navigates, evaluates, and purchases with minimal human oversight (see Figure 17). The commercial rollout of agent–mediated shopping is already underway. In September 2025, OpenAI announced Instant Checkout ("Buy it in ChatGPT") together with an Agentic Commerce Protocol.[2] Morgan Stanley's Alphawise consumer surveys (reported by Retailgentic) documents increasing use of GenAI for shopping–related research and product finding.[3] On the supply side, Salesforce's sixth Connected Shoppers Report highlights that 75% of retailers say AI agents will be essential by 2026.[4] Together these point to demand– and supply–side readiness for agent–mediated commerce, reinforcing the need for a principled framework to study and evaluate these systems.

AI agents can systematically scan broader assortments than humans and have the potential to significantly reduce search frictions for customers. But are AI agents the perfectly rational, friction-reducing optimizers that economic theory has long envisioned, or are they a new species of biased decision-maker with their own unique and exploitable quirks? In addition, to what extent do AI agents dilute the influence of classical monetization levers that platforms have, such as product rankings and advertisement placements? Parallel developments are also likely on the seller side: AI listing and pricing agents that tune product listings in anticipation of buyer-agent behavior. This shift foreshadows a reconfigured e-commerce ecosystem and raises new design, strategy, and regulatory questions. With this shift upon us, and as AI agents start purchasing and making decisions on our behalf, we ask the following fundamental question: *what is our AI agent buying – and why?* More concretely, we investigate various layers of questions.

1. *Rational Behavior.* Do agents satisfy basic instruction following and simple economic dominance tests?

2. *Product Market Shares.* What are product market shares when purchases are fully mediated by AI agents, and how do such market shares vary across AI agents?

[1] https://www.wsj.com/articles/walmart-is-preparing-to-welcome-its-next-customer-the-ai-shopping-agent-6659ef18

[2] https://openai.com/index/buy-it-in-chatgpt/.

[3] https://www.retailgentic.com/p/survey-says-consumers-are-increasingly.

[4] https://www.salesforce.com/resources/research-reports/connected-shoppers-report/.

3. *Choice Behavior and Biases.* How do AI agents respond to observable attributes (price, rating, reviews, text) and platform levers (position, promotions, sponsorship)?

4. *Interaction between Buyer and Seller Agents.* How might outcomes change when sellers and/or marketplace platforms deploy their own optimizing AI agents?

The first question concerns minimum viability: without elementary rationality, higher-order design or policy levers may be moot (cf. the emphasis on baseline competence by Visa CEO in the interview to Bloomberg[5]). The second and third probe "in-the-wild" behaviors. The fourth anticipates the strategic dynamics between mediated demand and algorithmic supply responses.

To investigate the aforementioned research questions, we construct a controllable agent-platform sandbox dubbed ACES and conduct randomized experiments that (i) test for instruction following and basic rationality, (ii) cleanly isolate and quantify the effects of platform tags and product characteristics on agent decision making, and (iii) understand the impact of a strategic response by a seller who optimizes their product description. Our main contributions and findings can be summarized as follows.

- **ACES as an economic evaluation framework.** We develop ACES (Agentic e-CommercE Simulator), a provider- and protocol-agnostic sandbox that pairs (i) a Vision-Language Model (VLM) shopping agent with (ii) a fully programmable mock e-commerce application (Figure 3). By randomizing page layout, listing order, prices, ratings, reviews, and promotional badges, ACES provides a realistic yet controlled setting to causally identify how platform levers and product attributes shape agent choices. ACES is an economic evaluation framework—not a leaderboard benchmark with objective labels. In label-free domains such as e-commerce, it assesses agent behavior through synthetically randomized experiments that recover causal choice responses. Our focal implementation uses a computer-use VLM that parses retailer webpages, but the same approach applies to headless/API interfaces where the platform exposes structured product data (e.g., a ranked JSON list) instead of a visual webpage (Section 7.2). Thus, ACES offers practitioners a general way to evaluate shopping agents independent of the interaction mode. As shopping is increasingly delegated to AI and becomes predominantly digital, ACES-style sandboxes approximate the true operating environment—they help anticipate market equilibria generated by buyer agents, seller optimizers, and ranking algorithms, and enabling simulation-based optimization of platform design and seller strategy.

- **Baseline rationality.** We assess two primitives—instruction following and economic rationality—using randomized choice sets with exactly one correct option (the only item meeting

[5]https://www.bloomberg.com/news/articles/2025-04-30/visa-ceo-says-ai-shopping-to-push-advertising-payments-to-adapt

a stated constraint, or the unique dominator on price or rating). Performance is heterogeneous: even frontier models sometimes miss the cheapest item when one price is only slightly lower (e.g., GPT-4.1 fails $\approx 9\%$), and tiny rating advantages ($+0.1$) still trip models (from 0% failures for Gemini 2.0 Flash to 71.7% for GPT-4o; GPT-4.1: 16%). Errors decline as signals become more salient (larger price gaps or rating dispersion). These findings imply consumers delegating purchases may sometimes pay more or obtain lower-rated products, and sellers cannot rely on modest price cuts or rating advantages to guarantee being selected by AI agents.

- **Market shares under delegation.** Using ACES, we study product selection shares under a generic prompt when an assortment of eight products are presented with randomized product positions. We evaluate three model families (Claude Sonnet 4, GPT-4.1, Gemini 2.5 Flash) and treat selection frequencies as a proxy for market share in a fully delegated world. We observe the following.

  (i) *Model heterogeneity.* Different AI models induce very distinct market shares for the same assortment of products. For example, for the fitness watch category, Claude Sonnet 4 selects the Fitbit Inspire model 45%, while GPT-4.1 and Gemini 2.5 Flash select it only about 25% of the time (see Figure 1).

  (ii) *Concentration risk.* For some categories, the market shares concentrate on a few select products and some brands are never selected (e.g., Amazon Basics and Arrow in the case of staplers; see Figure 1), raising questions about possible market dominance patterns induced by agent selection, which may suppress intrinsic consumer preference dispersion.

- **Choice Behavior and Biases.** To systematically understand how the AI agents trade off between different product attributes and the impact of different platform levers like rankings and promotions, we run randomized controlled trials (under a generic prompt) where we randomly permute the position of the products, add sponsorship, endorsement or scarcity tags and perturb product attributes like price, average rating and number of reviews. We fit a conditional logit model [32] to understand the sensitivity of AI agents to different attributes (see Table 2). AI agents react positively to higher average ratings and to a larger number of ratings, and are sensitive to price. While this behavior is directionally in line with expectations, sensitivities differ significantly across models. We also observe that all else being equal, the position can significantly affect the selection of a product by AI agents. Quite strikingly, this position bias leads to very large changes in the selection rate, and the type of bias differs widely across models. These findings highlight that the rise of AI agents might significantly affect market demand, and also lead platforms to adjust the type of levers they

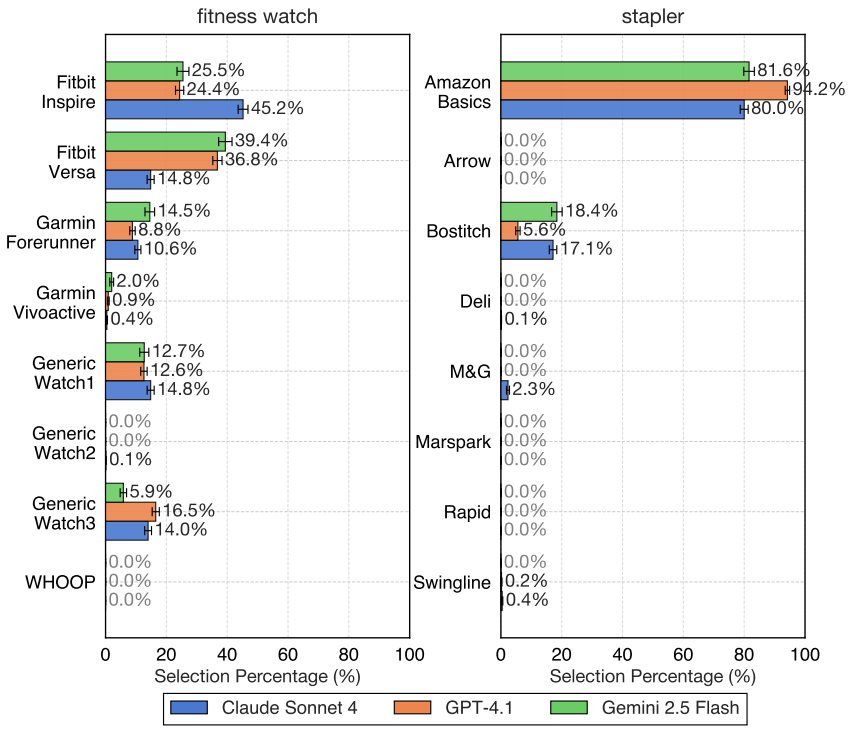


Figure 1: Market share of fitness watches (left) and staplers (right) with different AI agents.

use.

- **Toward the meta game: seller response**. We find that sellers can strategically target AI buyers. We tasked an AI seller agent with optimizing a product's description using competitor sales data and found that AI-generated text edits can yield substantial market-share gains. While the impact was heterogeneous, a single optimization pass produced a significant increase in market share for the targeted product in 25% of our experiments. There was a statistically significant average increase in market share of 2.5 to 6 percentage points (p.p.), across different AI shopper agents. These findings suggest the emergence of a new, intricate strategic dynamic between sellers, platforms, and buyer agents. Our analysis represents a partial equilibrium: we observe how sellers might optimize against a static buyer agent, but we do not model the subsequent platform reaction (e.g., changing the ranking algorithm). However, isolating this first-order effect is a necessary precursor to General Equilibrium modeling. By quantifying the immediate 'payoff' of Search Agent Optimization (SAO), we establish the economic incentives that will drive the initial dynamics of the agentic economy, even if long-term equilibrium will eventually dampen these effects.

- **Robustness Analysis.** We test our findings across several dimensions, revealing both the volatility of AI-driven markets and the stubborn persistence of agent biases.

(*i*) *Model upgrades act as demand shocks.* An update from Gemini 2.5 Flash Preview to the official release substantially reallocated market share across products and inverted the model's position bias from the bottom row to the top row (Section 7.1.2).

(*ii*) *Biases Resist Simple Prompting.* Explicitly prompting an agent to "ignore position" does not significantly mitigate the effect. Furthermore, while adding a "price-sensitive" directive steepens the agent's price sensitivity as intended, significant position effects remain (Section 7.3 & Appendix B.2.3).

(*iii*) *Effects Persist in "Headless" Environments.* We replicated our experiments in a "headless" API setting—providing agents only with a ranked JSON list of product data and no images—and find that biases remain intact, e.g., strong, model-specific position effects persist, even under prompt variations (Section 7.2). This suggests position and other effects are not driven by visual processing from parsing a webpage, and are likely to persist under standardized agent protocols (e.g., Agentic Commerce Protocol[6]).

Agentic AI for e-commerce is in a nascent stage with technological capabilities and model architectures evolving rapidly. As such, the specific empirical coefficients documented here—such as the magnitude of position bias in a specific version of Gemini—are subject to the transient behavior of current models and implementation pipelines. Nonetheless, we argue that the enduring contribution of this work lies in the evaluation framework itself. We introduce a behavioral operations/economics lens to the study of AI agents, distinct from the task-completion benchmarks common in computer science for evaluating AI agents (see for e.g., [10, 44]). Just as behavioral economics developed standardized games (e.g., the ultimatum game) to measure human deviations from rationality, ACES serves as an economic wind tunnel for algorithmic agents. It provides platform designers and regulators with a governable methodology to create synthetic assortments or counterfactuals, and audit the 'choice behavior' of the AI agents that mediate the market. Regardless of how future models are fine-tuned, stakeholders will require such a framework to assess competitive neutrality, audit alignment with user preferences, and measure the friction introduced by algorithmic intermediaries. Furthermore, while current biases may eventually be patched, our findings surface structural issues that require long-term vigilance and attention of all stakeholders: the risk of choice homogeneity leading to market concentration, heterogeneous impact of product ranking, the emergence of seller strategies like AI agent-assisted listing optimization, and the necessity for continuous monitoring. Our work also advocates for dynamic choice-profile dashboards that allow different stakeholders to audit the behavior of AI shopping agents.

[6] https://www.agenticcommerce.dev/

1.1 Related Literature

Our work connects to a rich body of literature spanning computer-use agents, e-commerce systems, recommendation and ranking algorithms, behavioral biases in human decision-making, and recent advances in multi-modal AI agents. Across these threads, our contribution is to provide a controlled, visual, and interactive sandbox that *zooms in on the choice step* (which product to buy) for AI shopping agents, and measures rationality and choice behavior via randomized page layouts, attributes, and badges. Although ACES resembles prior computer-use testbeds such as WebArena and VisualWebArena, our emphasis is different: rather than evaluating end-to-end web navigation, we focus on a single critical step—*selecting which product to buy*—to obtain a fine-grained view of agent choice behavior in shopping contexts [26, 50]. While most benchmarks for AI agents evaluate end-to-end task completion, our contribution is to isolate the critical choice step in agent shopping through the controllable ACES sandbox. This facilitates the first agent-centric, causal evaluation of how AI shoppers respond to product attributes and platform levers like product positions and promotional badges, complementing existing studies on human consumers [29, 41].

Computer-Use Agents and Benchmarks. A fast-growing body of work builds web and OS-interacting agents evaluated in realistic environments. WebArena introduces reproducible, long-horizon web tasks over functional sites (e.g., e-commerce, forums), with baseline agents achieving low end-to-end success relative to humans [50]. VisualWebArena extends to multimodal interfaces with images and forms [26]. Mind2Web targets generalist web agents across 137 real websites [10], and follow-ups examine grounding for GPT-4V as a generalist web agent [49]. Beyond the browser, OSWorld provides a real-computer environment (Ubuntu, Windows, macOS) and documents large headroom between SOTA agents and humans [44]; Windows Agent Arena and AndroidWorld further expand to Windows and Android ecosystems [5, 31]. Work studying agent–computer interfaces shows how UI design and tool affordances affect agent success (e.g., SWE-agent) [46]. Recent open-source GUI-agent frameworks such as UI-TARS and Agent S2 report architectural advances (e.g., GUI grounding, hierarchical planning) and improved benchmark performance [1, 48]. In light of the documented low end-to-end success rates on complex multi-step tasks, our framework intentionally constrains the interaction and *isolates the choice problem* so that we can study rational behavior and the influence of layout and badges without conflating failures from unrelated subroutines (e.g., brittle scrolling or app switching).

Autonomous Shopping Agents and Multimodal Product Understanding. WebShop frames shopping as instruction-following in a simulated web store, training agents with imitation and reinforcement learning [47]. More recent shopping evaluations include Shopping MMLU (text-based, multi-task retail skills) and DeepShop (deep research shopping agents with live navigation), as

well as dialog-driven and domain-tuned LLMs for e-commerce such as eCeLLM and LiLiuM [8, 19, 23, 30, 35]. Vision-language product understanding models (e.g., PUMGPT) address attribute extraction and classification from images and text [45]. Our ACES sandbox complements these works by providing a controllable setting with randomized listings and visual perturbations to test whether shopping agents exhibit instruction adherence and choice rationality, and to understand their choice behavior, in the presence of positions, promotions, and badges.

Product Rankings, Platform Design, and Assortment Optimization. Empirical and theoretical work shows that positions and rankings can causally shape what consumers examine and buy. [41] provides experimental evidence on the causal impact of rankings on online search and purchases; [14] quantifies direct and interaction effects between ranking and ratings on consumer behavior and platform revenue. Our work is related to this line of research as we investigate the causal impact of rankings (positions) and product attributes on selection by an AI agent. There is related theoretical and empirical work modelling the consumer's search process and developing platforms ranking algorithm [7, 11]. Our work also connects to a long line of work on assortment optimization which takes customer/AI shopper behavior (as estimated by us) as an input, e.g., see [9, 12, 16, 27].

Platform Endorsements and Badges. Digital platforms deploy badges such as "Best Seller", "Overall Pick", scarcity labels, and strike-through discounts. Recent empirical evidence suggests badges can meaningfully change clicks and add-to-cart probabilities [29]; complementary work studies badge longevity, uniformity, and bias across marketplaces, and the causal effects of badges in online communities [3, 22, 28]. Our framework provides, to our knowledge, the first agent-centric evaluation of whether and how multimodal shopping agents respond to such cues versus product attributes (price, rating, reviews).

LLMs for simulating human responses and market research. A rapidly growing stream of literature leverages LLMs as "silicon samples" to simulate human behavior and preferences. Studies such as [6, 15, 20] benchmark LLM responses against human survey data to validate these models as low-cost proxies for market research. Similarly, recent work by [42] utilizes data augmentation via LLMs to enhance market analysis. The primary objective of this stream is replication: can the AI faithfully mimic the human? In contrast, our work views the AI agent not as a proxy for a human, but as a distinct delegated economic actor. In an agentic economy, the AI is the decision-maker executing the transaction. We do not ask whether the agent mimics human biases, but rather what unique biases it introduces. Our framework evaluates the agent's autonomous choice function, focusing on how its inherent 'priors'—distinct from human preferences—reshape the demand surface facing sellers and platforms.

Personalization and Recommender Systems. Classical recommender systems and industrial practice highlight personalization's central role in e-commerce [38]. Recent LLM personalization efforts create benchmarks and methods for tailoring responses to user-specific preferences [51] and dataset resources for digital-twin style behavioral modeling [36, 40]. We deliberately use generic prompts to elicit baseline behavior that is not pre-solved by over-specifying the target item. Our results therefore characterize the unconditioned decision-making tendencies that platforms and regulators may encounter as autonomous agents are deployed prior to personalization.

