
The Risks and Rewards of User Expressiveness in Revenue-Driven Recommendation

Yeganeh Alimohammadi

Rigel Galgana

Negin Golrezaei

Abstract

1 Large-language-model recommendation interfaces let users communicate much
2 richer preference information than traditional, coarser search. Richer expres-
3 sion can improve product fit, but it also reveals *precisely* where a user is willing
4 to compromise—information a revenue-driven platform can use to favor higher-
5 revenue items. We study this tension in a feature-based model where a user sends
6 soft preferences over product features and optional hard filters to a platform with
7 a private catalog. The platform recommends the item maximizing per-purchase
8 revenue times a purchase probability evaluated at the reported match. We show
9 that truthful expressive reporting can be worse than coarser truthful reporting. We
10 then give a condition under which a strategic expressive user can find a near-
11 optimal item in polynomially many interactions by trying a one-dimensional fam-
12 ily of reports that counteracts the platform’s revenue bias by down-weighting high-
13 revenue features. When revenue differences are too large to be offset by soft re-
14 ports, expressive reporting can require exponentially many interactions. Similarly,
15 optimal reporting under coarse interfaces can also incur exponential search cost.

16 1 Introduction

17 Large language models are changing search and recommendation by expanding what a user can
18 communicate to a platform. Instead of expressing intent through keywords, menus, or a few hard
19 filters, users can describe nuanced tradeoffs in natural language. This richer expressiveness is valu-
20 able when the recommender acts solely on the user’s behalf. It also creates a new incentive prob-
21 lem when recommendations generate different commissions, sponsored-placement revenue, or self-
22 preferencing benefits [5, 42]. A detailed report reveals not only what the user prefers, but also where
23 the user is willing to compromise. A revenue-driven platform can use those compromises to steer
24 the user toward a more profitable item.

25 We ask: *when does richer expression actually help the user, when does it backfire, and when it*
26 *backfires, what can a careful user do and at what computational cost?* In this paper, we compare
27 a *ternary* interface, which records each feature as liked, disliked, or neutral, with an *expressive*
28 interface, which permits continuous feature-level reports. We distinguish a casual user who reports
29 truthfully from a strategic user—possibly an LLM shopping assistant—that is incentive-aware and
30 can intelligently adjust the user’s reported preferences to counteract a platform’s revenue bias.

31 Our results characterize the risks and rewards of expressive reporting. First, when every item gener-
32 ates the same per-purchase revenue, truthful expressive reporting is optimal. Once revenues differ,
33 however, revealing the precise strength of each preference can expose profitable compromises and
34 make a coarser truthful report strictly better. Second, a strategic expressive user can account for
35 the platform’s revenue incentive by sending reports of the form $\hat{s} = \alpha s - cp$, which lower the
36 reported value of high-revenue features. Trying a grid of values for the single scalar c finds an
37 approximately optimal item whenever changes in reported match have sufficiently large effects on
38 MNL purchase probability relative to revenue dispersion. Third, this polynomial guarantee is not
39 automatic. Bounded expressive reporting can require exponentially many interactions when rev-

40 enue differences cannot be overcome by any allowed report, while ternary reporting can require a
 41 combinatorial search even when one good ternary report exists.

42 2 Model

43 **User and catalog.** A user has a private preference vector $\mathbf{s} \in \mathcal{S} := \prod_{i=1}^d [-S_i, S_i]$ over d binary
 44 product features. A product $\mathbf{x} \in \{0, 1\}^d$ has user match $\langle \mathbf{s}, \mathbf{x} \rangle$. The platform has a private finite
 45 catalog $\mathcal{X} \subseteq \{0, 1\}^d$, drawn from a public distribution. We model purchase behavior by binary
 46 MNL with outside-option value m_0 :

$$f(m) = \frac{e^m}{e^{m_0} + e^m}, \quad u_{\mathbf{s}}(m) = \log(e^{m_0} + e^m), \quad \ell(m) := \log f(m). \quad (1)$$

47 The user-optimal item therefore maximizes $\langle \mathbf{s}, \mathbf{x} \rangle$. The derivative

$$\ell'(m) = 1 - f(m) = \frac{1}{1 + e^{m-m_0}} \quad (2)$$

48 measures how much the platform's log purchase probability changes after a one-unit increase in
 49 reported match. When m is high relative to m_0 , purchase probability is already close to one and this
 50 derivative is small.

51 **Reports and platform response.** A report consists of optional hard filters $(\mathcal{F}_0, \mathcal{F}_1)$ and a full soft-
 52 preference vector $\hat{\mathbf{s}}$. Hard filters require features in $\mathcal{F}_0$ to be absent and those in $\mathcal{F}_1$ to be present. In
 53 the ternary interface, $\hat{s}_i \in \{-S_i, 0, S_i\}$; in the expressive interface, $\hat{\mathbf{s}} \in \mathcal{S}$. Truthful binary reporting
 54 rounds each nonzero s_i to $S_i \cdot \text{sign}(s_i)$. If the hard filters are infeasible, the platform drops them and
 55 evaluates the same soft report on the full catalog. The platform earns per-purchase revenue

$$R(\mathbf{x}) = R_0(\mathbf{x}) + \delta_{\mathbf{x}}, \quad R_0(\mathbf{x}) = p_0 + \langle \mathbf{p}, \mathbf{x} \rangle, \quad |\delta_{\mathbf{x}}| < \bar{\delta}, \quad (3)$$

56 where $p_0 > 0$ and $\mathbf{p} \in \mathbb{R}_+^d$. The (strategic) user knows $(p_0, \mathbf{p}, \mathbf{S}, m_0, \bar{\delta})$ but not the catalog or
 57 item-specific errors. Among filter-feasible items, the platform maximizes

$$\log R(\mathbf{x}) + \ell(\langle \hat{\mathbf{s}}, \mathbf{x} \rangle). \quad (4)$$

58 A reporting plan submits T reports and observes one recommendation after each. We use T as the
 59 user's effort. Its loss is the utility gap between the user-optimal item and the best recommendation
 60 observed. To compare the revenue contribution of each feature with the strongest report the user can
 61 make about it, define

$$\rho := \max_{i \in [d]} \frac{p_i}{p_0 S_i}. \quad (5)$$

62 3 Risks and rewards of expressiveness

63 3.1 Truthful expressiveness can hurt

64 **Theorem 1.** *If $R(\mathbf{x}) = p_0$ for every item, truthful expressive reporting weakly dominates every*
 65 *other reporting plan. Moreover, there are random-catalog instances in which every single-shot*
 66 *ternary report loses $\Omega(\sqrt{d \log |\mathcal{X}|})$ expected MNL utility relative to truthful expressive reporting.*

67 With constant revenue, the platform ranks products only by reported match, so a truthful expressive
 68 report aligns its objective with the user's. Ternary reports discard the strength of each preference
 69 and can therefore select a worse item even when the platform and user otherwise prefer the same
 70 product. However, revenue dispersion can reverse this as a detailed report can reveal that a disliked
 71 high-revenue feature is only mildly bad, allowing the platform to trade a small loss in fit for a revenue
 72 gain. A coarse report may protect the user by exaggerating the dislike without banning every product
 73 containing that feature.

74 **Proposition 2.** *There are instances where truthful binary and truthful ternary reporting each have*
 75 *strictly smaller expected loss than truthful expressive reporting.*

76 3.2 Strategic expressive reporting

77 Let $\mathbf{x}^* \in \arg \max_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{s}, \mathbf{x} \rangle$. Let $\bar{m}_*$ upper-bound the reported match of $\mathbf{x}^*$ under the reports used
78 below, and define

$$\bar{\ell}' := \ell'(\bar{m}_*) = \frac{1}{1 + e^{\bar{m}_* - m_0}}. \quad (6)$$

79 Since ℓ' decreases with reported match, $\bar{\ell}'$ is a lower bound on the slope of ℓ at the relevant reports.

Algorithm 1: Strategic expressive reporting

Input: $(p_0, \mathbf{p}, \mathbf{S})$, $\bar{\ell}'$, preferences $\mathbf{s}$, and precision $\varepsilon > 0$.

1 $\rho \leftarrow \max_i p_i / (p_0 S_i)$; $\alpha \leftarrow 1 - \rho / \bar{\ell}'$; $C \leftarrow 1 / (p_0 \bar{\ell}')$, $\eta \leftarrow \alpha \bar{\ell}' \varepsilon / (2 \|\mathbf{p}\|_1)$;

2 Construct a grid $c_1 < \dots < c_T$ in $[0, C]$ with spacing at most η ;

3 **for** $t = 1, \dots, T$ **do**

4 submit $\hat{\mathbf{s}}^{(t)} = \alpha \mathbf{s} - c_t \mathbf{p}$ and observe $\hat{\mathbf{x}}^{(t)}$;

5 **return** $\arg \max_{t \in [T]} u_{\mathbf{s}}(\hat{\mathbf{x}}^{(t)})$;

80 Algorithm 1 is remarkably simple: it is essentially a line search and does not require *any* hard filters.
81 $\alpha \mathbf{s}$ preserves the direction of the user's preferences and $-c \mathbf{p}$ lowers the reported value of features
82 that increase platform revenue—effectively counter-acting the platform's revenue bias. The exact
83 value of c that gives the right tradeoff depends on the hidden catalog, so the user tries a line search
84 over this single number rather than searching over all expressive reports.

85 **Theorem 3.** *For exact linear revenue ($\bar{\delta} = 0$), Algorithm 1 returns an ε -optimal item using*

$$T = O\left(1 + \frac{\|\mathbf{p}\|_1}{p_0 \varepsilon (\bar{\ell}' - \rho)}\right) \quad (7)$$

86 *reports whenever $\bar{\ell}' > \rho$. More generally, fix a realized $(\mathbf{s}, \mathcal{X})$, let $r_{\min} := \min_{\mathbf{x} \in \mathcal{X}} R_0(\mathbf{x})$, and let
87 Δ_ε be the smallest user-match gap from $\mathbf{x}^*$ to any item whose user-utility loss exceeds ε . The same
88 guarantee holds for every $|\delta_{\mathbf{x}}| < \delta$ when $r_{\min} > \bar{\delta}$ and*

$$\alpha \bar{\ell}' \Delta_\varepsilon > 2\eta \|\mathbf{p}\|_1 + \log \frac{r_{\min} + \bar{\delta}}{r_{\min} - \bar{\delta}}. \quad (8)$$

89 At one value of c , the change in the platform's revenue term at $\mathbf{x}^*$ is exactly offset by the change
90 in reported match. Concavity then implies that no competing item has a larger platform score.
91 A sufficiently fine grid includes a nearby value of c . With item-specific revenue errors, the same
92 argument works when the resulting score difference is large enough to cover both the grid error and
93 the unknown errors.

94 3.3 When many interactions are necessary

95 The polynomial guarantee above does not hold for every instance. If revenue differences are large
96 enough, no bounded soft report can make the platform prefer the user-optimal item except when the
97 report is tailored to the hidden catalog. In other instances, every reported match is so high relative
98 to the outside option that purchase probability is close to one and changing reported match has little
99 effect on the platform's score.

100 **Theorem 4.** *For every fixed m_0 and every $\Delta_d = o(d)$, there are instances with arbitrarily large ρ
101 such that any adaptive expressive plan using $2^{o(d)}$ reports has expected loss at least $(1 - o(1))\Delta_d$.*

102 Soft reports face a similar problem when constrained to be *sparse*. In particular, a q -sparse ternary
103 report requires at most q coordinates across hard filters and nonzero soft entries. Even when one
104 such report would return the user-optimal item, the user may have to search over many possible
105 subsets of coordinates to find it.

106 **Theorem 5.** *For $q = q(d) \rightarrow \infty$ with $q = o(d)$, there are instances in which an adaptive q -
107 sparse ternary plan needs $\exp(\Omega(q \log(d/q)))$ reports to obtain $o(q)$ expected loss, even though
108 one q -sparse ternary report returns the user-optimal item. Without adaptivity, unrestricted ternary
109 reporting can require $\exp(\Omega(d))$ reports to obtain $o(d)$ expected loss.*

4 How much effort is needed to account for revenue bias?

Due to hard filters, the optimal item is always discoverable. However, as the catalog is hidden, the user cannot directly identify the user-optimal item or the report that returns it. Each report reveals only the single product, so measuring effort by the number of reports therefore captures how much experimentation is needed before the user sees a high-utility recommendation.

Polynomial effort with expressive reports. Algorithm 1 does not search over the full d -dimensional report space. It starts from the user’s preferences and changes only the scalar c multiplying the known revenue vector $\mathbf{p}$. All reports can be selected before any recommendations are observed; after receiving the recommendations, the user simply keeps the item with the highest user utility. The bound in (7) explains the required effort. Asking for half as much error requires roughly twice as many reports. Larger total revenue variation $\|\mathbf{p}\|_1/p_0$ requires a finer search. Most importantly, the number of reports grows as $\bar{\ell}' - \rho$ becomes small: when changes in reported match only barely outweigh the platform’s revenue incentive, the user must tune c much more precisely. When $\bar{\ell}'$ is comfortably larger than ρ , a much coarser grid is sufficient.

The reporting bounds S_i also have a direct interpretation. The ratio $p_i/(p_0 S_i)$ compares how much feature i can increase revenue with how strongly the user is allowed to report liking or disliking it. A larger reporting range gives the user more room to counteract the revenue incentive on that feature. The user does not need to know each item-specific error $\delta_{\mathbf{x}}$; condition (8) only requires those errors to be small relative to the advantage created for the user-optimal item over every item that is more than ε worse.

Exponential effort in the lower bounds. The negative results do not merely say that no good report exists. In the expressive construction, a suitable report exists when the realized catalog is known, but any one report works for only a very small number of the possible catalogs. Recommendations obtained from unsuccessful reports reveal too little about which catalog was drawn. Hence a user who tries fewer than exponentially many reports is unlikely to test a report that returns a near-optimal item. Allowing hard filters does not remove this problem: a successful filter must correctly identify most of the hidden coordinates of the desired product.

The ternary lower bound makes the source of effort especially transparent. A q -sparse user must decide which q of the d coordinates to mention. There are $\binom{d}{q}$ possible choices, and each report can work for only a small fraction of them, producing the $\exp(\Omega(q \log(d/q)))$ bound. With unrestricted but fixed-in-advance ternary reports, the three possible values on each coordinate still cannot represent all of the relevant hidden tradeoffs, so exponentially many reports may be needed.

These results give different implications for truthful and strategic users. A casual user submits one report, and a coarser truthful report can help by hiding the exact strength of weak preferences. A strategic user may instead spend several interactions trying to account for the platform’s revenue objective. With an expressive interface, the user need only vary one scalar and the required number of reports is polynomial under the condition in Theorem 3. With a ternary interface, or in the hard instances for expressive reporting, the required number can be exponential. Thus the user’s necessary effort ranges from one truthful report, to a polynomial number of carefully chosen expressive reports, to an exponential search when the interface or the catalog does not permit the revenue incentive to be offset efficiently.

References

- [1] S. Nageeb Ali, Greg Lewis, and Shoshana Vasserman. Voluntary disclosure and personalized pricing. *The Review of Economic Studies*, 90(2):538–571, 2023. doi: 10.1093/restud/rdac033.
- [2] Troels Andreasen, Gloria Bordogna, Guy De Tré, Janusz Kacprzyk, Henrik Legind Larsen, and Sławomir Zadrozny. The power and potentials of flexible query answering systems: A critical and comprehensive analysis. *Data & Knowledge Engineering*, 149:102246, 2024. doi: 10.1016/j.datak.2023.102246.
- [3] Santiago R. Balseiro and Antoine Désir. Incentive-compatible assortment optimization for sponsored products. *Management Science*, 69(8):4668–4684, 2023. doi: 10.1287/mnsc.2022.4603.
- [4] Dirk Bergemann and Stephen Morris. Information design: A unified perspective. *Journal of Economic Literature*, 57(1):44–95, 2019. doi: 10.1257/jel.20181489.
- [5] Ron Berman, Hangcheng Zhao, and Yi Zhu. Strategic design of recommendation algorithms. *SSRN Electronic Journal*, 2024. doi: 10.2139/ssrn.4301489. URL <https://ssrn.com/abstract=4301489>. Available at SSRN 4301489.
- [6] Musa Eren Celdir, Soo-Haeng Cho, and Elina H. Hwang. Popularity bias in online dating platforms: Theory and empirical evidence. *Manufacturing & Service Operations Management*, 26(2):537–553, 2024. doi: 10.1287/msom.2022.0132.
- [7] Sarah H. Cen, Andrew Ilyas, Jennifer Allen, Hannah Li, and Aleksander Madry. Measuring strategization in recommendation: Users adapt their behavior to shape future content, 2024.
- [8] Yeon-Koo Che, Kyungmin Kim, and Konrad Mierendorff. Keeping the listener engaged: A dynamic model of bayesian persuasion. *Journal of Political Economy*, 131(7):1797–1844, 2023. doi: 10.1086/722985.
- [9] Li Chen and Pearl Pu. Interaction design guidelines on critiquing-based recommender systems. *User Modeling and User-Adapted Interaction*, 19(3):167–206, 2009. doi: 10.1007/s11257-008-9057-x.
- [10] Michael Choi, Anovia Yifan Dai, and Kyungmin Kim. Consumer search and price competition. *Econometrica*, 86(4):1257–1281, 2018. doi: 10.3982/ECTA14837.
- [11] Vincent P. Crawford and Joel Sobel. Strategic information transmission. *Econometrica*, 50(6):1431–1451, 1982. doi: 10.2307/1913390.
- [12] Luca D’Amico-Wong, Gary Qiurui Ma, and David Parkes. Strategic recommendation: Revenue optimal matching for online platforms (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23468–23470, 2024. doi: 10.1609/aaai.v38i21.30432.
- [13] Mahsa Derakhshan, Negin Golrezaei, Vahideh Manshadi, and Vahab Mirrokni. Product ranking on online platforms. *Management Science*, 68(6):4024–4041, 2022. doi: 10.1287/mnsc.2021.4044.
- [14] Sébastien Destercke, Patrice Buche, and Valérie Guillard. A flexible bipolar querying approach with imprecise data and guaranteed results. *Fuzzy Sets and Systems*, 169(1):51–64, 2011. doi: 10.1016/j.fss.2010.12.014.
- [15] Paul Dütting, Safwan Hossain, Tao Lin, Renato Paes Leme, Sai Srivatsa Ravindranath, Haifeng Xu, and Song Zuo. Information design with large language models, 2025.
- [16] Ronald A. Dye. Disclosure of nonproprietary information. *Journal of Accounting Research*, 23(1):123–145, 1985. doi: 10.2307/2490910.
- [17] Jeffrey Ely, Alexander Frankel, and Emir Kamenica. Suspense and surprise. *Journal of Political Economy*, 123(1):215–260, 2015. doi: 10.1086/677350.