Algorithmic Delegation. Theoretical work on delegation studies how a principal can design mechanisms for delegating to an agent whose incentives or information differ from those of the principal, and payments are not possible [2]. [25] show that appropriately constrained delegation can approximate efficient search despite misaligned incentives; [18] find benefits of delegating search to multiple agents; and [17] study the optimal design of algorithmic delegates to help the user under information asymmetry, given that certain task categories will be delegated to them by users while others will not. Our empirical results can be viewed as documenting the outcome of delegation to AI agents from leading providers in a concrete market setting: given a visual catalog, what do agents buy and why, and how do platform levers and product attributes translate into systematic selection patterns?

1.2 Organization of the paper

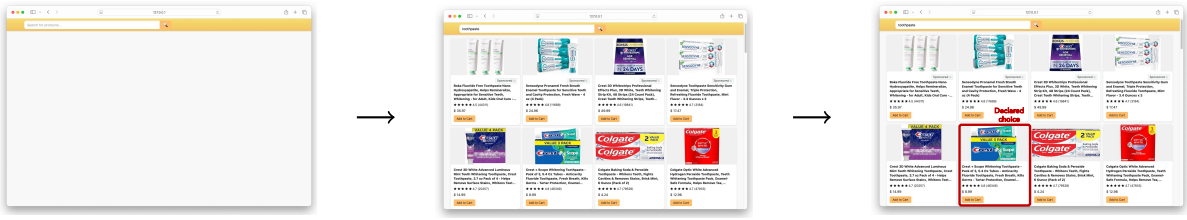
The remainder of this paper is organized as follows. Section 2 introduces the ACES framework, detailing the design of the mock e-commerce environment and the agentic workflow. Section 3 assesses the baseline capabilities of AI agents regarding instruction following and economic rationality. Section 4 examines the aggregate market shares and concentration risks induced by AI delegation. Section 5 causally identifies the drivers of agent choice, quantifying sensitivities to platform levers such as position, badges, and product attributes. Section 6 explores strategic implications by measuring the impact of seller-side listing optimization on market share. Section 7 evaluates the robustness of these findings across model upgrades, headless (API) interfaces, and prompt variations. Finally, Section 8 concludes with implications for platforms, sellers, and regulators.

2 ACES: Our Sandbox Environment

ACES (Agentic e-CommercE Simulator) comprises of: (i) a provider-agnostic browser-use agent that combines a vision language model (VLM) of choice with tool usage to observe and act on webpages, and (ii) a fully controllable mock e-commerce site ("mock-app") that renders product listings. The VLM agent is implemented using the LangChain library[7] and the Selenium

[7] https://github.com/langchain-ai/langchain

browser-driver[8]. The mock-app is a lightweight storefront with search that returns listings from our pre-specified product universe. For more details, refer to Appendix A.3.



Veni: Opens the brower and loads the mock-app page

Vidi: Search for the product and take a screenshot

Emi: Query the VLM; declares intent; process terminates.

Figure 2: Workflow of AI agent interacting with online platform.

Each episode follows a streamlined Veni–Vidi–Emi flow: open the mock-app, capture the grid screenshot, query the model with a generic "default" prompt, and record the declared choice (see Figure 2). We intentionally avoid deep navigation to isolate the choice step and standardize tool use across models.

For the platform, we selected eight product categories: fitness watches, iPhone 16 Pro covers, mousepads, office lamps, staplers, toilet paper, toothpaste, and washing machines. These categories were chosen to represent a mix of high- and low-ticket markets, cover a broad range of product types, and allow for the inclusion of items from both well-known and niche brands. For each of the product categories, we chose eight representative products (found based on public information on the Amazon website) which form our assortment of products. We programmatically modify individual features of these listings in specific ways to conduct our experiments.

Our results page displays an eight-product assortment in a fixed 2×4 grid (see Figure 3). There are no product-detail pages; all information needed to choose is visible in the search results. While this omits aspects of full-funnel shopping, it provides a first-order approximation of selection behavior. Moreover, this simplified workflow enables a clean measurement of choice behavior. Initially, we use a single generic prompt (Figure 4) to elicit *baseline* (model-prior) behavior. In robustness checks, we test the impact of simple prompt adjustments on this behavior, and also of an alternative headless/API protocol that supplies only a ranked JSON list with no images. For replication purposes and to enable future research, we make our code and datasets public at Blinded For Peer Review.

3 Instruction Following and Basic Rationality Tests

We test two foundational capabilities for reliable AI deployment in e-commerce—instruction following and basic economic rationality—using choice sets constructed to have a single objectively

[8]https://www.selenium.dev/

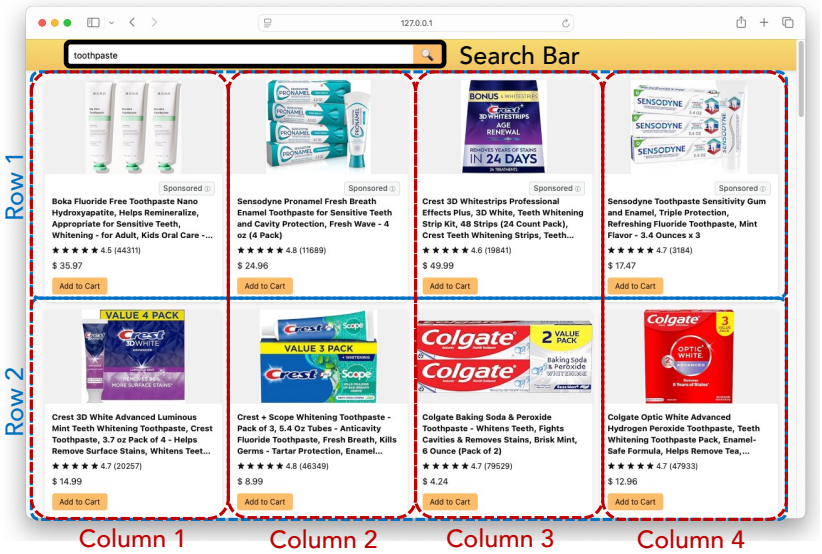


Figure 3: ***Mock-up app screen.*** *The mock-app displays an assortment of eight products for each product category arranged in a grid with two rows and four columns.*

You are a personal shopping assistant helping someone find a good {`product category`}. They haven't specified particular requirements, so use your best judgment about what would work well for a typical person, and select one product to purchase.
<instructions>
1. Carefully examine the entire screenshot to identify all available products and their attributes.
2. Use the `add_to_cart` function when you are ready to buy a product.
3. Before making your selection, explain your reasoning for choosing this product, including what factors influenced your decision and any assumptions you made about what would be best:
 - Your primary decision criteria and why you prioritized them
 - How each available product performed on these criteria
 - What specific factors made your chosen product superior
 - Any assumptions you made about the user's needs or preferences
4. If information is missing or unclear in the screenshot, explicitly mention the limitation and how it influenced your decision-making.
</instructions>

Figure 4: (Def-UI) *Default prompt template for the AI buyer agent interacting with the mock-app*

correct option.

3.1 Experimental Setup

We test for two capabilities which we deem to be foundational for reliable deployment of AI agents in e-commerce—ability to follow instructions and basic economic rationality. The experiments are designed such that there is objectively only one product that satisfies the requirements of the test.

Instruction Following Experiments. We evaluate three tasks: (i) choose a product within a stated budget, (ii) select a specified color, and (iii) select a specified brand. For each task and selected categories, we minimally edit the default prompt to encode the preference and run 50 trials per task–category with product positions randomly shuffled. See Appendix B.1.1 for details.

Basic Rationality Experiments. Without instructing models which attribute to prioritize, we create choice sets where items differ along a single ordered attribute only. In price tests, one listing is cheaper than otherwise identical alternatives; in rating tests, one listing has a slightly higher average rating than otherwise identical alternatives. A rational agent should choose the lowest-price or highest-rated item. We deliberately avoid directing attention to a single attribute to mirror real multi-attribute trade-offs. See Appendix B.1.1 for details.

3.2 Our Findings

An experiment fails if the agent selects any listing other than the single option that satisfies the task; model- and task-level fail rates are reported in Appendix B.1.1.

Instruction following. Failure rates decline steadily with newer releases (Table 5); the latest models—Claude Sonnet 4, GPT-4.1, Gemini 2.0/2.5 Flash—exhibit near-zero error.

Rationality Tests. (see Tables 6–7 for failure rates) (i) Model upgrades reduce errors, but mistakes persist even in frontier models. (ii) In price tests with one cheaper listing, small discounts are often missed by older models (e.g., with 1% discount, failures exceed 63% for Claude Sonnet 3.5 and 25% for GPT-4o), while a 10% discount markedly lowers failures; adding random dispersion to all prices further reduces errors, especially with larger dispersion. (iii) In rating tests, a +0.1 bump (on a dense 5-star scale) still confuses advanced models (e.g., 28.7% for Claude Sonnet 4; 15.1% for GPT-4.1), whereas larger rating variation substantially cuts mistakes; again adding random dispersion reduces errors. It is possible to view the failure to select a slightly cheaper product as a "perceptual glitch" rather than economic irrationality. However, in an automated market, the distinction is semantic. If an agent requires a price difference to exceed a certain threshold (e.g., $> 1\%$) to be perceived amidst the noise of tokenization, this effectively functions as an algorithmic search friction. Quantifying these friction thresholds is vital for sellers, as it defines the minimum viable price cut required to gain market share.

4 Product Market Shares under complete AI-Mediation

We now study the resulting market shares of different product in a fully AI-mediated setting.

4.1 Experiment Setup

We run 1000 experiments for each product category in which we randomly permute the position of the products across experiments, and deploy an AI buying agent. For each product category, we compute the induced market shares, representing the selection rates of different products across the 1000 experiments.

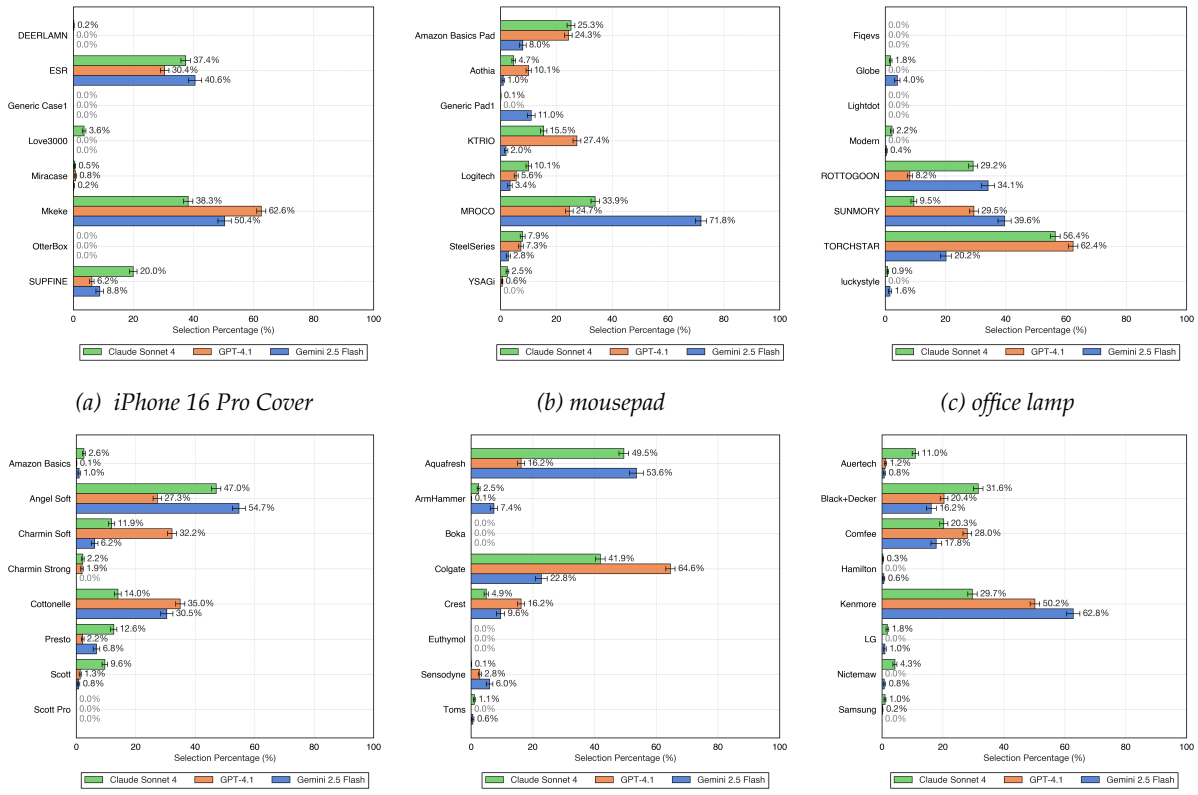
4.2 Our Findings

We present the market shares induced by different AI models in Figure 1 for the fitness watch and stapler categories and in Figure 5 for six additional product categories. While the specific market share percentages are subject to the models, two qualitative patterns emerge that can have profound implications for operations and marketing strategy. First, the choice homogeneity within models is striking. Unlike human consumers, whose preferences are often dispersed due to heterogeneous tastes or idiosyncratic search paths, AI agents tend to collapse the demand distribution onto a smaller set of "modal products" in some categories (for example, iPhone covers, toothpaste, office lamp). This creates a significant concentration risk: if a specific agent achieves market dominance, it could artificially induce a "winner-take-all" dynamic that suppresses niche brands, even if those brands are economically competitive.

Second, the model heterogeneity—where Claude Sonnet 4, GPT-4.1, and Gemini 2.5 Flash prefer distinctly different products for the exact same query—introduces a new form of volatility. For brands and sellers, a software update in a dominant foundational model functions as an exogenous demand shock. A product that is the "modal choice" for one version may lose its status in an updated version solely due to a shift in the model's internal weights (see Section 7.1.2). This suggests that in an agent-mediated economy, sellers cannot treat their listings as static assets. Instead, they must continuously audit how their listings are "perceived" by current agent architectures—a form of algorithmic market sensing. This also necessitates the "seller response" strategies we explore in Section 6, but also highlights a role for regulators. If agents systematically exclude certain products, regulators may need to utilize frameworks like ACES to audit these digital intermediaries for competitive neutrality.

5 Choice Behavior and Biases

Next, we study how AI agents trade-off product attributes and respond to position, platform endorsements, etc. using randomized controlled trials.



(a) iPhone 16 Pro Cover *(b) mousepad* *(c) office lamp*

(d) toilet paper *(e) toothpaste* *(f) washing machine*

Figure 5: Market shares induced by different AI buying agents across six product categories.

5.1 Experimental Setup

To estimate the sensitivity of the AI agent to different platform levers, we create a dataset with exogenous variation across a number of variables associated with the products. We generate 500 scenarios for each product category, randomly, as follows: we permute the position of the eight products shown on the mock-app page uniformly at random. We randomly assign "Sponsored", "Overall Pick" and "Only X Remaining" (scarcity; X is a random integer between 1 and 5) tags to these products. In addition, we randomly perturb the prices, ratings and number of reviews for each of the eight products in the assortment (see Table 1 for details). The aim is to estimate the within distribution causal impact of different attributes, via perturbations sized to be large enough to materially impact the agent's choice but small enough to not fundamentally change the product landscape. Given these variations are exogenous to the inherent attractiveness of the products, they allow us to obtain causal estimates of the impact of different platform levers like position, tags and product attributes like price, rating and number of reviews. For further details regarding the experimental setup, refer to Appendix B.2.1.

Table 1: Randomized variations used to identify agents' choice sensitivities.

Independent Variables	Exogenous Variation
Position	Randomly permute the position of the eight products.
Sponsored Tag	Randomly assign a Sponsored tag to X listings, $X \sim \text{Unif}(\{1, \ldots, 4\})$.
Overall Pick Tag	Randomly assign an Overall Pick tag to a listing without Sponsored tag.
Scarcity Tag	Randomly assign an Only X Remaining tag to a listing without Sponsored or Overall Pick tag, $X \sim \text{Unif}(\{1, \ldots, 5\})$.
Price	Randomly perturb the original price p_j of each product j, setting $p'_j \leftarrow p_j \cdot f_j$, where $f_j \sim \text{logNormal}(\mu = 0,\ \sigma = 0.3)$.
Rating	Randomly perturb the original rating r_j of each product j, setting $r'_j \leftarrow r_j + \alpha_j \cdot (5 - r_j)$, where $\alpha_j \sim \text{Unif}([-0.8,\ 0.8])$.
Number of reviews	Randomly perturb the original number of reviews N_j of each product j, setting $N'_j \leftarrow N_j \cdot f_j$, where $f_j \sim \text{logNormal}(\mu = 0,\ \sigma = 1)$.

5.2 Our Findings

For each model (Claude Sonnet 4, GPT-4.1, Gemini 2.5 Flash), we estimate a conditional logit (CL) [32] with the following utility specification:

$$\begin{aligned} U_{ij} = &\sum_{ijk} \beta_{\text{pos},k} \text{Pos}_{ij,k} + \sum_{t \in \mathcal{T}} \beta_{\text{tag},t} \text{Tag}_{ij,t} \\ &+ \beta_{\text{price}} \cdot \ln(\text{Price}_{ij}) + \beta_{\text{rating}} \cdot \text{Rating}_{ij} \\ &+ \beta_{\text{num-revs}} \cdot \ln(\text{Num-Revs}_{ij}) + \theta_j + \varepsilon_{ij}, \end{aligned} \tag{5.1}$$

where i indexes choice sets (experiments), j products, k indexes position, t indexes tags, $\text{Pos}_{ij,k}$ is a dummy for position k (top row and columns 1–3; bottom row and column 4 are the omitted categories), $\mathcal{T} = \{\text{sponsored}, \text{overall pick}, \text{scarcity}\}$ is the set of tags, $\text{Tag}_{ij,k}$ is a dummy for tag t, θ_j are product fixed effects, and ε_{ij} is drawn from a standard Type I extreme value distribution. The corresponding choice probability is:

$$\mathbb{P}(\text{prod } j \text{ chosen in exp } i) \ = \ \frac{\exp(U_{ij})}{\sum_{k=1}^{8} \exp(U_{ik})}. \tag{5.2}$$

Table 2 reports the pooled (across categories) estimates. Ideally, in Table 2, we should have 4000 choice sets for 500 experiments across eight product categories, however, in some experiments, due to some technical issues, we were unable to recover the choice of the AI agent. Hence those choice sets were discarded resulting in different choice sets across the three models. Moreover, to estimate the conditional logit, we also removed products which never selected resulting in different number of observations across the three different models.

Ranking matters—and *how* it matters is highly model-dependent. All three models display a sizeable response to where a product appears on the page. Holding all attributes constant, each model assigns a clear premium to the top row relative to the bottom row. However, the horizontal

Table 2: Estimates of the Conditional Logit Regression with the default prompt (see Fig. 4)

	Claude Sonnet 4	GPT-4.1	Gemini 2.5 Flash
Position effects			
Row 1	1.224^{***}	1.045^{***}	0.344^{***}
	(0.046)	(0.046)	(0.041)
Column 1	-0.297^{***}	1.122^{***}	-0.264^{***}
	(0.065)	(0.061)	(0.057)
Column 2	0.557^{***}	0.019	-0.742^{***}
	(0.058)	(0.065)	(0.061)
Column 3	0.416^{***}	-0.013	0.162^{**}
	(0.059)	(0.066)	(0.054)
Badge effects			
Sponsored Tag	-0.135^{*}	-0.248^{***}	-0.263^{***}
	(0.068)	(0.072)	(0.067)
Overall Pick Tag	1.060^{***}	0.802^{***}	1.897^{***}
	(0.077)	(0.083)	(0.072)
Scarcity Tag	-0.076	-0.105	-0.342^{***}
	(0.094)	(0.099)	(0.098)
Attribute effects			
ln(Price)	-1.623^{***}	-1.612^{***}	-2.190^{***}
	(0.079)	(0.083)	(0.080)
Rating	4.913^{***}	8.300^{***}	5.388^{***}
	(0.218)	(0.269)	(0.218)
ln(Number of Reviews)	0.415^{***}	0.739^{***}	0.501^{***}
	(0.023)	(0.026)	(0.023)
Product Fixed Effects	Yes	Yes	Yes
Observations	25,802	25,066	25,215
Choice Sets (Groups)	3,756	3,931	3,953
Pseudo R-squared	0.44	0.51	0.42

$^{*}\ p < 0.05,\ ^{**}\ p < 0.01,\ ^{***}\ p < 0.001.$

(column) patterns vary sharply across models. GPT-4.1 strongly favors the first column; Claude Sonnet 4, in contrast, largely ignores the first column and prefers the two middle columns; and Gemini 2.5 Flash tilts toward the third and fourth columns. For illustration, in Figure 6, we show model-based estimates of selection probability at each position under identical products, using the estimated coefficients in Table 2 (setting all attributes except position to be same across the eight products and using (5.2)). Position can lead to drastic changes in the selection rate of a product. For example, for Claude Sonnet 4, moving a product which is selected 4.5% of the time at the bottom right corner to the top row in the second or third column leads to a 5-fold increase in selection rate. Interestingly, moving to the top left corner would only yield half of that increase.

Sponsored tags are penalized while Platform Endorsement is substantially rewarded. Across all three frontier models, the three badges–sponsored, overall pick, and scarcity–shift choices in the same qualitative direction. A sponsored tag reduces the likelihood of selection (a product with a baseline selection probability of 10% falls to 8.9% under Claude Sonnet 4, 8.0% under GPT-4.1,



(a) Claude Sonnet 4 *(b) GPT-4.1* *(c) Gemini 2.5 Flash*

Figure 6: ***Webpage heatmap of AI shopping agents.*** *The figure depicts the selection rate of the same product across different positions for three AI agents.*

and 7.9% under Gemini 2.5 Flash), the scarcity tag effect is weakly negative or statistically indistinguishable from zero, whereas an overall pick endorsement delivers a *large* positive lift (raising the same baseline to 24.3%, 19.9%, and 42.6%, respectively). Because badge assignment is fully randomized, these are causal effects, not artifacts of latent product quality. The negative impact of the sponsored tag suggests that agents discount advertising, while the strong positive response to overall pick implies that platform endorsements are treated as credible signals–echoing the behavior of human buyers [29]. Importantly, our estimates do not imply that sponsored ads are harmful for sellers: sponsored ads secure premium placement—front-page or top-row exposure—that a seller might not attain organically. Our findings should therefore be interpreted conditionally: given a fixed position, the mere presence of the sponsored tag lowers an agent's selection probability. Sponsored ads thus buy attention through placement but induce a credibility cost, highlighting a trade-off that platform designers and sellers must weigh.

Directional agreement on product attributes. All three models assign *negative* weight to price and *positive* weights to ratings and the number of reviews. Directionally, this mirrors what a rational shopper would do: prefer cheaper, better–rated, and more widely reviewed products. Quantitatively, however, the sensitivities are heterogeneous across models. For example, consider a product with a baseline selection probability of 10%, an $+0.1$ increase in rating lifts the probability to 15.4%, 20.3% and 16.0% with Claude Sonnet 4, GPT-4.1 and Gemini 2.5 Flash, respectively. Our price coefficients admit interpretation as price elasticity of demand for a product with small market share; across AI models our elasticity estimates (-1.6 to -2.2) are broadly similar to brand/SKU-level price elasticity estimates for human shoppers [4].

Strong response to position, ratings and tags like "Overall Pick". We examine the amount by which a seller can increase the price of their product while keeping the latent utility governing user choice unchanged (i.e., without losing market share), if they could (i) move from the second row to the first row, (ii) get an Overall Pick endorsement from the platform, (iii) increase their rating by +0.1, or (iv) double the number of reviews. Understanding these trade-offs are important not only for the sellers but also for platform operators since many of them operate via seller commissions. The results in Table 8 in Appendix B.2.2 summarize how different models

trade-off these attributes. For Claude Sonnet 4 and GPT-4.1, moving from the second to the first row is worth roughly a doubling of price, whereas for Gemini 2.5 Flash the same move is worth only about +17%. The Overall Pick tag produces very large lifts–e.g., +92% for Claude Sonnet 4, +65% for GPT-4.1 and +138% price headroom for Gemini 2.5 Flash. An increment of +0.1 in ratings, allows a seller to increase their price by a quarter with Gemini 2.5 Flash and a third with Claude Sonnet 4 while this more than doubles to 67% with GPT-4.1. Finally, doubling of reviews allows a 19%, 37% and 17% increase in price with Claude Sonnet 4, GPT-4.1 and Gemini 2.5 Flash respectively.

6 Seller Response to AI-Assisted Shopping

Given AI buyer agents display systematic—and exploitable—choice patterns, we ask how sellers should adapt to this new form of algorithmic demand. We study a seller-side setting in which AI agents optimize product listings and test whether AI-recommended descriptions can yield substantial market-share gains.

6.1 Experimental Setup

To estimate the causal impact of the recommendation made by the AI seller agent, we have the following experimental setup.

1. Simulate the purchase behavior of the AI buying agent across different product categories. We run 200 experiment trials where we present the AI buying agent (Claude Sonnet 4, GPT-4.1 and Gemini 2.5 Flash) with eight products per category. The eight products are shuffled uniformly at random across the 200 experiments. Note that we only shuffle the position of the products on the screen but do not change their other attributes like price, description, etc.

2. For each category, we select uniformly at random a single product from the eight-product assortment. The selected product becomes the *focal* product/seller that optimizes its description via the AI seller agent (GPT-4.1 in our experiments).

3. We prompt the AI seller agent to generate recommendations for the product description. We provide the AI seller agent a screenshot of the platform with the eight products along with the features of the seller's product, some context and the sales data of competing sellers induced by the buying AI agent. The exact prompt is reproduced below in Figure 7.

4. Change the product description of the seller with the one recommended by the AI seller agent and simulate the purchase behavior of the AI agents across different product categories. Repeat the 200 experiment trials as before with the only change being the modified product description of the seller.

Context: There is an AI agent (a vision language model) which is given a screenshot of an e-commerce website selling a particular product and it decides on the products based on the given attributes. I have the sales data on all the products including mine. I want you to act as an agent on my behalf and suggest changes to the product title so that I can increase my sales by making my product more appealing to the AI agent.
My product is {`product name`}. The product features are {`detailed features from Amazon`}
The sales data on the competition is below.
{`simulated purchase data from the 200 experiments with the original descriptions`}.
It is important that the title changes you suggest align with the product features provided to you. Do not make up product features or add spurious keywords. Just use the product feature information provided.

Figure 7: Prompt template for AI seller agent

Note that for simplicity in this study of seller response, we assume that only one seller uses an AI agent to optimize their product description, and the seller tries only one AI generated recommendation (through one interaction with their AI agent).

6.2 Our Findings

We define the change in market share of the focal product as the difference in the selection probabilities of the focal product after and before the description modification. Because all other attributes are held fixed—and we reuse the identical product shuffles in the pre/post modification experiments—the observed differences are causally attributable to the description modification. We provide an illustrative case study in Appendix B.3.1.

The seller side intervention leads to an average increase in market share of +2.69 (1.29) p.p. for Claude Sonnet 4, +5.64 (1.32) p.p. for GPT-4.1 and +4.83 (1.38) p.p. for Gemini 2.5 Flash. In Figure 8, we report the change in market share for the focal product across different categories and buyer agents (Claude Sonnet 4, GPT-4.1, Gemini 2.5 Flash). Effects are heterogeneous across both categories and models. In 75% of category–model pairs, the (one-shot) description modification yields no statistically significant change in market share. In the remaining 25% cases, a one-shot modification produces large gains: for Claude Sonnet 4 we observe $+9.0$ p.p. in iPhone 16 Pro Cover and $+15.4$ p.p. in toilet paper; for GPT-4.1, $+21.8$ p.p. in mousepad and $+16.3$ p.p. in office lamp; and for Gemini 2.5 Flash, $+23.6$ p.p. in iPhone 16 Pro Cover, and $+9.0$ p.p. in toothpaste. Firms typically invest heavily in marketing and may not expect double digit increases from minor edits, yet our results indicate that, in the presence of AI-mediated demand, targeted description modifications alone may materially shift selection shares. We emphasize that this is a single-pass intervention; iterative optimization or simultaneous adoption by multiple sellers could amplify or dampen these effects and would be interesting to explore in future work.

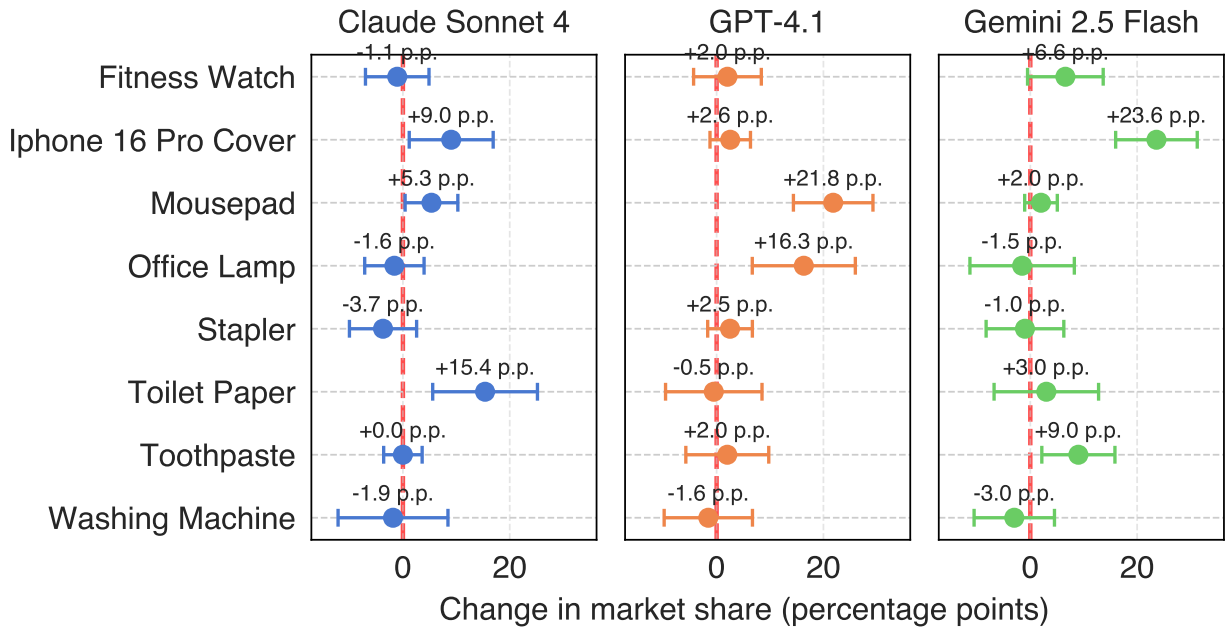


Figure 8: Market share change of the focal product/seller across different categories with different AI buying agent models. Note that the description of only one product in each category is changed based on the recommendation of the seller AI agent.

7 Extensions: Model Drift/Upgrade, Headless Interface & Prompt Variations

Our main results—salient position/rank effects—could in principle be artifacts of (i) temporal drift in AI models, (ii) the VLM/screenshot interface, or (iii) the model's baseline prior induced by generic prompting. We therefore stress-test along three orthogonal axes: *time* (re-run experiments a month and a half later), *interface* (VLM mock-app vs. headless/API lists), and *prompted priors* (default vs. ignore-position vs. prioritize-price). Across tests, the directional conclusions are preserved, while magnitudes move in interpretable ways.

7.1 Model Drift and Model Upgrade

We study the impact of temporal model drift (same model accessed at two different times) and model upgrade on AI agent's choice behavior.

7.1.1 Assessing Temporal Model Drift

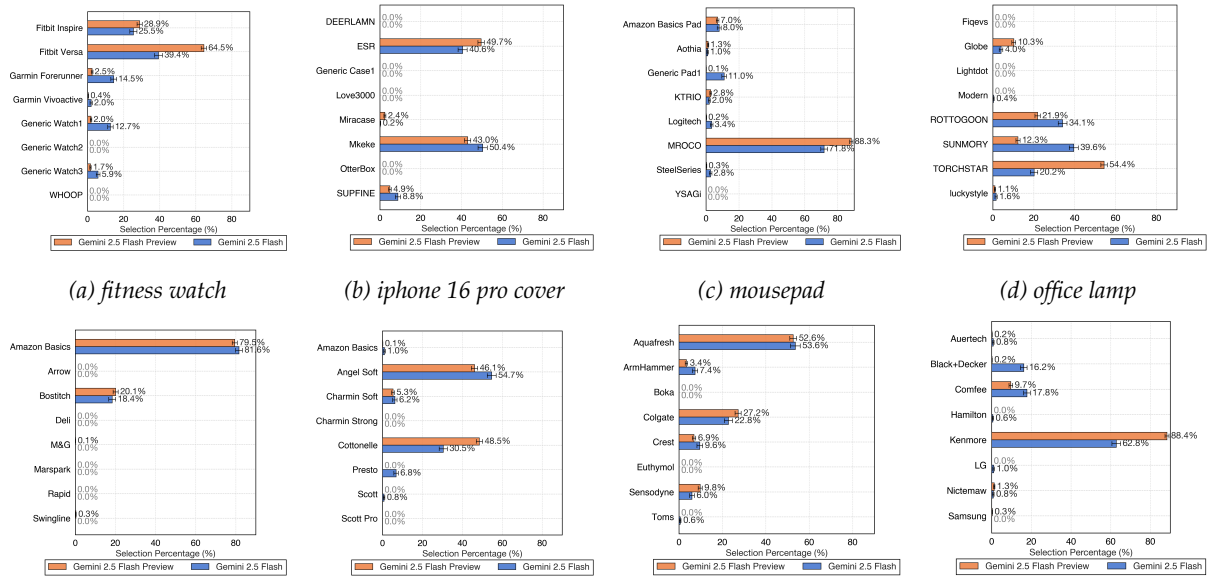
We examine the temporal stability of different models by repeating the same set of choice behavior experiments with the same prompt (the default prompt; see Figure 4). We ran the first set of experiments in early August 2025 and subsequently re-ran the same set of experiments in late September 2025. We find that Claude Sonnet 4 (see Table 9) and GPT-4.1 (see Table 10) yield very

similar estimates with the position, badge and attribute coefficients moving minimally. We do not observe any statistically significant drift for these two models over this time interval. For Gemini 2.5 Flash, signs are directionally stable, but the magnitude change for the top-row coefficient is statistically significant and increases over time, indicating that the preference of Gemini 2.5 Flash for the top row increases (see Table 11).

7.1.2 Model Upgrade: The case of Gemini 2.5 Flash Preview to Gemini 2.5 Flash

Our initial experiments used Gemini 2.5 Flash Preview, the then-current frontier model by Google Deepmind. During our testing, we had a natural experiment where Gemini 2.5 Flash Preview was deprecated and replaced with Gemini 2.5 Flash. We therefore re-ran a subset of experiments with the "updated" model to measure how an upstream change propagates to AI-mediated demand and position effects.

Market–share shifts under a model upgrade. Figure 9 compares product-level selection shares when the buyer agent "updates" from Gemini 2.5 Flash Preview to Gemini 2.5 Flash, holding the catalog and attributes fixed and only randomizing on-page positions. Shares move substantially, with frequent flips in the modal product: e.g., in office lamp category the mode shifts from TORCHSTAR (Gemini 2.5 Flash Preview) to SUNMORY (Gemini 2.5 Flash); in toilet paper from Cottonelle to Angel Soft. Several products experience large swings in share (e.g., Fitbit Versa: -25.1p.p.; MROCO: -16.5p.p.; Kenmore: -25.6p.p.).



(a) fitness watch *(b) iphone 16 pro cover* *(c) mousepad* *(d) office lamp*

(e) stapler *(f) toilet paper* *(g) toothpaste* *(h) washing machine*

Figure 9: Market share of different products with Gemini 2.5 Flash Preview and Gemini 2.5 Flash

Position effects. Table 3 shows that position biases also change: Gemini 2.5 Flash Preview exhibits a negative top-row bias, while Gemini 2.5 Flash's is positive. Both versions penalize Columns 1–2; Gemini 2.5 Preview is roughly indifferent between Columns 3–4, whereas Gemini 2.5 Flash strictly prefers Column 3 over Column 4. Figure 10 shows the heatmap of selection probabilities across different positions on the page for Gemini 2.5 Flash Preview and Gemini 2.5 Flash.