- [18] Boi Faltings, Pearl Pu, Marc Torrens, and Paolo Viappiani. Designing example-critiquing interaction. In *Proceedings of the 9th International Conference on Intelligent User Interfaces*, pages 22–29, 2004. doi: 10.1145/964442.964449.
- [19] Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. Advances and challenges in conversational recommender systems: A survey. *AI Open*, 2:100–126, 2021.
- [20] Daniel Garcia, Juha Tolvanen, and Alexander K. Wagner. Strategic responses to algorithmic recommendations: Evidence from hotel pricing. *Management Science*, 72(1):609–626, 2026. doi: 10.1287/mnsc.2022.03740.
- [21] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985. URL <https://api.semanticscholar.org/CorpusID:205092276>.
- [22] Keegan Harris, Nicole Immorlica, Brendan Lucier, and Aleksandrs Slivkins. Algorithmic persuasion through simulation, 2023.
- [23] Andreas Haupt, Dylan Hadfield-Menell, and Chara Podimata. Recommending to strategic users, 2023. URL <https://arxiv.org/abs/2302.06559>.
- [24] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. A survey on conversational recommender systems. *ACM Computing Surveys*, 54(5):1–36, 2021.
- [25] Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *American Economic Review*, 101(6):2590–2615, 2011. doi: 10.1257/aer.101.6.2590.
- [26] Werner Kießling. Foundations of preferences in database systems. In *Proceedings of the 28th International Conference on Very Large Data Bases*, 2002.
- [27] Akshit Kumar, Vahideh Manshadi, and Akhilesh Tumu. How much should a conversational recommender system converse?, 2026. Working paper.
- [28] Mehmet Kurmaz. Aware-us: Preference-aware infeasibility resolution in tool-calling agents. *arXiv preprint arXiv:2601.02643*, 2026.
- [29] Wenhao Li, Yue Lin, Xiangfeng Wang, Bo Jin, Hongyuan Zha, and Baoxiang Wang. Verbalized bayesian persuasion. *arXiv preprint arXiv:2502.01587*, 2025.
- [30] Tao Lin and Ce Li. Information design with unknown prior. In *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*, volume 325 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 72:1–72:1, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi: 10.4230/LIPIcs.ITCS.2025.72.
- [31] Brendan Lucier, Nicole Immorlica, Markus Mobius, Aleksandrs Slivkins, Daniel G. Goldstein, Jake M. Hofman, Sonia Jaffe, and David M. Rothschild. Agentic markets: Equilibrium effects of improving consumer search, 2026.
- [32] Paul R. Milgrom. Good news and bad news: Representation theorems and applications. *Bell Journal of Economics*, 12(2):380–391, 1981. doi: 10.2307/3003562.
- [33] Ali Shiralí. The burden of interactive alignment with inconsistent preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.
- [34] Ermis Soumalias, Yanchen Jiang, Kehang Zhu, Michael Curry, Sven Seuken, and David C. Parkes. Llm-powered preference elicitation in combinatorial assignment. *arXiv preprint arXiv:2502.10308*, 2025.
- [35] Mika Sumida, Guillermo Gallego, Paat Rusmevichientong, Huseyin Topaloglu, and James Davis. Revenue-utility tradeoff in assortment optimization under the multinomial logit model with totally unimodular constraints. *Management Science*, 67(5):2845–2869, 2021. doi: 10.1287/mnsc.2020.3657.

- 243 [36] Loukas Triantafyllopoulos, Evgenia Paxinou, Diamanto Tzanoulinou, Vassilios S. Verykios,
244 and Dimitris Kalles. The value alignment problem in advisory ai: a systematic literature review.
245 *AI and Ethics*, 6(1):147, 2026. doi: 10.1007/s43681-026-01015-4.
- 246 [37] Zhao Wang, Briti Gangopadhyay, Mengjie Zhao, and Shingo Takamatsu. Okg: On-the-fly
247 keyword generation in sponsored search advertising. In *Proceedings of the 31st Interna-*
248 *tional Conference on Computational Linguistics: Industry Track*, pages 115–127, 2025. URL
249 <https://aclanthology.org/2025.coling-industry.10/>.
- 250 [38] Jingheng Ye, Yong Jiang, Xiaobin Wang, Yinghui Li, Yangning Li, Hai-Tao Zheng, Pengjun
251 Xie, and Fei Huang. Productagent: Benchmarking conversational product search agent with
252 asking clarification questions. In *Proceedings of the 2025 Conference on Empirical Methods*
253 *in Natural Language Processing: Industry Track*, pages 383–398, 2025.
- 254 [39] Markus Zanker, Markus Jessenitschnig, and Wolfgang Schmid. Preference reasoning with soft
255 constraints in constraint-based recommender systems. *Constraints*, 15(4):574–595, 2010. doi:
256 10.1007/s10601-010-9098-8.
- 257 [40] Xiaoying Zhang, Hong Xie, Hang Li, and John C. S. Lui. Conversational contextual bandit:
258 Algorithm and application. In *Proceedings of The Web Conference 2020*, 2020.
- 259 [41] Yaochen Zhu, Harald Steck, Dawen Liang, Yinhan He, Nathan Kallus, and Jundong Li. Llm-
260 based conversational recommendation agents with collaborative verbalized experience. In
261 *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 2207–2220,
262 2025.
- 263 [42] Tianxin Zou and Bo Zhou. Self-preferencing and search neutrality in online retail platforms.
264 *Management Science*, 71(5):4087–4107, 2025. doi: 10.1287/mnsc.2022.01795. URL [https:](https://doi.org/10.1287/mnsc.2022.01795)
265 [//doi.org/10.1287/mnsc.2022.01795](https://doi.org/10.1287/mnsc.2022.01795).

266 Appendix contents

	Appendix A	Related work.
	Appendix B	Full model and notation.
267	Appendix C	Formal statements and proofs.
	Appendix D	Experimental design and additional results.

268 A Related work

269 **Conversational and expressive recommender systems.** Conversational and LLM-based recom-
 270 menders expand traditional search interfaces beyond keywords, clicks, ratings, and a small number
 271 of constraints. Earlier work on preference modeling and flexible constraint handling studied soft
 272 preferences and query relaxation [2, 14, 26], while example-critiquing and preference-based retrieval
 273 developed iterative ways for users to refine ambiguous needs [9, 18, 39]. Conversational recommen-
 274 dation systems extend this approach through adaptive questioning, diversified recommendations,
 275 and exploration–exploitation tradeoffs [19, 24, 40]. Recent LLM-based systems further broaden
 276 the communication channel by allowing natural-language product requirements, preference-aware
 277 repair of infeasible requests, and multi-turn intent inference [28, 37, 38, 41].

278 Much of this work views richer expression as a path to better personalization when the system uses
 279 additional information on the user’s behalf. Recent economic models show that this value depends
 280 on market primitives: platforms may choose how much to elicit [27], and agentic search can affect
 281 information acquisition, price competition, and consumer surplus [10, 31]. We ask a complementary
 282 question: when recommendations are revenue-sensitive, which reporting languages actually help the
 283 user? Our results show that expressiveness can enable efficient search when the platform’s purchase
 284 model is sufficiently responsive to fit, but can also expose naive users to revenue-driven steering;
 285 meanwhile, coarser reports may protect truthful users yet be exponentially hard to search over for
 286 strategic ones.

287 **Strategic users and revenue-driven platforms.** A related literature studies recommendation sys-
 288 tems with strategic users, sellers, or platforms. Users may be harmed by truthful or myopic in-
 289 teraction with systems that learn from behavior, and may strategically curate likes, consumption,
 290 or engagement signals to shape future recommendations [7, 23, 33]. Other work studies biased or
 291 revenue-sensitive recommendation and algorithmic advice, emphasizing that recommendation is not
 292 merely prediction when platform and user objectives diverge [12, 20, 34, 36]. Similar tensions arise
 293 in assortment, ranking, and sponsored recommendation, where platforms trade off revenue, market
 294 share, consumer welfare, and seller exposure [3, 6, 13, 35]. We connect these concerns to interface
 295 design: the question is not only whether the platform is biased, but whether users can computa-
 296 tionally find reports that counteract that bias.

297 Our model also relates to strategic disclosure and information design. Classical cheap-talk
 298 and voluntary-disclosure models study selective revelation by informed senders [11, 16, 32],
 299 while Bayesian persuasion studies committed information structures that shape receiver behavior
 300 [4, 8, 17, 25]. Recent work brings these ideas to algorithmic and market settings, including person-
 301 alized pricing [1], information design with unknown priors [30], and algorithmic or LLM-mediated
 302 persuasion [15, 22, 29]. Our setting focuses on structured recommender reports: hard filters con-
 303 strain feasibility, while soft feature-level reports affect ranking. This lets us separate how the plat-
 304 form responds to reported match, the expressiveness of the interface, and the number of interactions
 305 needed to find a good report.

306 B Full model and notation

307 **User’s problem.** The user has a private preference vector $\mathbf{s} \in \mathcal{S} := \prod_{i=1}^d [-S_i, S_i]$, drawn from
 308 a public distribution $F_{\mathbf{s}}$. Products are represented by binary vectors $\mathbf{x} \in \{0, 1\}^d$, and $\langle \mathbf{s}, \mathbf{x} \rangle$ is the
 309 user’s match score. We use the binary MNL purchase model with outside option $m_0 \in \mathbb{R}$:

$$f(m) = \frac{\exp(m)}{\exp(m_0) + \exp(m)}, \quad u_{\mathbf{s}}(m) = \log(\exp(m_0) + \exp(m)).$$

310 The additive normalization of utility is omitted because it does not affect comparisons. We write
 311 $u_s(\mathbf{x}) := u_s(\langle \mathbf{s}, \mathbf{x} \rangle)$. Since this utility is strictly increasing in match, the user-optimal item is any
 312 maximizer of $\langle \mathbf{s}, \mathbf{x} \rangle$.

313 The outside option may be interpreted as a publicly calibrated parameter of the platform's committed
 314 purchase model. The analysis requires that the user knows the scoring rule to which the platform
 315 commits; it does not require that the platform observe the user's private preference vector.

316 **Reporting interface.** A report contains a full soft-preference vector $\hat{\mathbf{s}} \in \mathcal{S}$ and optional hard filters
 317 $(\mathcal{F}_0, \mathcal{F}_1)$, where the two sets are disjoint. Features in $\mathcal{F}_0$ must be absent and features in $\mathcal{F}_1$ must
 318 be present. The soft report covers all features, including those also named in a filter. In a ternary
 319 interface, $\hat{s}_i \in \{-S_i, 0, S_i\}$; in an expressive interface, any $\hat{\mathbf{s}} \in \mathcal{S}$ is permitted.

320 A truthful reporting plan uses one interaction and deterministically coarsens $\mathbf{s}$. We study truthful
 321 expressive reporting ($\hat{\mathbf{s}} = \mathbf{s}$), truthful ternary reporting (coordinatewise rounding to the nearest point
 322 in $\{-S_i, 0, S_i\}$), and truthful binary reporting (each nonzero coordinate rounded to $S_i \text{sign}(s_i)$). A
 323 strategic plan may submit several reports and use the public platform parameters. It is adaptive if
 324 later reports depend on earlier recommendations and non-adaptive if all reports are fixed in advance.

325 Given recommendations $\hat{\mathbf{x}}^{(1)}, \dots, \hat{\mathbf{x}}^{(T)}$, the user's loss is

$$\text{Loss}_T = \max_{\mathbf{x} \in \mathcal{X}} u_s(\mathbf{x}) - \max_{t \leq T} u_s(\hat{\mathbf{x}}^{(t)}).$$

326 Expected-loss statements average over $\mathbf{s} \sim F_s$ and $\mathcal{X} \sim F_{\mathcal{X}}$.

327 **Platform's problem.** The platform has a private finite catalog $\mathcal{X} \subseteq \{0, 1\}^d$, drawn from a public
 328 distribution $F_{\mathcal{X}}$. Its per-purchase revenue is

$$R(\mathbf{x}) = R_0(\mathbf{x}) + \delta_{\mathbf{x}}, \quad R_0(\mathbf{x}) = p_0 + \langle \mathbf{p}, \mathbf{x} \rangle, \quad |\delta_{\mathbf{x}}| < \bar{\delta},$$

329 where $p_0 > 0$, $\mathbf{p} \in \mathbb{R}_+^d$, and $R(\mathbf{x}) > 0$. The user knows the linear revenue model R_0 and the error
 330 bound but not the realized catalog or the item-specific errors. Let

$$\ell(m) := \log f(m) = m - \log(\exp(m_0) + \exp(m)).$$

331 For a report $(\mathcal{F}_0, \mathcal{F}_1, \hat{\mathbf{s}})$, define

$$\mathcal{X}(\mathcal{F}_0, \mathcal{F}_1) := \{\mathbf{x} \in \mathcal{X} : x_i = 0 \ \forall i \in \mathcal{F}_0, \ x_i = 1 \ \forall i \in \mathcal{F}_1\}.$$

332 When this set is nonempty, the platform recommends a maximizer of

$$\log R(\mathbf{x}) + \ell(\langle \hat{\mathbf{s}}, \mathbf{x} \rangle)$$

333 over the filter-feasible catalog, breaking ties uniformly. If the filters are infeasible, the platform
 334 drops them and applies the same soft report to the full catalog. Thus hard filters determine which
 335 products may be recommended when they are feasible, while the soft report affects the ranking of
 336 those products.

337 **Sensitivity to reported match and revenue dispersion.** The slope of the platform's log purchase
 338 probability is

$$\ell'(m) = 1 - f(m) = \frac{1}{1 + \exp(m - m_0)}.$$

339 This quantity measures how much a change in reported match changes the platform's score. It is
 340 smaller when reported match is far above the outside option and purchase probability is already
 341 close to one. We compare this slope with

$$\rho = \max_i \frac{p_i}{p_0 S_i},$$

342 which measures revenue dispersion relative to the range of reports allowed on each feature. The
 343 denominator S_i reflects that the revenue contribution of feature i is easier to counteract when the
 344 user can report a stronger preference about that feature.

Role of hard filters. If the user already knew the desired product, hard filters on every coordinate could make that product the only feasible item. The lower bounds therefore concern how many interactions are needed to find a report that works when the catalog is hidden. They must also account for the rule for infeasible filters: if a set of hard filters is infeasible, the platform evaluates the same soft report over the full catalog.

C Formal statements and proofs for main-body results

We begin with a simple observation that separates two questions: whether a known item can be returned by some report, and how many interactions are needed to find such a report when the catalog is hidden.

Observation 6. *For every catalog $\mathcal{X} \subseteq \{0, 1\}^d$, every item $\mathbf{x} \in \mathcal{X}$ can be induced if the user already knows $\mathbf{x}$.*

Proof overview. The user can impose enough hard filters to make the known item the only feasible product.

Proof. For any item $\mathbf{x}$, let

$$\mathcal{F}_1 = \{i : x_i = 1\}, \quad \mathcal{F}_0 = \{i : x_i = 0\}.$$

Then every catalog item satisfying the hard filters agrees with $\mathbf{x}$ on every coordinate, so the feasible set inside $\mathcal{X}$ is the singleton $\{\mathbf{x}\}$. The platform therefore returns $\mathbf{x}$. This observation is not algorithmic: it requires the user to know the item they want to isolate. $\square$

C.1 Proof of Theorem 1

Theorem 7. *Assume $R(\mathbf{x}) = p_0$ for every item. Then truthful expressive reporting weakly dominates every other report in user utility. Moreover, in the special case $S_i = 1$ for every coordinate, if $N \rightarrow \infty$ satisfies $\log N = o(d)$, there exist distributions $F_{\mathbf{s}}$ and $F_{\mathcal{X}}$, with $|\mathcal{X}| = N$, such that truthful expressive reporting exceeds any single-shot ternary report by $\Omega(\sqrt{d \log N})$ expected MNL utility.*

Proof overview. With no revenue dispersion, the platform ranks products only by reported match. Truthful expressive reporting therefore makes the platform maximize the user's match. The quantitative separation uses a random catalog: continuous reports align with the full preference vector, while ternary reports can align only with a coarsened preference vector. Hard filters do not invalidate the separation. If they leave no feasible product, the platform drops them; if they leave a feasible set, a small number of hard filters can improve match only by a lower-order amount, while a large number are feasible with very small probability.