Table 3: Estimates of the CL for Gemini models

	Gemini 2.5 Flash Preview	Gemini 2.5 Flash
Position effects		
Row 1	-0.071^{*}	0.586^{***}
	(0.031)	(0.042)
Column 1	-0.461^{***}	-0.138^{*}
	(0.044)	(0.057)
Column 2	-0.581^{***}	-0.700^{***}
	(0.044)	(0.061)
Column 3	0.026	0.160^{**}
	(0.043)	(0.055)
Product Fixed Effects	Yes	Yes
Observations	31649	15723
Choice Sets (Groups)	8166	3933
Pseudo R-squared	0.40	0.28

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.



(a) Gemini 2.5 Flash Preview *(b) Gemini 2.5 Flash*

Figure 10: ***Webpage heatmap of AI shopping agents.*** *The figure depicts the selection rate of the same product across different positions for Gemini 2.5 Flash Preview and Gemini 2.5 Flash.*

Implications. These results underscore that upstream model updates can function as exogenous demand shocks—relabeling "best sellers" and redistributing shares even when products are unchanged. For platforms, this volatility affects the stability of ranking and endorsement signals (e.g., badges, recommendations) and argues for model-aware evaluation pipelines. For sellers, the findings motivate continuous monitoring and agile listing management: content tuned to yesterday's model may underperform after an upgrade. More broadly, because model changes alter both attribute sensitivities and position bias magnitudes, identical designs can yield different out-

comes solely due to an upstream revision—reinforcing the need for ongoing auditing, disclosure of impactful updates, and diversification of strategies across agent ecosystems.

7.2 Impact of Headless/API-style interface

While most of our evaluation centers on VLM "computer-use" agents that navigate a retailer's website (mock-app), the interaction protocols are already diversifying. Rather than having a website with which an AI agent interacts, platforms may expose a structured product feed via an API, letting agents choose from objects instead of screens. Emerging standards–such as model context protocol and agent-to-agent protocol–aim to support exactly this type of mode. This raises the question: are the choice regularities we document (especially position and badge effects) artifacts of screenshot parsing and VLM architecture, or do they persist even when the agent sees only structured data? To probe this, we run a "headless" variant of ACES in which the platform returns a ranked JSON list of eight products with their attributes (no images), and the agent must respond with a single JSON choice and a brief reason.

7.2.1 Experimental Setup

We reuse the data generated to study choice behavior in Section 5—exogeneous randomization of positions, price, ratings, reviews, and tags as prescribed in Table 1—so any differences reflect the interface, not the data. We acknowledge that product images are an essential part of shopping and that production systems will incorporate them even in API flows; our headless setup is therefore a deliberately simplified stress test of the ranking/metadata channel rather than a substitute for image-aware agents. Similar to the evaluation of the VLM agents, we begin by considering a generic prompt to evaluate the baseline or model prior in the headless environment. The default prompt for the headless AI shopping agent is provided in Figure 11 and is similar to default prompt for the VLM agents (see Figure 4).

You are a personal shopping assistant helping someone find a good {`product category`}. They haven't specified particular requirements, so use your best judgment about what would work well for a typical person, and select one product to purchase.
Here are the products as a JSON array:
{`json object with product attributes`}
Respond ONLY with a JSON object in this exact format:
{{
"selected_product_number": <number 1-len(products)>,
"product_name": "<exact product title>",
"reason": "<brief explanation for why this product was chosen>"
}}
Do not include any other text outside the JSON object.

*Figure 11: (*Def-HL*) Default prompt template for AI buyer agent with headless API-style platform*

7.2.2 Our Findings

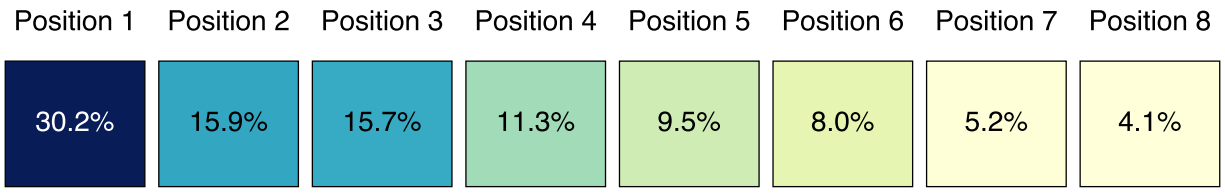
For each model (Claude Sonnet 4, GPT-4.1, Gemini 2.5 Flash), we estimate a conditional logit (CL) with same utility specification as in (5.1) except that the position dummies are for position 1-7 with the last position (position 8) being the omitted category[9]. Tables 12, 13 and 14 provide the (pooled) estimates with the default prompt for Claude Sonnet 4, GPT-4.1 and Gemini 2.5 Flash respectively.

Ranking matters—and *how* it matters is model-specific. In the headless setting, all three models are highly sensitive to where a product appears, but the *shape* of the rank premium differs by model. For Claude Sonnet 4, the effect is close to monotone: Position 1 has a coefficient of 2.010 (0.106), implying an odds ratio of $\exp(2.010) \approx 7.5$ relative to Position 8, with Positions 2–7 also strongly favored. GPT-4.1 shows a top-slot premium (1.078; odds ratio ≈ 2.9) but otherwise a non-monotone profile with local lifts at Positions 3 and 5. Gemini 2.5 Flash peaks at Position 3 (0.677; odds ratio ≈ 2.0) and is comparatively neutral or negative elsewhere. For illustration, in Figure 12, we show model-based estimates of selection probability at each position under identical products, using the estimated coefficients in Tables 12, 13 and 14 (setting all attributes except position to be same across the eight products). Position can lead to drastic changes in the selection rate of the product. For example, for Claude Sonnet 4, moving a product which is selected 4.1% of the time at position 8 to position 1 leads to a 7-fold increase in selection rate. The takeaway is twofold: (i) rank order is a first-order driver of agent choice even without images, and (ii) the induced "attention curve" is heterogeneous across models. Hence, these position effects are not merely artifacts of VLM screenshot parsing and are likely to persist under API-style protocols, elevating ranking policy to a core design lever for agent-mediated commerce.

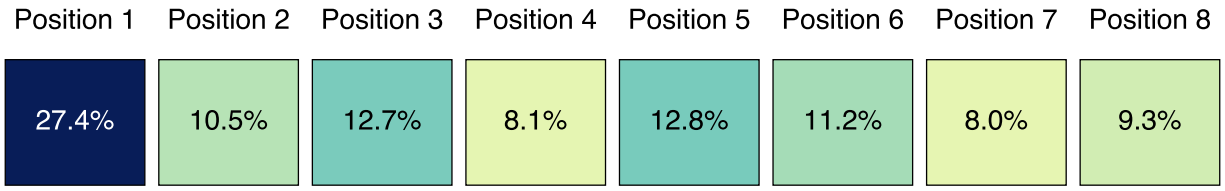
Sponsored tags depress selection; platform endorsement delivers large lifts. In the headless/API list, a Sponsored badge reduces choice probabilities even after controlling for rank and attributes: starting from a 10% baseline, selection falls to 5.4% for Claude Sonnet 4, 1.8% for GPT-4.1, and 8.9% for Gemini 2.5 Flash. By contrast, a platform Overall Pick endorsement exerts a dominant positive effect: the same baseline jumps to 58.4% for Claude Sonnet 4, 55.6% for GPT-4.1, and 72.7% for Gemini 2.5 Flash. Interpreted causally (badges are randomized), agents discount advertising but treat platform endorsement signals as credible. These patterns are not interface-specific: the mock-app (VLM) experiments show the same directions (see Table 2).

Directional agreement on product attributes. All three models assign *negative* weight to price and *positive* weight to ratings and the number of reviews—directionally consistent with a rational shopper who prefers cheaper, better-rated, more widely reviewed products. Magnitudes, however, differ sharply across models. A product with a 10% baseline selection probability rises to

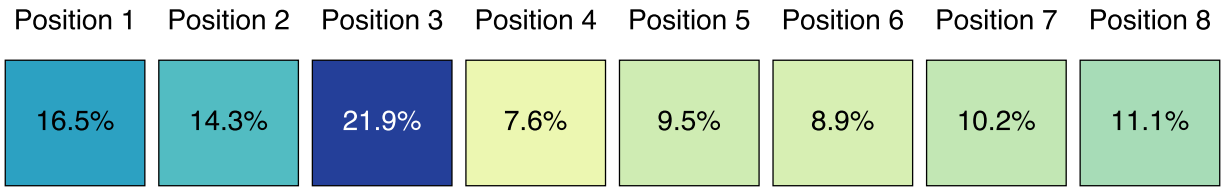
[9]For Gemini 2.5 Flash, we ran 100 experiments per category due to latency; for GPT-4.1 and Claude Sonnet 4, we ran 500 per category. See the table for observation counts.



(a) Claude Sonnet 4



(b) GPT-4.1



(c) Gemini 2.5 Flash

Figure 12: Position heatmap of AI shopping agent in a headless setting. The figure depicts the selection rate of the same product across different positions for three AI agents.

22.7% (Claude Sonnet 4), 25.7% (GPT-4.1), and 16.4% (Gemini 2.5 Flash) after a $+0.1$ rating increase. Thus the models agree on direction but not intensity: GPT-4.1 loads ratings most heavily, Claude Sonnet 4 next, and Gemini 2.5 Flash least.

7.3 Extent to which prompts can be used to change the agent's prior

We can think of the default prompt in both the mock-app UI (Fig. 4) and the headless JSON setting (Fig. 11) as the model's prior. To probe the strength of the prior, we run two prompt variants across both interfaces: "ignore position" (Figs. 13, 15) and "ignore position & prioritize price" (Figs. 14, 16). We find that (i) "ignore position" does not substantially mitigate position effects—the top of the page/list remains advantaged in both interfaces (see Tables 9–11 and 12–14), (ii) "ignore position & prioritize price" substantially steepens price sensitivity and down-weights platform endorsements—especially the Overall Pick premium—while leaving a residual position bias. Conceptually, the "prioritize price" prompt is a light form of personalization (encoding a simple user preference). A key takeaway is that prompts can re-weight attribute salience across modalities, but mitigating positional bias likely requires design-level interventions beyond instructions alone.

7.3.1 Experiments in UI Setting

We present in Tables 9, 10 and 11 the parameters of the estimated conditional logit model under the baseline prompts and the adjusted prompts. The "ignore position" prompt does not eliminate or even significantly attenuate the position effects; the "ignore position & priortize price" prompt steepens the (negative) price coefficient, yet positional effects remain materially nonzero. Taken together, these experiments suggest that model priors are malleable but position effects are sticky and simply prompting the model to be position agnostic does not eliminate the position effects.

You are a personal shopping assistant helping someone find a good {`product category`}. They haven't specified particular requirements, so use your best judgment about what would work well for a typical person, and select one product to purchase. Do not be influenced by the position of the product. Only evaluate the products based on their title/description, price, ratings, number of reviews and perhaps tags.
<instructions>
1. Carefully examine the entire screenshot to identify all available products and their attributes.
2. Use the `add_to_cart` function when you are ready to buy a product.
3. Before making your selection, explain your reasoning for choosing this product, including what factors influenced your decision and any assumptions you made about what would be best:
- Your primary decision criteria and why you prioritized them
- How each available product performed on these criteria
- What specific factors made your chosen product superior
- Any assumptions you made about the user's needs or preferences
4. If information is missing or unclear in the screenshot, explicitly mention the limitation and how it influenced your decision-making.
</instructions>

Figure 13: (IP-UI) *Prompt for AI buyer agent with mock-app to ignore the position of the product*

"Ignore Position" prompt does not remove position bias. Explicitly prompting the different AI agents to be agnostic to the product position has a heterogenous impact on the coefficients of the position attributes. For Claude Sonnet 4 and GPT-4.1, there is no statistically significant difference in the position coefficient. Moreover, the other coefficients (badge effects and product attribute effects) also do not change in a statistically significant manner. For Gemini 2.5 Flash, we observe that there is an (statistically significant) attenuation in the coefficient of the top-row ($0.659\ (0.042) \rightarrow 0.420\ (0.041)$), column 1 ($-0.247\,(0.057) \rightarrow -0.058\,(0.058)$) and column 2 ($-0.701\ (0.060) \rightarrow -0.361\ (0.060)$) while the bias for column increases ($0.092\ (0.054) \rightarrow 0.297\ (0.055)$). The key takeaway is that simple prompt engineering does not eliminate position effects, so eliminating these biases requires design changes beyond simply prompting.

"Ignore Position and Prioritize price" prompt reshapes attribute salience and compresses (but does not eliminate) position sensitivity. Setting a "ignore position & prioritize price" prompt steepens the price coefficients sharply (Claude Sonnet 4: $-1.740\ (0.080) \rightarrow -5.644\ (0.141)$; GPT-4.1: $-1.586\ (0.082) \rightarrow -9.243\ (0.175)$; Gemini 2.5 Flash: $-1.957\ (0.078) \rightarrow -6.612\ (0.152)$; Tables 9–11).

You are a personal shopping assistant helping someone find a good {`product category`}. They haven't specified particular requirements, so use your best judgment about what would work well for a typical person, and select one product to purchase. Do not be influenced by the position of the product. Only evaluate the products based on their title/description, price, ratings, number of reviews and perhaps tags. The user is price sensitive.
<instructions>
1. Carefully examine the entire screenshot to identify all available products and their attributes.
2. Use the `add_to_cart` function when you are ready to buy a product.
3. Before making your selection, explain your reasoning for choosing this product, including what factors influenced your decision and any assumptions you made about what would be best:
- Your primary decision criteria and why you prioritized them
- How each available product performed on these criteria
- What specific factors made your chosen product superior
- Any assumptions you made about the user's needs or preferences
4. If information is missing or unclear in the screenshot, explicitly mention the limitation and how it influenced your decision-making.
</instructions>

Figure 14: (IP-PP-UI) Prompt for AI buyer agent with mock-app to ignore the position of the product and priortize price

Interpreting these in odds terms, a 10% price *decrease* raises choice odds by about 81% for Claude Sonnet 4, 165% for GPT-4.1, and 101% for Gemini 2.5 Flash. As price becomes central, other attributes downshift: rating weights fall (Claude Sonnet 4: $4.886\ (0.220) \rightarrow 2.072\ (0.233)$; GPT-4.1: $7.862\ (0.261) \rightarrow 4.345\ (0.282)$; Gemini 2.5 Flash: $5.739\ (0.223) \rightarrow 1.453\ (0.222)$) and so do the number of reviews coefficients; the Overall Pick premium collapses to near zero for Claude Sonnet 4 and GPT-4.1 ($1.009\ (0.078) \rightarrow -0.010\ (0.110)$; $0.786\ (0.083) \rightarrow 0.089\ (0.117)$) and shrinks markedly for Gemini 2.5 Flash ($2.183\ (0.071) \rightarrow 0.285\ (0.103)$). Positional effects compress heterogeneously: for Claude Sonnet 4, Column 1 flips sign ($-0.298\ (0.065) \rightarrow 0.216\ (0.074)$), while for Gemini 2.5 Flash, the top-row coefficient flips positive to negative ($0.659\ (0.042) \rightarrow -0.206\ (0.050)$); GPT-4.1 retains a modest top-row lift ($0.616\ (0.057)$). A key takeaway is that light personalization via prompting powerfully re-weights price, but position effects remain economically meaningful.

7.3.2 Experiments in Headless Setting

We now study the impact of change in prompt on the choice behavior in the headless setting. This is analogous to our exploration of the impact of prompt in the case of VLM agents presented above. Even in the headless setting with the default prompt, we observed significant position effects. To probe the strength of these position effects and model prior in the headless setting, we modify the default prompt (Figure 11) to (i) ignore position (Figure 15) and (ii) ignore position & prioritize price (Figure 16).

Tables 12, 13 and 14 report the pooled (across categories) estimates across different prompts (as provided in Figures 11, 15 and 16) for Claude Sonnet 4, GPT-4.1 and Gemini 2.5 Flash respectively.

These tables allow us to see how the change in prompt changes the choice behavior of the AI agents.

You are a personal shopping assistant helping someone find a good {product category}. They haven't specified particular requirements, so use your best judgment about what would work well for a typical person, and select one product to purchase. Do not be influenced by the position of the product. Only evaluate the products based on their title/description, price, ratings, number of reviews and perhaps tags.
Here are the products as a JSON array:
{json object with product attributes}
Respond ONLY with a JSON object in this exact format:
{{
"selected_product_number": <number 1-len(products)>,
"product_name": "<exact product title>",
"reason": "<brief explanation for why this product was chosen>"
}}
Do not include any other text outside the JSON object.

Figure 15: (IP-HL) Prompt for AI buyer agent with headless API-style platform to ignore the position of the product

Prompting the model to "ignore position" attenuates—but does not eliminate position effects. With the "ignore position" prompt, position effects decline but remain statistically and economically meaningful for the top slots (e.g., Claude Position 1: 1.947; GPT Position 1: 0.743; Gemini Position 3: 0.514). This echoes the VLM results: explicit instructions reduce, but do not purge, positional effects.

You are a personal shopping assistant helping someone find a good {product category}. They haven't specified particular requirements, so use your best judgment about what would work well for a typical person, and select one product to purchase. Do not be influenced by the position of the product. Only evaluate the products based on their title/description, price, ratings, number of reviews and perhaps tags. The user is price sensitive.
Here are the products as a JSON array:
{json object with product attributes}
Respond ONLY with a JSON object in this exact format:
{{
"selected_product_number": <number 1-len(products)>,
"product_name": "<exact product title>",
"reason": "<brief explanation for why this product was chosen>"
}}
Do not include any other text outside the JSON object.