Proof. First suppose $R(\mathbf{x}) = p_0$ for every item. Under the truthful expressive report with no hard filters, the platform maximizes $\ell(\langle \mathbf{s}, \mathbf{x} \rangle)$ over all $\mathbf{x} \in \mathcal{X}$. Since ℓ is strictly increasing, every returned item maximizes $\langle \mathbf{s}, \mathbf{x} \rangle$ over $\mathcal{X}$. Since $u_{\mathbf{s}}(m)$ is strictly increasing in m , this item is user-optimal. Any other report, possibly with hard filters, can only induce an item in $\mathcal{X}$, whether while the filters are feasible or after infeasible filters are dropped. It therefore cannot induce an item with strictly larger user utility. This proves pointwise weak dominance.

For the ternary lower bounds, take $S_i = 1$ for every coordinate, set $s_i = i/d$, and draw N products $\mathbf{x}^1, \dots, \mathbf{x}^N$ independently and uniformly from $\{0, 1\}^d$. Write $z_i^n = 2x_i^n - 1$. Under truthful expressive reporting,

$$\langle \mathbf{s}, \mathbf{x}^n \rangle = \frac{1}{2} \sum_{i=1}^d s_i + \frac{1}{2} \sum_{i=1}^d s_i z_i^n.$$

The standard moderate-deviation estimate for maxima of weighted Rademacher sums gives

$$\mathbb{E} \left[\max_{n \leq N} \sum_i s_i z_i^n \right] = (1 + o(1)) \|\mathbf{s}\|_2 \sqrt{2 \log N} = \left(\sqrt{\frac{2d}{3}} + o(\sqrt{d}) \right) \sqrt{\log N},$$

385 because $\|\mathbf{s}\|_2^2 = d/3 + O(1)$ and $\log N = o(d)$. Thus truthful expressive reporting induces expected
 386 match

$$\frac{1}{2} \sum_{i=1}^d s_i + \left(\frac{1}{\sqrt{6}} + o(1) \right) \sqrt{d \log N}.$$

387 Now fix any single-shot ternary report, including hard filters. Let $h = |\mathcal{F}_0| + |\mathcal{F}_1|$ be the number
 388 of hard-filtered coordinates. If $h \geq h_0 := \lceil \log_2 N + 3 \log_2 d \rceil$, then a random catalog item satisfies
 389 the hard filters with probability at most 2^{-h_0} , so the probability that any catalog item is feasible is
 390 at most $N 2^{-h_0} \leq d^{-3}$. On the complementary event, the platform drops the hard filters and applies
 391 the same ternary soft report over the full catalog. The rare feasible event can change the expected
 392 match by at most $d \cdot d^{-3} = o(\sqrt{d \log N})$. Therefore reports with $h \geq h_0$ are, up to a lower-order
 393 term, no better than the same ternary soft report with no hard filters.

394 It remains to consider reports with $h < h_0$. The hard filters can affect at most $h_0 = o(\sqrt{d \log N})$
 395 coordinates, so their contribution to the selected product's user match is lower order. Thus it suffices,
 396 up to lower-order terms, to upper-bound ternary soft reports without hard filters.

397 Fix $\hat{\mathbf{s}} \in \{-1, 0, 1\}^d$, and let $A = \{i : \hat{s}_i \neq 0\}$. The product selected by maximizing $\sum_i \hat{s}_i z_i^n$ has
 398 expected true-match gain at most

$$\frac{1}{2} (1 + o(1)) \frac{\langle \mathbf{s}, \hat{\mathbf{s}} \rangle}{\sqrt{|A|}} \sqrt{2 \log N},$$

399 with the trivial bound used if $|A|$ is bounded. The best signs are positive on selected coordinates,
 400 and the best support of size k consists of the k largest coordinates of $\mathbf{s}$. Hence

$$\max_{\hat{\mathbf{s}} \in \{-1, 0, 1\}^d} \frac{\langle \mathbf{s}, \hat{\mathbf{s}} \rangle}{\sqrt{\|\hat{\mathbf{s}}\|_0}} = \max_{1 \leq k \leq d} \frac{\sum_{i=d-k+1}^d i/d}{\sqrt{k}} = \left(\frac{2\sqrt{2}}{3\sqrt{3}} + o(1) \right) \sqrt{d}.$$

401 Therefore every ternary report has expected match at most

$$\frac{1}{2} \sum_{i=1}^d s_i + \left(\frac{2}{3\sqrt{3}} + o(1) \right) \sqrt{d \log N}.$$

402 Since $1/\sqrt{6} > 2/(3\sqrt{3})$, truthful expressive reporting has an $\Omega(\sqrt{d \log N})$ match advantage.

403 Finally, the truthful expressive match is at least $d/5$ with probability $1 - \exp(-\Omega(d))$. Let M_E and
 404 M_T be the truthful expressive and ternary matches. If $M_T \geq m_0$, then $u'_s(m) = f(m) \geq 1/2$
 405 on $[M_T, M_E]$. If $M_T < m_0$, then $u_s(M_E) - u_s(M_T) \geq u_s(d/5) - u_s(m_0) = \Omega(d)$, which
 406 is larger than the claimed separation. Thus the match gap converts into an $\Omega(\sqrt{d \log N})$ expected
 407 MNL utility gap. $\square$

408 C.2 Proof of Proposition 2

409 **Proof overview.** The construction has two catalog states. In each state, one good product avoids one
 410 mildly disliked high-revenue feature, while the high-revenue product contains both such features. A
 411 truthful expressive report reveals that each dislike is mild, so the high-revenue product wins. A
 412 truthful ternary or binary report exaggerates every dislike, so the state-specific good product wins.
 413 Hard filters cannot replicate this across all states: ruling out a disliked feature helps only in the state
 414 whose good product omits that feature, and in all other states the platform drops the hard filters and
 415 returns the same item as the expressive report without hard filters.

416 *Proof.* Take $m_0 = 0$, $d = 3$, and $S_i = 1$ for all coordinates. Let

$$s_0 = 1, \quad s_1 = s_2 = -a,$$

417 where $a = 3/4$. The common coordinate 0 is liked, and coordinates 1 and 2 are mildly disliked.
 418 Define

$$\mathbf{x}^H = \mathbf{1}_{\{0,1,2\}}, \quad \mathbf{x}_1^L = \mathbf{1}_{\{0,2\}}, \quad \mathbf{x}_2^L = \mathbf{1}_{\{0,1\}}.$$

419 The catalog is $\mathcal{X}_j = \{\mathbf{x}^H, \mathbf{x}_j^L\}$ with probability $1/2$ for each $j \in \{1, 2\}$. The user matches are

$$\langle \mathbf{s}, \mathbf{x}^H \rangle = 1 - 2a, \quad \langle \mathbf{s}, \mathbf{x}_j^L \rangle = 1 - a,$$

so $\mathbf{x}_j^L$ is user-optimal in state j . Now, for convenience, we choose a log-revenue advantage Γ satisfying

$$\ell(1-a) - \ell(1-2a) < \Gamma < \ell(0) - \ell(-1). \quad (9)$$

For $a = 3/4$, the left side is strictly smaller than the right side, so such a Γ exists. Because $m_0 = 0$, the right side is strictly smaller than $\log 2$, so the chosen Γ can be realized with additive linear revenue by setting the same positive price p on coordinates 1 and 2 and choosing $p_0 > 0$ such that

$$\log \frac{p_0 + 2p}{p_0 + p} = \Gamma.$$

Now, assume that all item-specific errors are zero. Under truthful expressive reporting with no hard filters, $\mathbf{x}^H$ beats $\mathbf{x}_j^L$ in every state by the left inequality in (9). Under truthful binary or truthful ternary reporting, the disliked coordinates are reported as -1 , so $\mathbf{x}_j^L$ has reported score one unit larger than $\mathbf{x}^H$. The right inequality in (9) implies that $\mathbf{x}_j^L$ beats $\mathbf{x}^H$ in every state. Thus truthful binary and ternary reporting achieve $u_s(\mathbf{x}_j^L)$ in every state.

It remains to consider truthful expressive reporting with arbitrary hard filters. Requiring the liked coordinate to be present does not change which (feasible) item is preferred over one another. Requiring it to be absent makes no catalog item feasible, so the platform drops the hard filters and returns the expressive outcome without hard filters $\mathbf{x}^H$. Requiring a disliked coordinate to be present cannot remove $\mathbf{x}^H$, so it cannot make $\mathbf{x}_j^L$ beat $\mathbf{x}^H$ when both remain feasible. The only potentially useful hard filters are therefore requirements that one or both disliked coordinates be absent. If no disliked coordinate is required absent, the outcome is $\mathbf{x}^H$ in both states. If coordinate $r \in \{1, 2\}$ is required absent, then $\mathbf{x}_r^L$ is feasible and $\mathbf{x}^H$ is infeasible in state r , so the user obtains the good product in that state. In the other state, neither product satisfies the hard filter, so the platform drops the hard filter and returns $\mathbf{x}^H$. If both disliked coordinates are required absent, no product is feasible in either state and the platform then returns $\mathbf{x}^H$ in both states. Hence every truthful expressive plan with hard filters obtains the good product in at most one of the two states, while truthful binary and ternary obtain it in both states. Since $u_s(\mathbf{x}_j^L) > u_s(\mathbf{x}^H)$, the latter have strictly smaller expected loss. $\square$

C.3 Proof of Theorem 3

Proof overview. The proof first shows that every report used by Algorithm 1 remains within the allowed range. It then identifies a value of c for which the change in the linear revenue term at the user-optimal item is exactly offset by the change in reported match. Concavity of ℓ implies that no other item has a larger platform score. Finally, the proof bounds the effects of using a nearby grid point and of allowing item-specific errors in the linear revenue model.

Proof. Let $\mathbf{x}^* \in \arg \max_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{s}, \mathbf{x} \rangle$. If $\|\mathbf{p}\|_1 = 0$, the algorithm submits the truthful expressive report. For any item $\mathbf{y}$ with $u_s(\mathbf{x}^*) - u_s(\mathbf{y}) > \varepsilon$, the platform-score advantage of $\mathbf{x}^*$ under the truthful report is at least $\bar{\ell}' \Delta_\varepsilon$. The item-specific errors can shrink the log-revenue advantage of the user-optimal item over the other item by at most $\log((r_{\min} + \bar{\delta})/(r_{\min} - \bar{\delta}))$. Thus condition (8), read as a single truthful report with no grid error, implies that no bad item can be returned. Hence the theorem holds in this case. We now assume $\|\mathbf{p}\|_1 > 0$.

For every grid value $c \in [0, C]$ and every coordinate i ,

$$|\alpha s_i - c p_i| \leq \alpha S_i + C p_i \leq S_i \left(1 - \frac{\rho}{\ell'}\right) + \frac{p_i}{p_0 \ell'} \leq S_i,$$

so every report used by the algorithm lies in $\mathcal{S}$.

Because ℓ' is decreasing and the reported match of $\mathbf{x}^*$ under every report used by the algorithm is at most $\bar{m}_*$, we have

$$\ell'(\langle \alpha \mathbf{s} - c \mathbf{p}, \mathbf{x}^* \rangle) \geq \bar{\ell}'$$

for all $c \in [0, C]$. At $c = 0$,

$$c \ell'(\langle \alpha \mathbf{s} - c \mathbf{p}, \mathbf{x}^* \rangle) < \frac{1}{R_0(\mathbf{x}^*)},$$

while at $c = C$,

$$c \ell'(\langle \alpha \mathbf{s} - c \mathbf{p}, \mathbf{x}^* \rangle) \geq \frac{1}{p_0 \bar{\ell}'} \bar{\ell}' \geq \frac{1}{R_0(\mathbf{x}^*)}.$$

461 By continuity, there exists $c^* \in [0, C]$ such that

$$c^* \ell'(\langle \alpha s - c^* \mathbf{p}, \mathbf{x}^* \rangle) = \frac{1}{R_0(\mathbf{x}^*)}. \quad (10)$$

462 Fix any competitor $\mathbf{y}$. By concavity of the logarithm,

$$\log R_0(\mathbf{y}) - \log R_0(\mathbf{x}^*) \leq \frac{\langle \mathbf{p}, \mathbf{y} - \mathbf{x}^* \rangle}{R_0(\mathbf{x}^*)}.$$

463 By concavity of ℓ ,

$$\ell(\langle \alpha s - c^* \mathbf{p}, \mathbf{y} \rangle) - \ell(\langle \alpha s - c^* \mathbf{p}, \mathbf{x}^* \rangle) \leq \ell'(\langle \alpha s - c^* \mathbf{p}, \mathbf{x}^* \rangle) \langle \alpha s - c^* \mathbf{p}, \mathbf{y} - \mathbf{x}^* \rangle.$$

464 Adding the two inequalities and using (10) cancels the linear revenue term. The score difference
465 under R_0 between $\mathbf{y}$ and $\mathbf{x}^*$ is therefore at most

$$\alpha \ell'(\langle \alpha s - c^* \mathbf{p}, \mathbf{x}^* \rangle) \langle \mathbf{s}, \mathbf{y} - \mathbf{x}^* \rangle. \quad (11)$$

466 In particular, if $\mathbf{y}$ is bad, meaning $u_s(\mathbf{x}^*) - u_s(\mathbf{y}) > \varepsilon$, then

$$\text{score under } R_0 \text{ of } \mathbf{y} \leq \text{score under } R_0 \text{ of } \mathbf{x}^* - \alpha \bar{\ell}' \Delta_\varepsilon. \quad (12)$$

467 Let $\bar{c}$ be a grid point with $|\bar{c} - c^*| \leq \eta$. Since ℓ is one-Lipschitz,

$$|\ell(\langle \alpha s - \bar{c} \mathbf{p}, \mathbf{x} \rangle) - \ell(\langle \alpha s - c^* \mathbf{p}, \mathbf{x} \rangle)| \leq \eta \|\mathbf{p}\|_1$$

468 for every item $\mathbf{x}$. Therefore the score difference in (12) can shrink by at most $2\eta \|\mathbf{p}\|_1$ when using
469 $\bar{c}$ instead of c^* . Finally, changing from R_0 to $R = R_0 + \delta$ can shrink the log-revenue advantage of
470 the user-optimal item over the other item by at most

$$\log \frac{R_0(\mathbf{x}^*)}{R_0(\mathbf{x}^*) - \bar{\delta}} + \log \frac{R_0(\mathbf{y}) + \bar{\delta}}{R_0(\mathbf{y})} \leq \log \frac{r_{\min} + \bar{\delta}}{r_{\min} - \bar{\delta}},$$

471 where $r_{\min} = \min_{\mathbf{x} \in \mathcal{X}} R_0(\mathbf{x})$ and $r_{\min} > \bar{\delta}$. Condition (8) implies that, at the grid point $\bar{c}$, ev-
472 ery bad item has strictly smaller platform score using R than $\mathbf{x}^*$. Therefore the recommendation
473 observed at $\bar{c}$ is not bad. It is ε -optimal, and Algorithm 1 also returns an ε -optimal item because it
474 keeps the best observed recommendation by user utility. Since $\alpha = (\bar{\ell}' - \rho)/\bar{\ell}'$, the grid size satisfies

$$T = O\left(1 + \frac{C}{\eta}\right) = O\left(1 + \frac{\|\mathbf{p}\|_1}{p_0 \varepsilon \alpha (\bar{\ell}')^2}\right) = O\left(1 + \frac{\|\mathbf{p}\|_1}{p_0 \varepsilon \bar{\ell}' (\bar{\ell}' - \rho)}\right).$$

475 □

476 C.4 Proof of Theorem 4

477 **Proof overview.** The proof has two parts. The first construction uses exponentially many possible
478 hidden catalogs and shows that any bounded soft report can return the user-optimal item for at most
479 one of them. The second construction uses paired coordinates and chooses the outside option so that
480 purchase probability is close to one for every bounded report. Changing reported match then has
481 very little effect, and hard filters succeed only when they correctly identify almost all of the target's
482 hidden coordinates. If the hard filters are infeasible, dropping them does not help because the same
483 soft report is compared with products that still beat the target.

484 **Proof. No hard filters.** Fix any target loss sequence $\Delta_d = o(d)$. Choose an integer $k = k_d$
485 such that $k \rightarrow \infty$, $k = o(d)$, and $k > 2\Delta_d$. The construction uses three coordinate blocks: a
486 common liked block of size h_0 , a revenue block of size k , and a tail block of size m , with $h_0 =$
487 $2m + \lceil \max\{m_0, 0\} \rceil + 1$ and $h_0 + k + m = d$. Thus $m = \Theta(d)$ and $h_0 = \Theta(d)$.

488 Choose a code $\mathcal{C} \subseteq \{0, 1\}^m$ with minimum Hamming distance at least $2k$ and cardinality $|\mathcal{C}| =$
489 $\exp(\Omega(d))$. Such a code exists by a standard greedy packing argument because $2k = o(m)$. For
490 each hidden codeword $\mathbf{u} \in \mathcal{C}$, define the catalog

$$\mathcal{X}_{\mathbf{u}} := \{\mathbf{x}_{\mathbf{u}}^+\} \cup \{\mathbf{x}_{\mathbf{t}}^- : \mathbf{t} \in \mathcal{C}, \mathbf{t} \neq \mathbf{u}\}, \quad \text{where } \mathbf{x}_{\mathbf{u}}^+ = (\mathbf{1}^{h_0}, \mathbf{0}^k, \mathbf{u}), \quad \mathbf{x}_{\mathbf{t}}^- = (\mathbf{1}^{h_0}, \mathbf{1}^k, \mathbf{t}).$$

491 Set user preferences to 1 on the common block, -1 on the revenue block, and 0 on the tail block. Set
 492 $p_0 = 1$, set price $a > 0$ on each revenue-block coordinate, and set all other prices to zero. Choose
 493 a so that $\Gamma := \log(1 + ka) > k$; for instance, take $a = (e^{2k} - 1)/k$, which gives $\Gamma = 2k$. Then
 494 $\delta_x = 0$ for every item, and $\rho = a$ can be made arbitrarily large.