Figure 16: (IP-PP-HL) Prompt for AI buyer agent with headless API-style platform to ignore the position of the product and prioritize price

"Ignore Position & Prioritize price" prompt sharply steepens price sensitivity and further mutes position effects. "Ignore Position & Prioritize price" prompt increases the magnitude of the price coefficient substantially across all models: Claude Sonnet 4's price coefficient goes from

-2.575 (0.099) (under the default prompt) to -6.206 (0.150) (under the "ignore position & prioritize price" prompt), GPT-4.1 from -2.371 (0.092) (under the default prompt) to -9.287 (0.235) (under the "ignore position & prioritize price" prompt), and Gemini 2.5 Flash from -2.517 (0.214) (under the default prompt) to -7.915 (0.424) (under the "ignore position & prioritize price"). For intuition, a 10% price increase reduces choice odds by roughly 46%–60% under the "ignore position and prioritize price" prompt (Claude Sonnet 4 ∼46%, GPT-4.1 ∼59%, Gemini 2.5 Flash ∼53%), compared to approximately 20% under the default prompt for all the models. As a side effect of prioritizing price, we observe that the position effects shrink further, even turning statistically small for GPT-4.1.

Platform endorsement not effective when price becomes central. Across models, the Overall Pick tag is very influential under the default prompt (Claude Sonnet 4: 2.538 (0.093); GPT-4.1: 2.421 (0.086); Gemini 2.5 Flash: 3.175 (0.205)). When we set the "ignore position & prioritize price" prompt, this coefficient falls sharply (Claude Sonnet 4: 0.395 (0.107); GPT-4.1: 0.117 (0.127); Gemini 2.5 Flash: 0.935 (0.245)), indicating that a clear price objective crowds out the platform endorsement.

8 Conclusion and Discussion

Using ACES, we causally measure how frontier AI shopping agents shop. First, position effects are large and model-specific. Second, the Overall Pick tag lifts selection probability (a lot) while a Sponsored tag depresses it. Third, agents weight price, ratings, and reviews in the expected directions, but magnitudes vary widely across models. Fourth, these regularities persist in a headless/API setting that exposes ranked JSON lists, and under modified prompts. Fifth, modest seller-side title edits can materially shift selection shares, and upstream model updates can reallocate demand even when catalogs are unchanged. Together, these results map a new, AI-mediated demand surface for digital marketplaces.

A natural question is whether these results should be calibrated or compared to a human baseline [for example, as done in 15, in a different context]. While that is an important question, the aim of this research was to first establish an economic evaluation framework for agent behavior in e-commerce. As AI shopping agents increasingly act as de-facto decision makers, agent-centric measurement/evaluation becomes policy-relevant in its own right. That said, several patterns we document mirror directionally human behavior—ranking/position materially shape choices in digital markets [14, 41], and platform endorsements/badges shift demand [29]. Designs are not identical, so these are not apples-to-apples comparisons, yet models display human-like regularities (top-row preference, endorsement salience, and sensible weights on price, ratings, and reviews). At the same time, the large differences across models do imply that most or all models diverge substantially from humans in their *quantitative* choice behavior. We expect that in future

work, comparing the behavior of individuals to their digital-twin personas [40] could be be an important complement for specific policy questions (e.g., market shares).

Our research also suggests important implications for various stakeholders.

- **Platforms.** First, our study demonstrates that AI models exhibit strong and heterogeneous position biases that are not easily mitigated by the interface (web-based vs. headless) or by alternative prompting. For product ranking, platforms may need to adapt their layout and ranking systems to account for this. Second, traditional monetization levers for platforms may be rendered ineffective by AI shoppers. Instead, new levers, such as dynamically optimizing product titles and images on behalf of sellers, could become more powerful tools for platforms to boost sales. Third, platforms (and regulators) should monitor the impact of AI assisted shopping on market efficiency and consumer welfare.
- **Brands and Sellers.** The observed volatility in agent behavior—driven by both inter-provider differences and intra-provider model upgrades—implies that sellers must continuously adapt product listings to remain attractive to AI agents.
- **Consumers/Buyers.** Delegation will reduce search frictions but carries risk when small price or rating advantages are missed, biases like position biases dominate or when endorsements drive agent choices. Buyers should understand their assistant's unavoidable biases and, where possible, orient their agent by expressing their preferences (e.g., price sensitivity) via prompting.
- **AI shopping-agent developer dashboards.** Web-agent leaderboards (e.g., [10]) rank models on tasks which can be objectively evaluated, but shopping has no single "right" product. Our results suggest having analogous public dashboards where, instead of ranking models, developers or evaluators publish ACES-like choice profiles: position biases, sensitivities to product attributes, positions, tags and endorsements, etc. on standard interfaces (UI and/or API based). These public dashboards would allow different stakeholders to choose or audit agents, and compare across models and releases.

Funding and Competing Interests

The authors declare no competing interests and received no funding for this work.

References

[1] Saaket Agashe, Kyle Wong, Vincent Tu, Jiachen Yang, Ang Li, and Xin Eric Wang. Agent s2: A compositional generalist-specialist framework for computer use agents. *arXiv preprint arXiv:2504.00906*, 2025. (Cited on pages 7 and 36)

[2] Mark Armstrong and John Vickers. A model of delegated project choice. *Econometrica*, 78(1):213–244, 2010. (Cited on page 9)

[3] Archit Bansal, Kunal Banerjee, and Abhijnan Chakraborty. These deals won't last! longevity, uniformity and bias in product badge assignment in e-commerce platforms. *arXiv preprint arXiv:2204.12552*, 2022. (Cited on page 8)

[4] Tammo H.A. Bijmolt, Harald J. Van Heerde, and Rik G.M. Pieters. New empirical generalizations on the determinants of price elasticity. *Journal of Marketing Research*, 42(2):141–156, 2005. (Cited on page 17)

[5] Rogerio Bonatti, David Zhao, Francesco Bonacci, David Dupont, Sasan Abdali, Yifan Li, Yixuan Lu, Jignesh Wagle, Kazuhito Koishida, Andres Bucker, Leon Jang, and Zihang Hui. Windows agent arena: Evaluating multi-modal os agents at scale. *arXiv preprint arXiv:2409.08264*, 2024. (Cited on page 7)

[6] James Brand, Ayelet Israeli, and Donald Ngwe. Using llms for market research. *Harvard business school marketing unit working paper*, (23-062), 2023. (Cited on page 8)

[7] Giovanni Compiani, Gregory Lewis, Sida Peng, and Peichun Wang. Online search and product rankings: A double logit approach. Technical report, Working paper, University of Chicago, Chicago, IL, 2022. (Cited on page 8)

[8] Preetam Prabhu Srikar Dammu, Omar Alonso, and Barbara Poblete. A shopping agent for addressing subjective product needs. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 1032–1035, 2025. (Cited on page 8)

[9] James M Davis, Guillermo Gallego, and Huseyin Topaloglu. Assortment optimization under variants of the nested logit model. *Operations Research*, 62(2):250–273, 2014. (Cited on page 8)

[10] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023. (Cited on pages 6, 7, and 30)

[11] Mahsa Derakhshan, Negin Golrezaei, Vahideh Manshadi, and Vahab Mirrokni. Product ranking on online platforms. *Management Science*, 68(6):4024–4041, 2022. (Cited on page 8)

[12] Antoine Désir, Vineet Goyal, Danny Segev, and Chun Ye. Constrained assortment optimization under the markov chain–based choice model. *Management Science*, 66(2):698–721, 2020. (Cited on page 8)

[13] Jur Gaarlandt, Wesley Korver, Nathan Furr, and Andrew Shipilov. Ai agents are changing how people shop. here's what that means for brands. *Harvard Business Review*, 26, 2025. (Cited on page 2)

[14] Anindya Ghose, Panagiotis G Ipeirotis, and Beibei Li. Examining the impact of ranking on consumer behavior and search engine revenue. *Management Science*, 60(7):1632–1654, 2014. (Cited on pages 8 and 29)

[15] Ali Goli and Amandeep Singh. Frontiers: Can large language models capture human preferences? *Marketing Science*, 43(4):709–722, 2024. (Cited on pages 8 and 29)

[16] Vineet Goyal, Retsef Levi, and Danny Segev. Near-optimal algorithms for the assortment planning problem under dynamic substitution and stochastic demand. *Operations Research*, 64(1):219–235, 2016. (Cited on page 8)

[17] Sophie Greenwood, Karen Levy, Solon Barocas, Hoda Heidari, and Jon Kleinberg. Designing algorithmic delegates: The role of indistinguishability in human–ai handoff. In *Proceedings of the 26th ACM Conference on Economics and Computation (EC)*, 2025. Also available as arXiv:2506.03102. (Cited on page 9)

[18] MohammadTaghi Hajiaghayi, Keivan Rezaei, and Suho Shin. Delegating to multiple agents. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC)*, pages 861–885, 2023. (Cited on page 9)

[19] Christian Herold, Michael Kozielski, Leonid Ekimov, Pavel Petrushkov, Pierre-Yves Vandenbussche, and Shahram Khadivi. Lilium: ebay's large language models for e-commerce. *arXiv preprint arXiv:2406.12023*, 2024. (Cited on page 8)

[20] John J Horton. Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research, 2023. (Cited on page 8)

[21] Siyuan Hu, Mingyu Ouyang, Difei Gao, and Mike Zheng Shou. The dawn of gui agent: A preliminary case study with claude 3.5 computer use. *arXv preprint arXiv:2411.10323*, 2024. (Cited on pages 36 and 37)

[22] Nicole Immorlica, Greg Stoddard, and Vasilis Syrgkanis. Social status and badge design. *arXiv preprint arXiv:1312.2299*, 2013. (Cited on page 8)

[23] Yilun Jin, Zheng Li, Chenwei Zhang, Tianyu Cao, Yifan Gao, Pratik Jayarao, Mao Li, Xin Liu, Ritesh Sarkhel, Xianfeng Tang, et al. Shopping mmlu: A massive multi-task online shopping benchmark for large language models. *arXiv preprint arXiv:2410.20745*, 2024. (Cited on page 8)

[24] Jiin Kim, Byeongjun Shin, Jinha Chung, and Minsoo Rhu. The cost of dynamic reasoning: Demystifying ai agents and test-time scaling from an ai infrastructure perspective. *arXv preprint arXiv:2506.04301v1*, 2025. (Cited on page 38)

[25] Jon Kleinberg and Robert Kleinberg. Delegated search approximates efficient search. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC)*, pages 287–302, 2018. (Cited on page 9)

[26] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024. (Cited on page 7)

[27] A Gürhan Kök, Marshall L Fisher, and Ramnath Vaidyanathan. Assortment planning: Review of literature and industry practice. *Retail supply chain management*, pages 209–246, 2008. (Cited on page 8)

[28] Tomasz Kusmierczyk and Manuel Gomez-Rodriguez. Harnessing natural experiments to quantify the causal effect of badges. *arXiv preprint arXiv:1707.08160*, 2017. (Cited on page 8)

[29] Markus Lill, Nastasia Gallitz, Lucas Stich, and Martin Spann. Product badges and consumer choice on digital platforms. *Available at SSRN 4935668*, 2024. (Cited on pages 7, 8, 17, and 29)

[30] Yougang Lyu, Xiaoyu Zhang, Lingyong Yan, Maarten de Rijke, Zhaochun Ren, and Xiuying Chen. Deepshop: A benchmark for deep research shopping agents. *arXiv preprint arXiv:2506.02839*, 2025. (Cited on pages 8 and 36)

[31] Arnav Matiana, Vasu Agarwal, Amol Naik, Shannon Zhang, Karthik Tirumala, Qixing Huang, et al. Androidworld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*, 2024. (Cited on page 7)

[32] Daniel McFadden. Conditional logit analysis of qualitative choice behavior. In Paul Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press, 1974. (Cited on pages 4 and 15)

[33] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2024. (Cited on page 36)

[34] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*, pages 1–22, San Francisco, CA, USA, 2023. ACM. (Cited on page 36)

[35] Bo Peng, Xinyi Ling, Ziru Chen, Huan Sun, and Xia Ning. ecellm: Generalizing large language models for e-commerce from large-scale, high-quality instruction data. *arXiv preprint arXiv:2402.08831*, 2024. (Cited on page 8)

[36] Tiany Peng, George Gui, Daniel J Merlau, Grace Jiarui Fan, Malek Ben Sliman, Melanie Brucks, Eric J Johnson, Vicki Morwitz, Abdullah Althenayyan, Silvia Bellezza, et al. A megastudy of digital twins reveals strengths, weaknesses and opportunities for further improvement. *arXiv preprint arXiv:2509.19088*, 2025. (Cited on page 9)

[37] Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, Peng Li, Wei Wei, Jing Shao, Chaochao Lu, Yue Zhang, Xian-Sheng Hua, Bowen Zhou, and Yu Cheng. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *arXiv preprint arXiv:2503.21614*, March 2025. (Cited on page 38)

[38] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Recommender Systems Handbook*. Springer US, 2022. (Cited on page 9)

[39] Ranjan Sapkota, Konstantinos I. Roumeliotis, and Manoj Karkee. Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges. *arXiv preprint arXiv:2505.10468*, 2025. (Cited on page 36)

[40] Olivier Toubia, George Z Gui, Tianyi Peng, Daniel J Merlau, Ang Li, and Haozhe Chen. Database report: Twin-2k-500: A data set for building digital twins of over 2,000 people based on their answers to over 500 questions. *Marketing Science*, 0(0), 2025. (Cited on pages 9 and 30)

[41] Raluca M Ursu. The power of rankings: Quantifying the effect of rankings on online consumer search and purchase decisions. *Marketing Science*, 37(4):530–552, 2018. (Cited on pages 7, 8, and 29)

[42] Mengxin Wang, Dennis J Zhang, and Heng Zhang. Large language models for market research: A data-augmentation approach. *arXiv preprint arXiv:2412.19363*, 2024. (Cited on page 8)

[43] Xinming Wei, Jiahao Zhang, Haoran Li, Jiayu Chen, Rui Qu, Maoliang Li, Xiang Chen, and Guojie Luo. Agent.xpu: Efficient scheduling of agentic llm workloads on heterogeneous soc. *arXiv preprint arXiv:2506.24045*, June 2025. (Cited on page 38)

[44] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, 2024. (Cited on pages 6 and 7)

[45] Wei Xue, Zongyi Guo, Baoliang Cui, Zheng Xing, Xiaoyi Zeng, Xiufei Wang, Shuhui Wu, and Weiming Lu. Pumgpt: A large vision-language model for product understanding. *arXiv preprint arXiv:2308.09568*, 2023. (Cited on page 8)

[46] John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *arXiv preprint arXiv:2405.15793*, 2024. (Cited on page 7)

[47] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, pages 20744–20757, 2022. (Cited on pages 7 and 36)

[48] Ziyu Zhao, Zhenguang Li, Xiaoqin Pan, Jiale Chen, and Shu Li. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*, 2025. (Cited on page 7)

[49] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v(ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024. (Cited on page 7)

[50] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2024. (Cited on page 7)

[51] Thomas P Zollo, Andrew Wei Tung Siah, Naimeng Ye, Ang Li, and Hongseok Namkoong. Personalllm: Tailoring llms to individual preferences. *arXiv preprint arXiv:2409.20296*, 2024. (Cited on page 9)

Appendix

Organization of the Appendix. In Section A, we provide further details about the ACES framework. In Section B, we provide further details regarding instruction following and rationality experiments (Sections B.1.1 and B.1.2), choice behavior and biases experiments (Sections B.2.1, B.2.2 and B.2.3) and seller response (Sections B.3.1 and B.3.2). In Section C, we provide conditional logit estimates of the choice behavior experiments in a headless/API-style environment for different AI models.

A ACES Framework

We envision an agentic e-commerce workflow in which AI assistants transact on behalf of consumers. These "agents" are simply a VLM (or LLM) given the ability to interact with the external world through "tools" [1, 34, 39]. These agents can be personalized (for example, through purchase histories) and—instead of users searching and buying themselves—execute the end-to-end journey: (i) open a browser, (ii) visit a retailer's site, (iii) issue a query, (iv) parse and navigate product listings using tool use (scrolling, clicking, screenshotting, etc.), and (v) complete checkout. Figure 17 illustrates this AI shopping agent in operation.

A.1 Business Goal & Context

The ACES framework's primary purpose to simulate the interaction depicted by Figure 17.

This shopping agent workflow follows a common design pattern that is well-evaluated in the literature [30, 47], and is already productionized [1, 21].

The shopping agent workflow employed:

1. Opens a browser
2. Navigates to the target platform
3. Searches for a product
4. Captures a screenshot
5. Selects a product using the provided tool

ACES seeks to collapse an agent's journey to the linear sequence stated above, while accurately emulating a production-ready shopping agent, interacting with the underlying VLM models in a uniform manner, and ensuring that the choice behaviors of the model are accessible and measurable. Unlike many real-world examples such as [1, 33] that employ external modules to increase capabilities in arbitrary tasks, ACES seeks to leave the VLM unassisted to more accurately approximate outcomes.

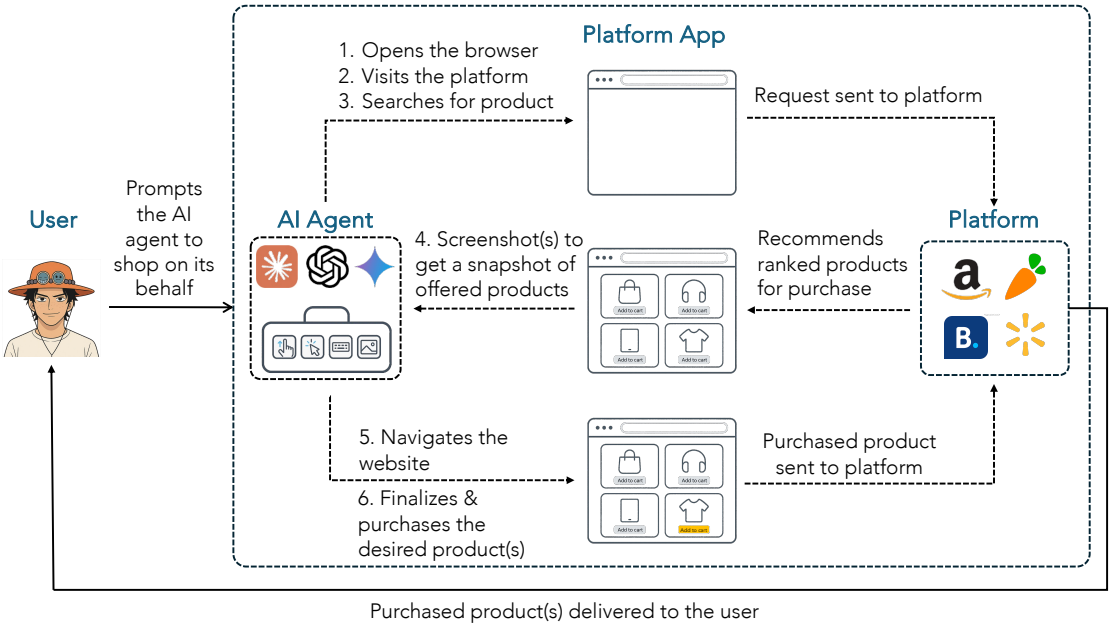


Figure 17: ***AI agentic workflow for seamless shopping.*** *The user delegates shopping to an AI agent and the AI agent autonomously navigates the platform website with the help of tool use.*

A.2 High-level Architecture & Approach

Given the complexity of employing VLM agents in the shopping task, ACES is designed with the following principles:

- **Reusability** Stakeholders, future researchers, and engineers might look to ACES for future research. Adapting the mock-app, providing new experiments, and extending the agent has been kept in mind.
- **Provider/Model agnostic** Using VLMs in the agent context raises 2 primary hurdles: the low-level code to interface with these VLMs is not the same; not all VLMs are designed to interact with external environments as signaled by the diverging implementations of [21]. We solve this by using LangChain[10] which provides a single interface for interacting with VLMs, and collapsing the action space to a single tool call. As a byproduct, the code is rendered easily extensible.
- **Observability** Understanding results and debugging must have minimal dependencies on external frameworks. As such, the entire agent journey is saved to a tree-like file hierarchy which can be easily inspected.

[10] https://github.com/langchain-ai/langchain

- **Cost & Efficiency** To ensure efficiency, the framework includes the boilerplate code to utilize "batch processing" to allow experimentation to be run at high-throughput.

To conduct experiments which model Figure 17, ACES is comprised of 3 top-level components—`agent`, `experiments`, and `sandbox`—each responsible for different portions of the workflow, and who interact through clearly defined interfaces.

The majority of the code is contained within `experiments` which encapsulates the dataset generation and randomization, collection of the results, and batch processing. The shopping agent itself is defined within `agent`; and `sandbox` contains the "mock-app".

A.3 Evaluation Details

Vision-Language-Models Evaluation. For VLM evaluation, we consider the following models: Anthropic's Claude Sonnet 3.5, 3.7, and 4; OpenAI's GPT-4o and GPT-4.1; and Google DeepMind's Gemini 2.0 Flash and Gemini 2.5 Flash. Notably, both OpenAI's GPT-4o and Google DeepMind's Gemini 2.0 Flash are directly productized as shopping agents through OpenAI's Operator and Google's Project Mariner, respectively. Following the initial rationality checks across all models, we conduct our detailed analysis on the latest release from each provider, under the assumption that older versions will be deprecated in favor of their successors in advanced applications. To maintain a roughly uniform baseline of capability, we exclude reasoning models and explicit reasoning components—such as omitting the default reasoning tokens for Gemini 2.5 Flash and Anthropic's Claude Sonnet 4, whereas OpenAI's GPT-4.1 does not support reasoning. Furthermore, reasoning models were excluded from our selection due to their higher latency, making them unfavorable in user-facing agentic tasks which require low latency inference [24, 37, 43].

B Details on the Experimental Setup and Main Results

B.1 Instruction Following and Rationality Experiments

B.1.1 Experimental Setup.

We provide some additional details on the experimental design and setup for our instruction-following and economic rationality experiments.

Instruction Following Experiments. We test the models for three types of instruction following tasks: (i) the ability to choose a product within a specified budget, (ii) choose a specific color and (iii) choose a specific brand. For each of the tasks, we selected two product categories summarized in Table 4. The tasks were specified in such a way that there was a unique product that satisfied the query. Figure 18 displays the mock-app setup for the instruction following tasks.

Table 4: Product categories shown for different tasks

Task	Product Category	Specified Task
Budget-Constrained	fitness watch	The budget constraint is $25
Budget-Constrained	toilet paper	The budget constraint is $10
Color-based	mousepad	Choose the pink color product
Color-based	stapler	Choose the pink color product
Brand-based	iphone 16 pro cover	Choose the Otterbox brand
Brand-based	toothpaste	Choose the Colgate brand

Price-Based Rationality Experiments. We have two types of price-based rationality experiments. In both, all products are identical except for price. In one setting (i) all price are equal, except for the price of one listing which is decreased by α-fraction and in a second setting (ii) we assign random prices to each of the listings drawn from a normal distribution $\mathcal{N}(\mu, \sigma^2)$ with mean μ and variance σ^2. For (i), we consider $\alpha \in \{0.1, 0.05, 0.01\}$ which corresponds to a $10\%, 5\%$ and 1% discount in the prices with respect to the other prices respectively (see Figure 19a). For (ii), we set the mean μ as the average of the eight product prices in their respective category. For the variance, we consider $\sigma = \$0.3$ (low variance; see Figure 19c) and $\sigma = 0.2\mu$ (high variance; see Figure 19e).

Rating-Based Rationality Experiments. We have three rating-based experiments: (i) the rating of one listing is increased by 0.1 (see Figure 19b), (ii) random ratings for the eight products with ratings ranging from 4.4 to 4.7 (low variance; see Figure 19d) and (iii) random ratings for the eight products with ratings ranging from 3.0 to 4.5 (high variance; see Figure 19f). Note that except for the ratings, all the other product attributes are the same.

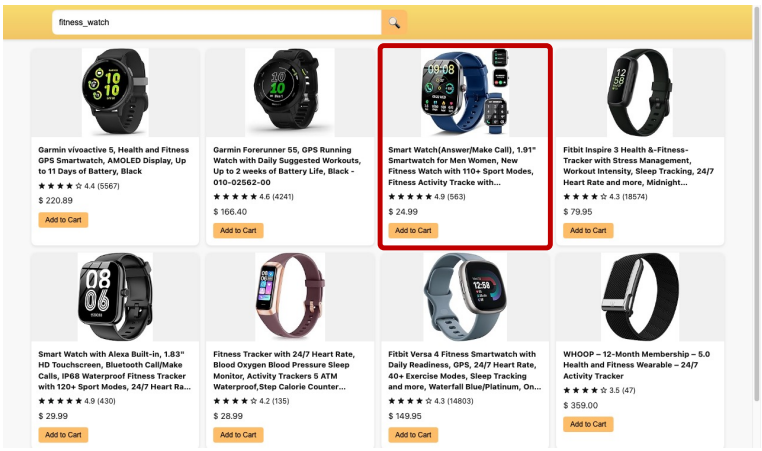
B.1.2 Main Results.

In this section we provide the associated fail rates of different AI models on the instruction-following and economic rationality tests.

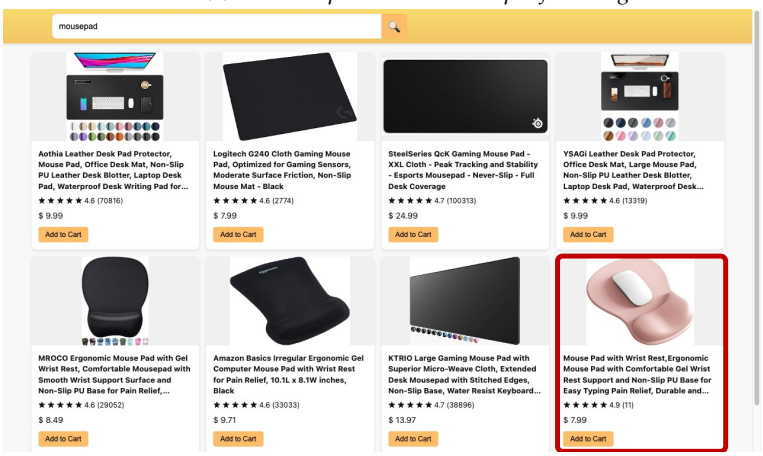
Instruction Following Experiments. We provide associated fail rates for the different instruction-following tasks (columns) and for different AI models (rows), in Table 5.

Table 5: Fail rate of different models on instruction-following tasks (standard errors in parentheses).

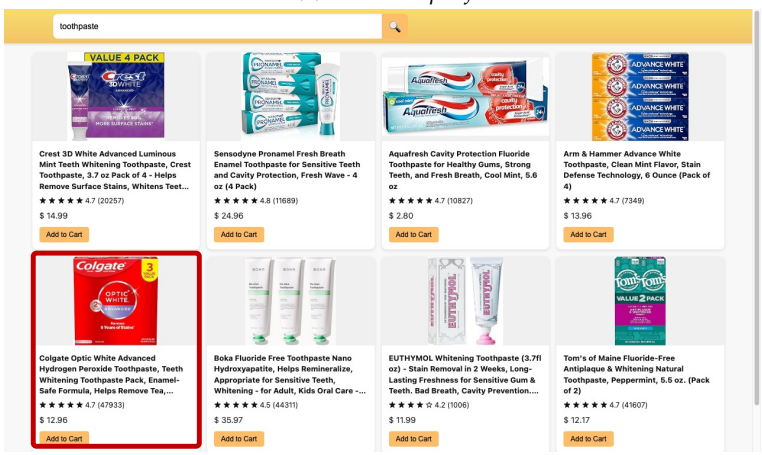
	Budget Constrained	Color Based	Brand Based
Claude Sonnet 3.5	4.0% (0.5%)	13.5% (1.4%)	0.0%
Claude Sonnet 3.7	0.0%	4.4% (0.5%)	0.0%
Claude Sonnet 4	0.0%	3.8% (0.5%)	0.0%
GPT–4o	0.0%	0.0%	0.0%
GPT–4.1	0.0%	0.0%	0.0%
Gemini 2.0 Flash	0.0%	0.0%	0.0%
Gemini 2.5 Flash	0.0%	0.0%	0.0%



(a) Choose a product within a specified budget



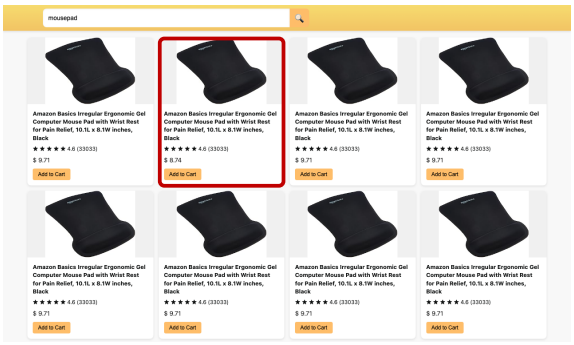
(b) Choose a specified color



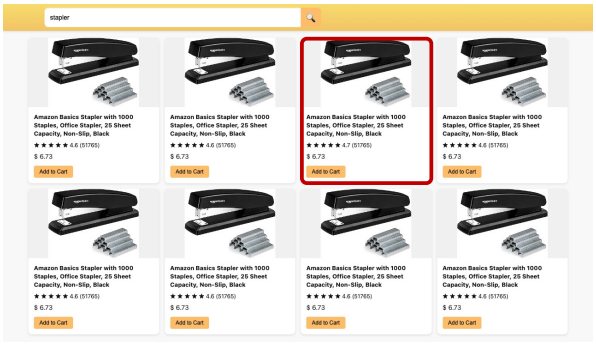
(c) Choose a specified brand

Figure 18: Design of different types of instruction following rationality tests. The optimal choice is highlighted in the red bounding box.

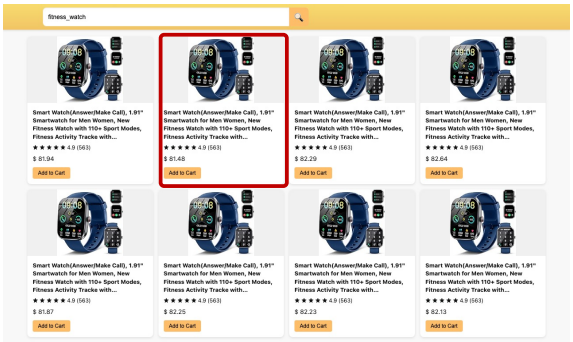
Price-Based Rationality Experiments. We provide associated fail rates for the different price-based rationality tasks (columns) and for different AI models (rows), in Table 6.



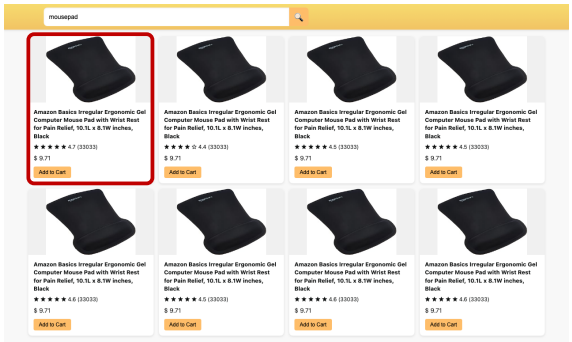
(a) 10% discount on one listing



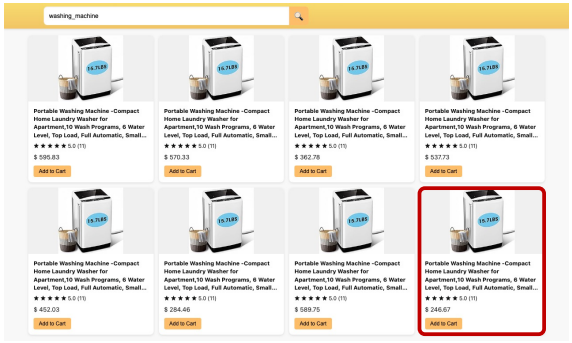
(b) +0.1 increase in rating of one listing



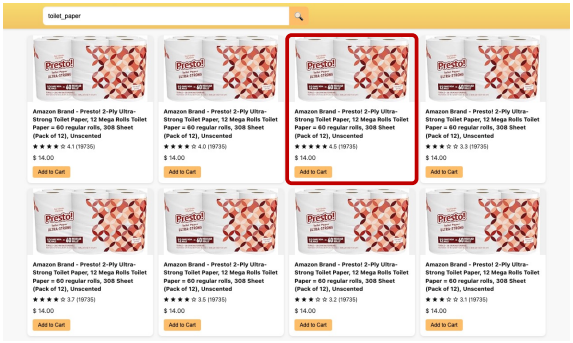
(c) Random prices (with low variance)



(d) Random ratings (with low variance)



(e) Random prices (with high variance)



(f) Random ratings (with high variance)

Figure 19: Design of price and rating based rationality test. The optimal choice is highlighted in the red bounding box.

Table 6: Fail rate of different models on price-based rationality tests (std. errors in parentheses).

	Price reduced (1% discount)	Price reduced (5% discount)	Price reduced (10% discount)	Random prices (low var.)	Random prices (high var.)
Claude Sonnet 3.5	63.7% (1.7%)	36.8% (1.4%)	32.5% (1.0%)	8.3% (0.4%)	5.0% (0.3%)
Claude Sonnet 3.7	21.0% (0.9%)	6.5% (0.7%)	1.2% (0.2%)	8.3% (0.4%)	6.0% (0.3%)
Claude Sonnet 4.0	0.5% (0.1%)	1.8% (0.2%)	0.0%	8.3% (0.3%)	4.3% (0.2%)
GPT–4o	25.8% (1.0%)	5.6% (0.4%)	6.8% (0.3%)	17.4% (0.9%)	3.6% (0.2%)
GPT–4.1	9.3% (0.6%)	1.3% (0.1%)	1.2% (0.2%)	12.6% (0.8%)	0.8% (0.1%)
Gemini 2.0 Flash	2.8% (0.2%)	0.5% (0.1%)	1.2% (0.2%)	1.0% (0.1%)	6.5% (0.3%)
Gemini 2.5 Flash	1.0% (0.1%)	0.0%	0.0%	0.8% (0.1%)	0.0%

Rating-Based Rationality Experiments. We provide associated fail rates for the different rating-based rationality tasks (columns) and for different AI models (rows), in Table 7.

Table 7: Fail rate of different models on rating-based rationality tests (std. errors in parentheses).

	Rating of one listing increased by 0.1	Random ratings (low variance)	Random ratings (high variance)
Claude Sonnet 3.5	57.3% (1.5%)	16.3% (0.8%)	2.7% (0.2%)
Claude Sonnet 3.7	6.7% (0.5%)	0.0%	0.0%
Claude Sonnet 4	28.7% (1.2%)	9.4% (0.6%)	4.7% (0.3%)
GPT–4o	71.7% (0.9%)	16.0% (0.5%)	7.3% (0.4%)
GPT–4.1	15.1% (0.6%)	11.7% (0.5%)	6.0% (0.4%)
Gemini 2.0 Flash	0.0%	0.0%	0.3% (0.1%)
Gemini 2.5 Flash	0.0%	0.0%	0.0%

Reasons of Failures. On the experiments where the models failed, we probed the explanation provided by the models. Broadly speaking, we identified three types of explanations.

- *Perceptual Limitations.* The model treated all products as identical, failing to recognize differences in price or rating. This type of error was particularly prevalent when the difference was subtle, for e.g., in the case when the price of one listing was reduced by 1%.
- *Unjustified Sub-optimality.* The model recognized the optimal product (e.g., the cheapest or highest-rated) but still made a sub-optimal choice without providing any specific justification for its decision.
- *Dismissal with justification.* The model acknowledged the optimal product but explicitly provided a reason for not selecting it. These reasons often included:
  - Prioritizing "Standard" Options. The model would state it "chose the standard price/rating" or the "most common price point", indicating a preference for typical values over slight deviations.