495 For catalog $\mathcal{X}_u$, the target x_u^+ has user match h_0 , while every decoy has user match $h_0 - k$. By
 496 the definition of h_0 , both matches are at least m_0 for all sufficiently large d . Hence every decoy has
 497 MNL utility loss at least $k/2 > \Delta_d$.

498 If the hidden codeword u is known, the target is inducible by the expressive report without hard
 499 filters that assigns -1 to every common-block coordinate, assigns -1 to every revenue-block coor-
 500 dinate, and assigns $2u_i - 1$ to the i th tail coordinate. Under this report, the reported score of every
 501 relevant item is far below the fixed outside option for large d , so ℓ' is bounded below by a positive
 502 constant on every interval. For every $t \neq u$, the reported-match advantage of the target over x_t^- is

$$k + \text{Ham}(u, t) \geq 3k.$$

503 Thus, by increasing the code distance constant if necessary, the target's log-purchase advantage is
 504 larger than Γ , and the target strictly maximizes the platform score.

505 Now fix an arbitrary expressive report without hard filters $\hat{s} \in [-1, 1]^d$. Let $t(\hat{s})$ be a codeword
 506 maximizing the tail inner product $\langle \hat{s}_{\text{tail}}, t \rangle$ over $t \in \mathcal{C}$, with ties broken deterministically. If the
 507 hidden codeword is $u \neq t(\hat{s})$, then the decoy $x_{t(\hat{s})}^-$ belongs to $\mathcal{X}_u$. The common block cancels, and
 508 by definition of $t(\hat{s})$,

$$\begin{aligned} \langle \hat{s}, x_u^+ \rangle - \langle \hat{s}, x_{t(\hat{s})}^- \rangle &= - \sum_{i \in \text{rev}} \hat{s}_i + \langle \hat{s}_{\text{tail}}, u - t(\hat{s}) \rangle \\ &\leq k. \end{aligned}$$

509 Because ℓ is one-Lipschitz, the target's log-purchase advantage over this decoy is at most k , while
 510 the decoy's log-revenue advantage is $\Gamma > k$. Therefore this report cannot return the target unless
 511 $u = t(\hat{s})$.

512 Draw u uniformly from $\mathcal{C}$. For any deterministic adaptive algorithm, as long as the target has not
 513 been returned, the response to a query $\hat{s}$ is the canonical decoy $x_{t(\hat{s})}^-$ whenever $u \neq t(\hat{s})$. Thus each
 514 query can return the target for at most one hidden codeword. After Q reports,

$$\Pr(\text{target has been returned}) \leq \frac{Q}{|\mathcal{C}|}.$$

515 On the complementary event, every observed item has loss greater than Δ_d . Hence

$$\mathbb{E}[\text{Loss}_Q] \geq \Delta_d \left(1 - \frac{Q}{|\mathcal{C}|} \right).$$

516 Since $|\mathcal{C}| = \exp(\Omega(d))$, every $Q = 2^{o(d)}$ report algorithm has expected loss at least $(1 - o(1))\Delta_d$.

517 **Arbitrary hard filters when purchase probability is close to one.** Let $d = 2m$, and group co-
 518 ordinates into pairs (A_j, B_j) for $j \in [m]$. Set $S_i = 1$ for every coordinate. Choose $k = k_d$ with
 519 $k > 2\Delta_d$ and $k = o(m)$, and set $K = \lfloor m/2 \rfloor$. For each subset $T \subseteq [m]$, let $x(T)$ be the one-hot-
 520 per-pair product that chooses A_j if $j \in T$ and B_j if $j \notin T$. A hidden set $U \subseteq [m]$ of size K defines
 521 the target $x(U)$, and the common lower-utility family is

$$\mathcal{D} := \{x(T) : |T| = K - k\}.$$

522 The catalog is $\mathcal{X}_U = \{x(U)\} \cup \mathcal{D}$. Set $s_{A_j} = 1$, $s_{B_j} = 0$, $p_{A_j} = 0$, $p_{B_j} = 1$, $p_0 = 1$, and $\delta_x = 0$
 523 for every item. Choose

$$m_0 = -m - B_d, \quad B_d = 3 \log d.$$

524 Every bounded report and every one-hot-per-pair product has reported score in $[-m, m]$. Therefore

$$\ell'(z) \leq \frac{1}{1 + \exp(B_d)} \leq e^{-B_d} \quad \text{for every } z \in [-m, m]. \quad (13)$$

525 The target has user match K , while every decoy has user match $K - k$. Since $m_0 = -m - B_d$, these
 526 matches are far above the outside option, and every decoy has MNL utility loss at least $k/2 > \Delta_d$
 527 for large d .

Fix an arbitrary report with hard filters and soft vector $\hat{s} \in [-1, 1]^d$. In the paired catalog, the hard filters can be represented by two sets $P, Q \subseteq [m]$: P is the set of pairs forced to choose A_j , and Q is the set of pairs forced to choose B_j . If the hard filters are internally inconsistent, or if the target does not satisfy them, then the target cannot be returned through the feasible set. If the feasible set is empty, the platform drops the hard filters and evaluates the same soft report over the full catalog; this is the case $P = Q = \emptyset$ below.

Suppose first that the target satisfies the hard filters and $|P| \leq K - k$. Then there exists $T \subseteq U$ with $|T| = K - k$ and $P \subseteq T$. Since $Q \cap U = \emptyset$, this T also avoids Q , so the decoy $x(T) \in \mathcal{D}$ satisfies the same hard filters. The target and this decoy differ in exactly k pairs, so their reported-match difference is at most $2k$. By (13), the target's advantage from the ℓ term is at most $2ke^{-B_d}$.

The decoy has k more B -coordinates than the target, so its log-revenue advantage is

$$\log \frac{1 + (m - K + k)}{1 + (m - K)} \geq c \frac{k}{m}$$

for a universal constant $c > 0$ and all large d , because $k = o(m)$ and $K = \lfloor m/2 \rfloor$. Since $B_d = 3 \log d$ and $m = \Theta(d)$, we have $2ke^{-B_d} < ck/m$ for large d . Thus this decoy strictly beats the target. The same argument applies after infeasible filters are dropped by taking $P = Q = \emptyset$.

Consequently, a query can return the target only if the target satisfies the hard filters and $|P| \geq K - k + 1$. For a uniformly random hidden set U among all K -subsets of $[m]$, any fixed P with $|P| \geq K - k + 1$ is contained in U with probability at most

$$\left(\frac{K}{m}\right)^{K-k+1} \leq 2^{-(K-k+1)} = \exp(-\Omega(d)).$$

Whenever the target is not returned, the platform returns an item from the common lower-utility family, either before or after infeasible filters are dropped. This response is determined by the report and not by the hidden set except on the exponentially small success event. A union bound over Q adaptive reports gives

$$\Pr(\text{Loss}_Q \leq \Delta_d) \leq Q \exp(-\Omega(d)) \implies \mathbb{E}[\text{Loss}_Q] \geq \Delta_d(1 - Q \exp(-\Omega(d))).$$

Thus every $Q = 2^{o(d)}$ report plan has loss at least $(1 - o(1))\Delta_d$ even with arbitrary hard filters. $\square$

C.5 Proof of Theorem 5

Proof overview. The construction sets $S_i = 1$ for every coordinate and $\delta_x = 0$ for every item. In the sparse construction, the useful coordinates form a hidden q -subset, and a report must name almost all of them to return the target. In the second construction, each block has four possible hidden choices but the soft report has only three values, so any fixed report fails on a constant fraction of blocks for most hidden catalogs. If the hard filters are infeasible, the same soft report is evaluated over the full catalog and the same comparisons continue to apply.

Proof. Adaptive sparse ternary reports. Fix a target loss sequence $\Delta_d = o(q)$. Choose $k = k_d$ such that $k \rightarrow \infty$, $k = o(q)$, and $k > 2\Delta_d$. Split the coordinates into two blocks $\mathcal{G}$ and $\mathcal{B}$ of size $m = \lfloor d/2 \rfloor$. Let $s_i = 1$ on $\mathcal{G}$ and $s_i = 0$ on $\mathcal{B}$. For each hidden set $U \subseteq \mathcal{G}$ of size q , the target is $x_U^+ = \mathbf{1}_U$. The common lower-utility family is

$$\mathcal{D} := \{\mathbf{1}_{A \cup C} : A \subseteq \mathcal{G}, |A| = q - k, C \subseteq \mathcal{B}, |C| = k\}.$$

The catalog is $\mathcal{X}_U = \{x_U^+\} \cup \mathcal{D}$. Give the $\mathcal{B}$ coordinates a small positive price $a > 0$ and the $\mathcal{G}$ coordinates zero price. Choose a small enough that, when U is known, the q -sparse ternary report assigning $+1$ to coordinates in U and zero elsewhere induces the target: the target has reported-match advantage at least k , while every decoy has only k small revenue coordinates.

The target has match q , and every decoy has match $q - k$. Since m_0 is fixed and $q - k \rightarrow \infty$, both matches are eventually above m_0 , and every decoy has MNL utility loss at least $k/2 > \Delta_d$.