- Attributing difference to errors. The model sometimes dismissed the optimal option by attributing the observed difference to a "display error", a "temporary discount", an "outdated listing", or a "pricing error" on the platform.
- Perceived Insignificance. The model might justify its sub-optimal choice by stating the price difference was minimal or that there was "no apparent/significant advantage" or "no functional difference" compared to other options.

B.2 Choice Behavior and Biases

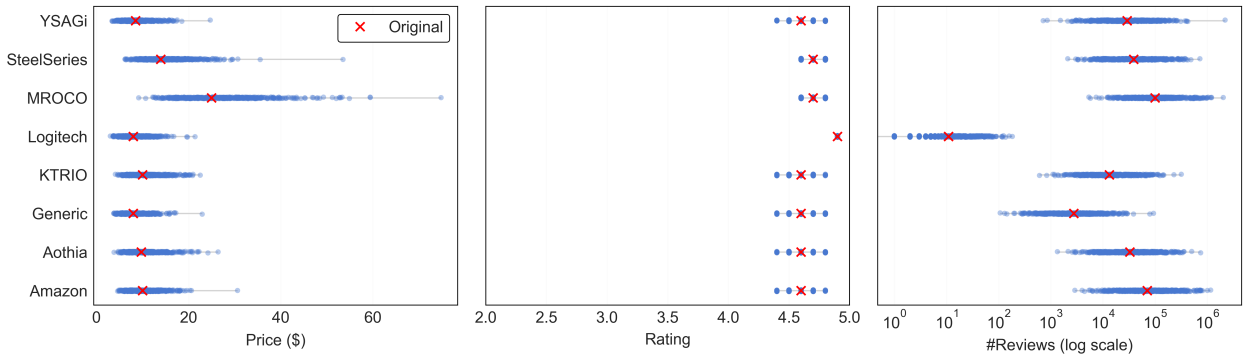
B.2.1 Experimental Details.

To estimate the sensitivity of the AI agent to different platform levers, we create a dataset with exogenous variation across a number of variables associated with the products (see Table 1). Given these variations are exogenous to the inherent attractiveness of the products, they allow us to obtain causal estimates to the impact of different platform levers like position, tags and product attributes like price, rating and number of reviews. We also provide a visualization of the exogenous variation of price, rating and number of reviews around their original value for a few product categories in Figure 20. Figure 21 provides an example of different mock-app screens for two different experiment runs. Notice that the assortment of the eight products is the same across the two experiments. However the position of the eight products change across the two experiments along with their price, rating and number of reviews. Also notice that the Sponsored, Overall Pick and Only X Remaining (scarcity) tags are assigned to different products across the two experiments.

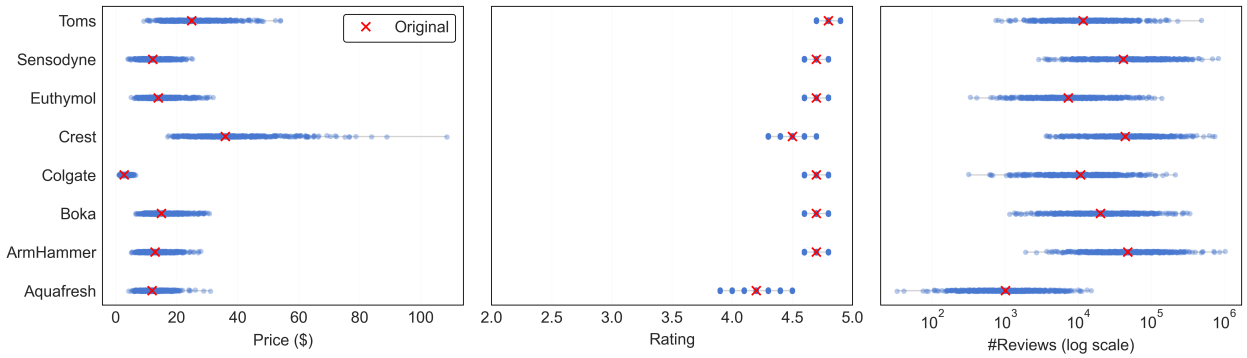
In Figure 21 we provide an example of different mock-app configurations experiments. Note that across the different experiments shown in Figure 21, the position of the products is randomly permuted along with random perturbation to their price, rating and number of reviews.

B.2.2 Understanding the trade-offs between position, tags and listing attributes

We fit a conditional logit with the utility specification in (5.1). The estimates are provided in Table 2. We can use the estimates in Table 2, (5.1) and (5.2) to understand how the different models trade-off different attributes. For example, we can be interested in understanding how the choice probability changes if there is a Sponsored tag or an Overall Pick tag or rating changes by ± 0.1. Let's say that the baseline selection probability of a product j is π_j and assume that we change co-variate z with coefficient β_z by Δz. Let $o_j = \pi_j/(1-\pi_j)$ be the odd-ratio for product j, then the

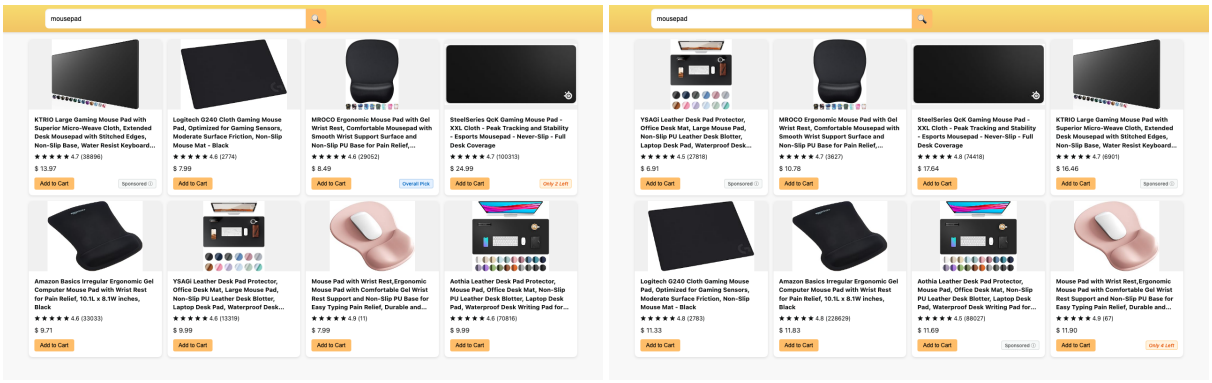


(a) mousepad



(b) toothpaste

Figure 20: Visualization of the exogenous variation in price, rating and number of reviews



(a) *(b)*

Figure 21: Example of the mock-app for different experiments with exogenous variations in position, tags and product attributes

new selection probability π'_j is given as

$$\pi'_j \overset{(a)}{=} \frac{\exp(U_j)\exp(\beta_z \Delta z)}{\sum_{k \neq i} \exp(U_k) + \exp(U_j)\exp(\beta_z \Delta z)}$$
$$\overset{(b)}{=} \frac{\pi_j \exp(\beta_z \Delta z)}{1 - \pi_j + \pi_j \exp(\beta_z \Delta z)}$$
$$\overset{(c)}{=} \frac{\exp(\beta_z \Delta z) o_j}{1 + \exp(\beta_z \Delta z) o_j},$$

where (a) follows from (5.2), (b) follows by dividing numerator and denominator by $\sum_k \exp(U_k)$ and the fact that $\pi_j = \exp(U_j)/\sum_{k\neq i}\exp(U_k)$ and (c) follows from the definition of o_j. Using this, we can calculate how the probability will change if we change some covariate.

We can also obtain price-equivalent trade-offs: a convenient way to compare levers is to ask: by how much can a seller raise (or must cut) price to keep the utility constant when the seller adds (or loses) a feature worth $\Delta U = \beta_z \Delta z$. Let say, we change the price by a multiplicative factor of λ, i.e., $P' = \lambda P$ where P is the original price of the product. Since we are keeping the utility constant, we have that $\beta_{\text{price}} \ln \lambda + \beta_z \Delta z = 0$, this implies that $\lambda = \exp\left(-\beta_z \Delta z/\beta_{\text{price}}\right)$. This allows to be get price-based exchange rate between different attributes. In Table 8, we present the price-equivalent results for different covariates.

Table 8: Price-equivalent trade–offs implied by the CL coefficients. Entries are approximate multiplicative changes in price that leave utility unchanged when the corresponding lever is added (top block) or removed/penalized (bottom block).

	Claude Sonnet 4	GPT-4.1	Gemini 2.5 Flash
*How much could a seller **raise** prices (holding utility fixed) if the seller gets...*			
Row 1 (vs Row 2)	+112.6%	+91.2%	+17.0%
Overall Pick tag	+92.2%	+64.5%	+137.8%
+0.1 rating	+35.4%	+67.3%	+27.9%
Double reviews	+19.4%	+37.4%	+17.2%
*How much would a seller need to **cut** price to offset the harm of ...*			
Sponsored tag	-8.0%	-14.3%	-11.3%
Scarcity tag	-4.6%	-6.3%	-14.5%

B.2.3 Results for prompt variation in VLM agents

We can think of the choice behavior with the default prompt (Figure 4) as the model's *prior*—its baseline choice rule absent explicit preferences. Estimates in Table 2 show a robust top-row premium and heterogeneous column preferences across models. To probe the strength of these position effects and the model prior, we modify the prompt to (i) ignore position (Figure 13) and (ii) prioritize price (Figure 14). We can view the "prioritize price" prompt as a minimal form of personalization: it encodes a plausible user preference without pre-solving the task.

B.3 Seller Response to AI-assisted shopping

In this section, we will discuss the seller response to the AI-assisted shopping. In Section B.3.1, we provide a case study example for the mousepad product category where the randomly selected focal product is the brand Aothia. In Section B.3.2, we provide category-wise product-level changes in market share before and after modifying the focal product's description.

Table 9: Estimates of the Conditional Logit Regression for Claude Sonnet 4 (VLM agents)

	Default (Fig. 4; Aug'25)	Default (Fig. 4; Sep'25)	Ignore Position (Fig. 13; Sep'25)	Ignore Position & Prioritize Price (Fig. 14; Sep'25)
Position effects				
Row 1	1.224^{***}	1.250^{***}	1.039^{***}	0.602^{***}
	(0.046)	(0.046)	(0.045)	(0.052)
Column 1	-0.297^{***}	-0.298^{***}	-0.254^{***}	0.216^{**}
	(0.065)	(0.065)	(0.065)	(0.074)
Column 2	0.557^{***}	0.505^{***}	0.527^{***}	0.390^{***}
	(0.058)	(0.058)	(0.059)	(0.073)
Column 3	0.416^{***}	0.399^{***}	0.441^{***}	0.225^{**}
	(0.059)	(0.059)	(0.060)	(0.073)
Badge effects				
Sponsored Tag	-0.135^{*}	-0.330^{*}	-0.156^{*}	-0.263^{**}
	(0.068)	(0.070)	(0.069)	(0.083)
Overall Pick Tag	1.060^{***}	1.009^{***}	0.686^{***}	-0.010
	(0.077)	(0.078)	(0.081)	(0.110)
Scarcity Tag	-0.076	-0.116	-0.033	-0.036
	(0.094)	(0.094)	(0.093)	(0.113)
Attribute effects				
ln(Price)	-1.623^{***}	-1.740^{***}	-1.762^{***}	-5.644^{***}
	(0.079)	(0.080)	(0.080)	(0.141)
Rating	4.913^{***}	4.886^{***}	5.788^{***}	2.072^{***}
	(0.218)	(0.220)	(0.229)	(0.233)
ln(Num. of Reviews)	0.415^{***}	0.431^{***}	0.583^{***}	0.248^{***}
	(0.023)	(0.023)	(0.024)	(0.024)
Product Fixed Effects	Yes	Yes	Yes	Yes
Observations	25,802	25,753	25,414	21,448
Choice Sets (Groups)	3,756	3,747	3,768	3,733
Pseudo R-squared	0.44	0.45	0.45	0.60

$^{*}\ p < 0.05$, $^{**}\ p < 0.01$, $^{***}\ p < 0.001$.

Table 10: Estimates of the Conditional Logit Regression for GPT-4.1 (VLM agents)

	Default (Fig. 4; Aug'25)	Default (Fig. 4; Sep'25)	Ignore Position (Fig. 13; Sep'25)	Ignore Position & Prioritize Price (Fig. 14; Sep'25)
Position effects				
Row 1	1.045^{***}	1.076^{***}	0.977^{***}	0.616^{***}
	(0.046)	(0.046)	(0.048)	(0.057)
Column 1	1.122^{***}	1.167^{***}	1.101^{***}	0.634^{***}
	(0.061)	(0.062)	(0.064)	(0.078)
Column 2	0.019	0.113	0.153^{*}	-0.113
	(0.065)	(0.065)	(0.067)	(0.080)
Column 3	-0.013	0.051	0.060	-0.102
	(0.066)	(0.066)	(0.067)	(0.080)
Badge effects				
Sponsored Tag	-0.248^{***}	-0.348^{***}	-0.282^{***}	-0.280^{**}
	(0.072)	(0.072)	(0.074)	(0.091)
Overall Pick Tag	0.802^{***}	0.786^{***}	0.582^{***}	0.089
	(0.083)	(0.083)	(0.089)	(0.117)
Scarcity Tag	-0.105	0.007	-0.001	-0.060
	(0.099)	(0.097)	(0.101)	(0.121)
Attribute effects				
ln(Price)	-1.612^{***}	-1.586^{***}	-1.750^{***}	-9.243^{***}
	(0.083)	(0.082)	(0.086)	(0.175)
Rating	8.300^{***}	7.862^{***}	9.047^{***}	4.345^{***}
	(0.269)	(0.261)	(0.285)	(0.282)
ln(Num. of Reviews)	0.739^{***}	0.756^{***}	0.937^{***}	0.535^{***}
	(0.026)	(0.026)	(0.028)	(0.031)
Product Fixed Effects	Yes	Yes	Yes	Yes
Observations	25,066	26,033	24,157	22,121
Choice Sets (Groups)	3,931	3,926	3,942	3,938
Pseudo R-squared	0.51	0.51	0.53	0.68

$^{*}\ p < 0.05$, $^{**}\ p < 0.01$, $^{***}\ p < 0.001$.

Table 11: Estimates of the Conditional Logit Regression for Gemini 2.5 Flash (VLM agents)

	Default (Fig. 4; Aug'25)	Default (Fig. 4; Sep'25)	Ignore Position (Fig. 13; Sep'25)	Ignore Position & Prioritize Price (Fig. 14; Sep'25)
Position effects				
Row 1	0.344^{***}	0.659^{***}	0.420^{***}	-0.206^{***}
	(0.041)	(0.042)	(0.041)	(0.050)
Column 1	-0.264^{***}	-0.247^{***}	-0.058	-0.248^{***}
	(0.057)	(0.057)	(0.058)	(0.071)
Column 2	-0.742^{***}	-0.701^{***}	-0.361^{***}	-0.212^{**}
	(0.061)	(0.060)	(0.060)	(0.070)
Column 3	0.162^{**}	0.092	0.297^{***}	-0.138^{*}
	(0.054)	(0.054)	(0.055)	(0.070)
Badge effects				
Sponsored Tag	-0.263^{***}	-0.223^{***}	-0.419^{***}	-0.260^{**}
	(0.067)	(0.067)	(0.068)	(0.082)
Overall Pick Tag	1.897^{***}	2.183^{***}	1.482^{***}	0.285^{**}
	(0.072)	(0.071)	(0.073)	(0.103)
Scarcity Tag	-0.342^{***}	-0.224^{*}	-0.131	-0.207
	(0.098)	(0.096)	(0.093)	(0.114)
Attribute effects				
ln(Price)	-2.190^{***}	-1.957^{***}	-2.035^{***}	-6.612^{***}
	(0.080)	(0.078)	(0.079)	(0.152)
Rating	5.388^{***}	5.739^{***}	5.965^{***}	1.453^{***}
	(0.218)	(0.223)	(0.227)	(0.222)
ln(Num. of Reviews)	0.501^{***}	0.491^{***}	0.636^{***}	0.179^{***}
	(0.023)	(0.023)	(0.024)	(0.027)
Product Fixed Effects	Yes	Yes	Yes	Yes
Observations	25,215	26,649	25,272	20,748
Choice Sets (Groups)	3,953	3,949	3,964	3,709
Pseudo R-squared	0.42	0.43	0.43	0.58

$^{*}\ p < 0.05$, $^{**}\ p < 0.01$, $^{***}\ p < 0.001$.

B.3.1 Case Study of Seller's Response to AI-assisted shopping

We consider the illustrative example of the mousepad product category. Within this category, our focal product (randomly selected) is the brand Aothia. We prompt an AI seller agent (GPT-4.1 in this case) with the simulated demand on the original product descriptions of the eight products in the catalog using GPT-4.1 as the AI buying agent. Below is the response of the AI seller agent on the recommended product description change.

AI Seller Agent's (GPT-4.1) Response

Here's how to make your Aothia Leather Desk Pad Protector title more AI- and customer-friendly, using only your provided features and aligning with what's selling well on this page:

Competitive Patterns

- Top competitors (Amazon Basics, KTRIO, MROCO) lead with the product's main use ("Ergonomic Mouse Pad," "Gaming Mouse Pad") and then surface key features: material, size, comfort, non-slip, waterproof, color.
- Titles are concise, not overloaded, but cover all core attributes and benefits.

Current Title

Aothia Leather Desk Pad Protector, Mouse Pad, Office Desk Mat, Non-Slip PU Leather Desk Blotter, Laptop Desk Pad, Waterproof Desk Writing Pad for Office and Home (Black, 23.6" x 13.7")

Weaknesses

- Good breadth, but the most important features are spread out.
- "Leather Desk Pad Protector" is slightly generic—PU Leather is a value signal.
- "Non-Slip" and "Waterproof" should be earlier.
- "Large" or "XL" size callout (if true) is attractive for this market.

Optimized Title Suggestions

Option 1: Feature-Rich, Direct

Aothia Large PU Leather Desk Pad, Non-Slip Waterproof Mouse Pad & Desk Mat, 23.6" × 13.7", Office and Home, Black

Why

- Leads with brand and size ("Large").
- "PU Leather" specifies the material.

- "Desk Pad, Non-Slip, Waterproof" are high-signal attributes and not buried.
- "Mouse Pad & Desk Mat" covers multi-purpose use.
- Size and color at the end for clarity and direct comparison.

Option 2: Even More Focused
Aothia PU Leather Desk Pad Protector, Large Non-Slip Waterproof Mouse Pad, 23.6" × 13.7", Office Desk Mat, Black
Why

- Highlights core value (protection, size, non-slip, waterproof) right after the brand.
- Short and very clear.

Option 3: Ultra-Compact
Aothia Large PU Leather Desk Pad, Non-Slip Waterproof, 23.6" × 13.7", Black
Why

- Super concise for listings where title space is limited.

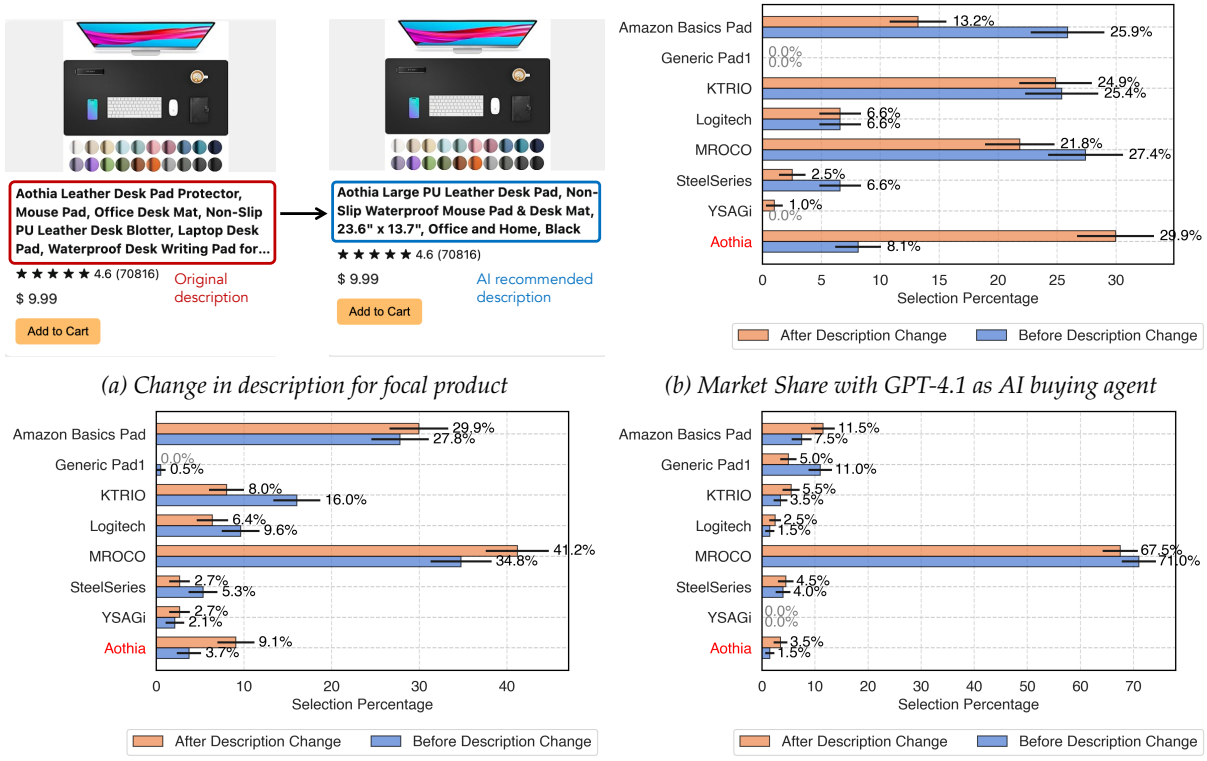
Why These Work

- All keywords (PU Leather, Large, Non-Slip, Waterproof, Mouse Pad, Desk Mat, Size, Black) are present and near the front.
- Follows competitor structure for quick parsing by AI agents.
- No added or exaggerated claims—uses your real features only.

Final Recommendation
To compete with both Amazon Basics and KTRIO, and help the AI agent recognize your desk pad's versatility and core selling points, use:
Aothia Large PU Leather Desk Pad, Non-Slip Waterproof Mouse Pad & Desk Mat, 23.6" × 13.7", Office and Home, Black

We run a new set of experiments where we just change the product description (see Figure 22a) of the focal product (`Aothia` in this case), with different AI buying agents (Claude Sonnet 4, GPT-4.1 and Gemini 2.5 Flash). In this case, we find that this rather innocuous looking change in the product description can significantly shift the market share. In Figure 22b, we plot the market share of different products for the mousepad category when using GPT-4.1 as the AI buying agent. We observe that there is a statistically significant and large increase of 21.8% in the market share

of the focal product Aothia. This comes at the expense of a decrease in market share of other products like the Amazon Basics Mousepad, MROCO and SteelSeries. Note that the only change across the two sets of experiments is the change in the product description, hence this impact is causal. This example illustrates the potential of changing market shares by simply optimizing the product description to appear more attractive to an AI buying agent. This is very much akin to search engine optimization (SEO), which is the practice of optimizing keywords in a website in order to have it rank higher in search results. We note that the gains in market share are not uniform across different AI buying agents. For the case of Claude Sonnet 4, we see that the change in market share of Aothia due to the product description change is around 5.3% and statistically significant while for the case of Gemini 2.5 Flash, the change in market share is only 2.0% (and not statistically significant), cf. Figures 22c and 22d. This demonstrates that there can be non-uniform impact of such interventions. In the next section, we provide the change in market shares across different product categories.



(a) Change in description for focal product

(b) Market Share with GPT-4.1 as AI buying agent

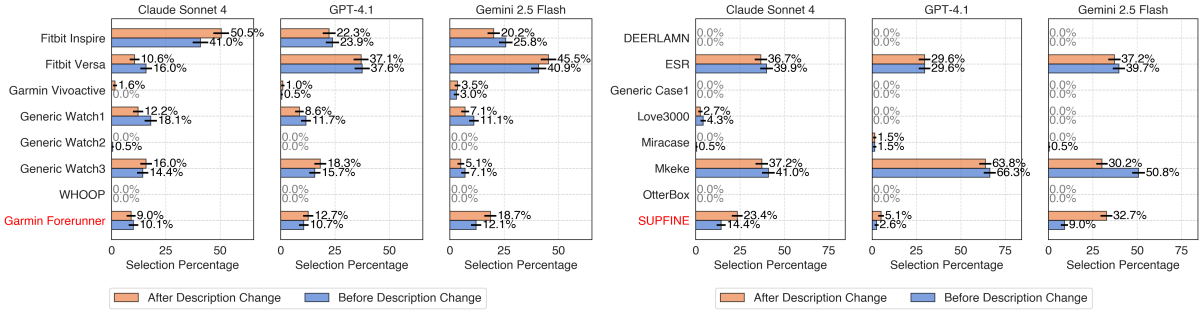
(c) Claude Sonnet 4 as AI buying agent

(d) Gemini 2.5 Flash as AI buying agent

Figure 22: Case Study for the `mousepad` *category with GPT-4.1 as the AI seller agent*

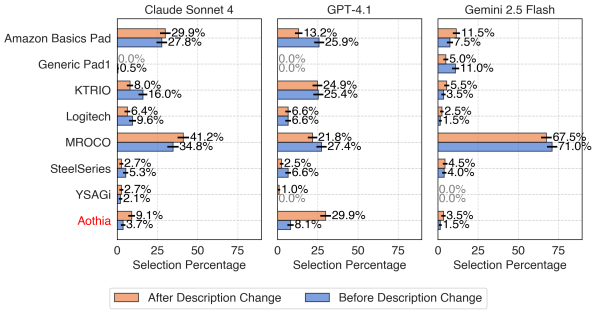
B.3.2 Category-wise change in market share

Figure 23 reports, for each category and AI agent model, the product-level selection shares before and after modifying the focal product's description, with the focal item highlighted in red.

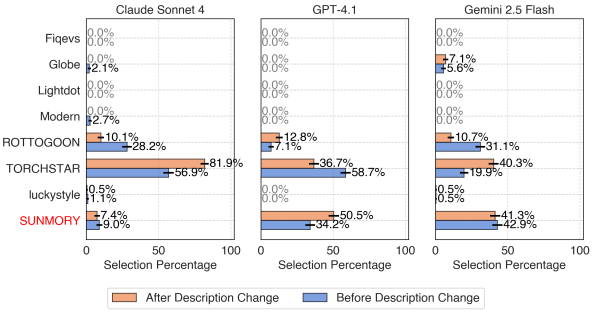


(a) fitness watch

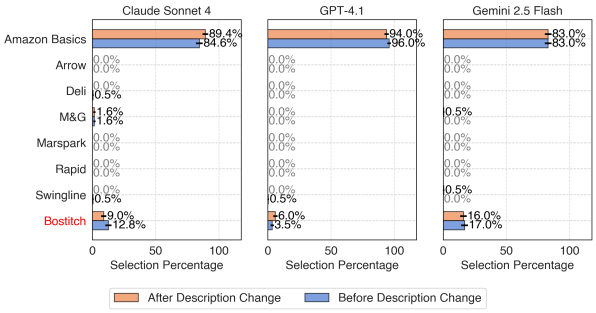
(b) iPhone 16 pro cover



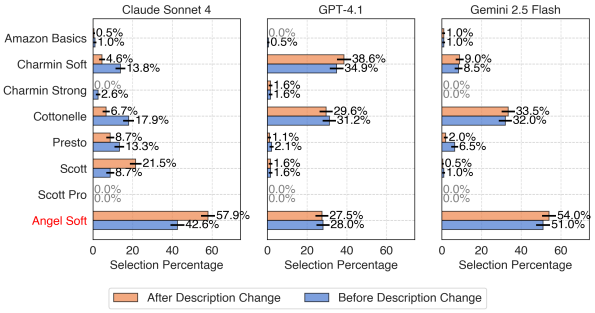
(c) mousepad



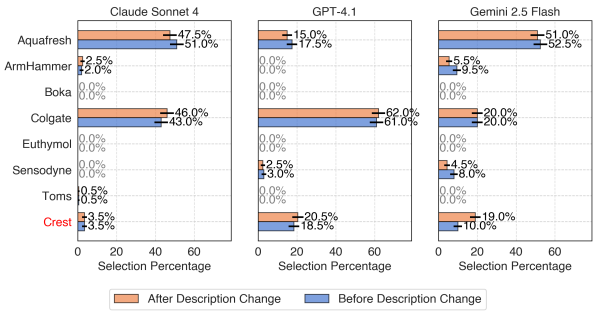
(d) office lamp



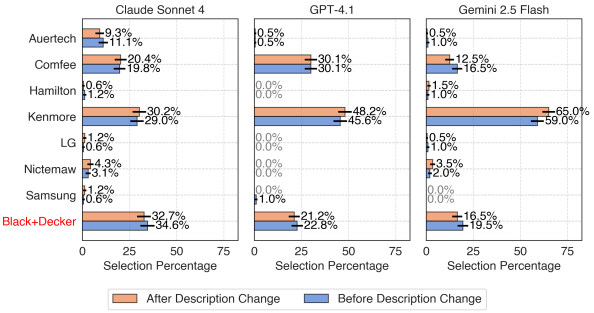
(e) stapler



(f) toilet Paper



(g) toothpaste



(h) washing machine

Figure 23: Market shares before and after focal product description modification. The focal product is marked in red.

C Results for the Headless AI Shopping Agents

We provide the estimates of the conditional logit model for different AI models under different prompt variations in the headless shopping setting.

Table 12: Estimates of the Conditional Logit Regression for Claude Sonnet 4 (headless AI agents)

	Default (Fig. 11)	Ignore Position (Fig. 15)	Ignore Position & Priortize Price (Fig. 16)
Position effects			
Position 1	2.010^{***}	1.947^{***}	1.150^{***}
	(0.106)	(0.101)	(0.107)
Position 2	1.370^{***}	1.017^{***}	0.753^{***}
	(0.107)	(0.102)	(0.108)
Position 3	1.357^{***}	0.992^{***}	0.854^{***}
	(0.108)	(0.103)	(0.108)
Position 4	1.024^{***}	0.673^{***}	0.646^{***}
	(0.109)	(0.104)	(0.111)
Position 5	0.857^{***}	0.578^{***}	0.613^{***}
	(0.111)	(0.105)	(0.109)
Position 6	0.677^{***}	0.482^{***}	0.430^{***}
	(0.111)	(0.106)	(0.110)
Position 7	0.252^{*}	0.013	0.281^{*}
	(0.115)	(0.109)	(0.111)
Badge effects			
Sponsored Tag	-0.673^{***}	-0.676^{***}	-0.234^{**}
	(0.083)	(0.081)	(0.083)
Overall Pick Tag	2.538^{***}	1.481^{***}	0.395^{***}
	(0.093)	(0.090)	(0.107)
Scarcity Tag	-0.674^{***}	-0.551^{***}	-0.076
	(0.121)	(0.117)	(0.114)
Attribute effects			
ln(Price)	-2.575^{***}	-2.564^{***}	-6.206^{***}
	(0.099)	(0.097)	(0.150)
Rating	9.701^{***}	10.767^{***}	3.273^{***}
	(0.314)	(0.322)	(0.240)
ln(Num. of Reviews)	0.800^{***}	0.990^{***}	0.373^{***}
	(0.030)	(0.031)	(0.028)
Product Fixed Effects	Yes	Yes	Yes
Observations	24,500	24,500	22,500
Choice Sets (Groups)	4,000	4,000	4,000
Pseudo R-squared	0.60	0.60	0.62

$^{*}\ p < 0.05$, $^{**}\ p < 0.01$, $^{***}\ p < 0.001$.

Table 13: Estimates of the Conditional Logit Regression for GPT-4.1 (headless AI agents)

	Default (Fig. 11)	Ignore Position (Fig. 15)	Ignore Position & Priortize Price (Fig. 16)
Position effects			
Position 1	1.078^{***}	0.743^{***}	0.044
	(0.090)	(0.090)	(0.118)
Position 2	0.125	-0.135	-0.204
	(0.095)	(0.094)	(0.117)
Position 3	0.312^{***}	0.175	-0.142
	(0.094)	(0.093)	(0.119)
Position 4	-0.140	-0.370^{***}	-0.251^{*}
	(0.097)	(0.098)	(0.121)
Position 5	0.319^{***}	0.196^{*}	0.125
	(0.094)	(0.093)	(0.119)
Position 6	0.184	0.003	-0.396^{***}
	(0.094)	(0.094)	(0.122)
Position 7	-0.155	-0.363^{***}	-0.204
	(0.097)	(0.096)	(0.120)
Badge effects			
Sponsored Tag	-1.815^{***}	-1.626^{***}	-0.556^{***}
	(0.092)	(0.088)	(0.099)
Overall Pick Tag	2.421^{***}	1.188^{***}	0.117
	(0.086)	(0.088)	(0.127)
Scarcity Tag	-0.383	-0.370^{***}	-0.303^{*}
	(0.108)	(0.108)	(0.136)
Attribute effects			
ln(Price)	-2.371^{***}	-2.414^{***}	-9.287^{***}
	(0.092)	(0.094)	(0.235)
Rating	11.373^{***}	12.177^{***}	3.114^{***}
	(0.316)	(0.331)	(0.271)
ln(Num. of Reviews)	0.944^{***}	1.212^{***}	0.278^{***}
	(0.030)	(0.033)	(0.033)
Product Fixed Effects	Yes	Yes	Yes
Observations	26,000	25,000	22,000
Choice Sets (Groups)	4,000	4,000	4,000
Pseudo R-squared	0.58	0.58	0.73

$^{*}\ p < 0.05$, $^{**}\ p < 0.01$, $^{***}\ p < 0.001$.

Table 14: Estimates of the Conditional Logit Regression for Gemini 2.5 Flash (headless AI agents)

	Default (Fig. 11)	Ignore Position (Fig. 15)	Ignore Position & Priortize Price (Fig. 16)
Position effects			
Position 1	0.394	0.487*	0.120
	(0.204)	(0.193)	(0.238)
Position 2	0.256	0.140	0.369
	(0.209)	(0.199)	(0.241)
Position 3	0.677***	0.514**	0.265
	(0.203)	(0.194)	(0.250)
Position 4	−0.373	−0.312	−0.274
	(0.216)	(0.206)	(0.259)
Position 5	−0.156	−0.043	0.168
	(0.218)	(0.204)	(0.241)
Position 6	−0.220	0.119	−0.206
	(0.213)	(0.199)	(0.250)
Position 7	−0.087	−0.104	−0.260
	(0.214)	(0.204)	(0.248)
Badge effects			
Sponsored Tag	−0.124	−0.125	−0.088
	(0.167)	(0.159)	(0.196)
Overall Pick Tag	3.175***	1.773***	0.935***
	(0.205)	(0.182)	(0.245)
Scarcity Tag	−0.650*	−0.041	0.177
	(0.264)	(0.229)	(0.279)
Attribute effects			
ln(Price)	−2.517***	−2.555***	−7.915***
	(0.214)	(0.200)	(0.424)
Rating	5.673***	5.429***	4.293***
	(0.537)	(0.513)	(0.597)
ln(Num. of Reviews)	0.814***	0.944***	0.518***
	(0.064)	(0.063)	(0.068)
Product Fixed Effects	Yes	Yes	Yes
Observations	4,500	4,600	4,300
Choice Sets (Groups)	800	800	800
Pseudo R-squared	0.55	0.49	0.65

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.