Now fix an arbitrary q -sparse ternary report, including hard filters. Let $S \subseteq \mathcal{G}$ be the useful coordinates named favorably, meaning coordinates either hard-filtered to one or assigned positive soft

value. Then $|S| \leq q$. Suppose $|U \cap S| \leq q - k$. We show that the report cannot return the target. If the target satisfies the hard filters, choose $A \subseteq U$ of size $q - k$ containing all coordinates in $U \cap S$ and avoiding every coordinate hard-filtered to zero. Next choose $C \subseteq \mathcal{B}$ of size k avoiding every coordinate in $\mathcal{B}$ that is hard-filtered to zero or assigned negative soft value; this is possible for large d because the report names at most $q = o(d)$ coordinates while $|\mathcal{B}| = \Theta(d)$. The decoy $\mathbf{1}_{A \cup C}$ belongs to $\mathcal{D}$, satisfies the same hard filters, has weakly larger reported match than the target, and has strictly larger revenue. Hence it beats the target.

If the target does not satisfy the hard filters, then it cannot be returned through the feasible set. If the feasible set is empty and the platform drops the hard filters, the same construction without the hard-filter constraints gives a decoy in $\mathcal{D}$ with weakly larger reported match and strictly larger revenue. Thus dropping the filters also does not return the target. Therefore a report can return the target only for hidden sets satisfying $|U \cap S| \geq q - k + 1$. For a fixed S of size at most q , the number of hidden sets satisfying this condition is at most

$$M := \sum_{\ell=0}^{k-1} \binom{q}{\ell} \binom{m-q}{\ell},$$

where we have padded S to size q , which can only increase the count. The total number of hidden sets is $H = \binom{m}{q}$. Standard binomial estimates yield $\log(H/M) = \Omega(q \log(d/q))$ when $k = o(q)$ and $q = o(d)$. Now, draw U uniformly from all q -subsets of $\mathcal{G}$. Unless a query returns the target, the response is from the common lower-utility family or from the same common lower-utility family after infeasible filters are dropped, and hence does not reveal the hidden set except through the success event. Thus after T adaptive reports,

$$\Pr(\text{target has been returned}) \leq T \frac{M}{H}.$$

Since every non-target recommendation has loss greater than Δ_d ,

$$\mathbb{E}[\text{Loss}_T] \geq \Delta_d \left(1 - T \frac{M}{H}\right).$$

This proves that $\exp(\Omega(q \log(d/q)))$ reports are necessary for $o(q)$ expected loss.

Non-adaptive ternary reports. Fix a target loss sequence $\Delta_d = o(d)$. Let $d = 4m$, and partition the coordinates into m blocks of four coordinates, indexed by $(b, 1), (b, 2), (b, 3), (b, 4)$. Fix $\gamma \in (0, 1/4)$ and set

$$s_{b,1} = 1, \quad s_{b,2} = 1 - \gamma, \quad s_{b,3} = 1 - 2\gamma, \quad s_{b,4} = 1 - 3\gamma.$$

For each hidden word $u \in [4]^m$, let $\mathbf{x}(u)$ be the one-hot product that chooses coordinate u_b in block b . The catalog is

$$\mathcal{X}_u := \{\mathbf{x}(v) : v_b \geq u_b \text{ for all } b \in [m]\}.$$

Thus $\mathbf{x}(u)$ is the unique user-optimal item. Set revenues to increase slightly from left to right in each block, with the increments small enough that truthful expressive reporting still induces $\mathbf{x}(u)$ when u is known.

Fix a ternary report with hard filters. If the hard filters leave a nonempty feasible set for a hidden word u , let $A_b \subseteq [4]$ be the symbols allowed in block b . A symbol $i \in [4]$ is good in block b if

$$i \in A_b \quad \text{and} \quad \hat{s}_{b,i} > \max\{\hat{s}_{b,j} : j \in A_b, j > i\},$$

with the maximum over the empty set equal to $-\infty$. Scanning allowed symbols from right to left, each good symbol is a strict new record among the three values $-1, 0, 1$, so each block has at most three good hidden symbols.

If a hidden block symbol is not good, then any platform maximizer moves strictly to the right in that block: either the hidden symbol is infeasible, or some feasible rightward symbol has weakly larger reported value and strictly larger revenue. Therefore, if a report is bad in B blocks for hidden word u , every non-null returned product has match loss at least γB .

If the hard filters leave no feasible product, the platform drops them and uses the same ternary soft vector with no hard filters. This is the same record argument with $A_b = [4]$ in every block, so again

at most three hidden symbols per block are good. Hence, for every hidden word on which the report obtains loss at most Δ_d , the hidden word must be bad in fewer than

$$r_d := \left\lceil \frac{2\Delta_d}{\gamma} \right\rceil = o(m)$$

blocks for either the comparison with the submitted hard filters or the full-catalog comparison after infeasible filters are dropped. For a fixed report, the number of hidden words with fewer than r_d bad blocks is at most

$$\sum_{a < r_d} \binom{m}{a} 4^a 3^{m-a} \leq 4^m e^{-cm}$$

for some constant $c > 0$ and all sufficiently large m . Taking the union over the hard-filter and full-catalog interpretations only changes the constant. Thus a fixed ternary report succeeds on at most an $e^{-\Omega(m)}$ fraction of hidden words.

For a non-adaptive list of T reports and a uniformly random hidden word, the probability that any report obtains loss at most Δ_d is at most $T e^{-\Omega(m)}$. Since $m = \Theta(d)$, every non-adaptive $T = \exp(o(d))$ strategy has loss at least $(1 - o(1))\Delta_d$. This proves the non-adaptive lower bound. $\square$

D Experimental design and additional results

D.1 Overview

We instantiate the reporting problem on two real-world product domains—laptops and cars. In each domain, the platform has a hidden catalog of 32 products, each represented by $d = 10$ binary features. The platform uses the true catalog price when selecting a product, while the user observes only an approximate per-feature price model. We compare three reporting interfaces—filter-only, ternary, and expressive—under truthful reporting and under a careful LLM user that chooses its report strategically.

The experiment is single-shot. A user observes its preference vector $\mathbf{s}$, submits one report, and receives one recommendation. The platform ranks every filter-consistent product using

$$\log(\text{Price}(\mathbf{x})) + w \langle \hat{\mathbf{s}}, \mathbf{x} \rangle, \quad w \in \{1/3, 3\}. \quad (14)$$

Thus, a one-unit increase in reported match is equivalent to multiplying price by approximately 1.4 when $w = 1/3$ and by approximately 20 when $w = 3$. The two values of w compare a setting in which the platform is more willing to trade user match for price against one in which reported match has greater influence.

The experiments evaluate the paper’s main qualitative predictions in a deliberately demanding setting. The platform uses true product prices, which are not exactly additive in the binary features; the strategic user sees only a regression-based approximation; and the user does not observe the catalog or the products’ true prices. Even with this limited information and only one report, the results reproduce the main pattern: truthful expressive reporting can perform poorly when price has greater influence, while expressive reporting is particularly effective when reported match has greater influence. A careful user can also improve substantially over truthful soft reporting, although aggressive hard filters sometimes leave no feasible product.

D.2 Data sources and feature extraction

Laptops. We use a public dataset of 1,303 laptop specifications and prices maintained by 37Degrees.¹ Each row contains the manufacturer, model, product category, screen description, CPU string, RAM, storage configuration, GPU string, operating system, weight, and price in euros. We parse the text fields as follows:

- **RAM:** extract the numeric value from entries such as “8GB” and “16GB.”
- **Screen size:** extract the numeric screen diagonal after removing quotation marks.
- **Resolution:** extract horizontal and vertical pixel counts from strings of the form 1920x1080.

¹<https://github.com/37Degrees/DataSets/blob/master/laptops.csv>

- **CPU tier:** flag strings containing i7, i9, Xeon, Ryzen 7, or Ryzen 9 as high-performance CPUs.
- **Storage:** extract every capacity/type pair and sum SSD, HDD, and Flash storage after converting TB to GB.
- **GPU:** identify discrete GPUs using Nvidia, GeForce, GTX, RTX, Quadro, and AMD model strings, while excluding integrated AMD APU graphics.
- **Price:** convert European decimal formatting to numeric values and convert euros to dollars using a fixed factor of 1.1.

We exclude rows with missing or non-positive prices, weight above 5 kg, RAM outside {2, 4, 6, 8, 12, 16, 24, 32, 64} GB, or screen size outside [10, 18.5] inches. Table 1 gives the resulting binary features.

Table 1: Binary laptop features.

Feature	Rule	Interpretation
High RAM	$\text{RAM} \geq 16 \text{ GB}$	Power-user memory
Fast CPU	i7/i9/Xeon/Ryzen 7/9	High-performance processor
Large screen	$\text{Screen} \geq 15 \text{ inches}$	Desktop-replacement size
High-resolution display	$\text{Pixels} > 1920 \times 1080$	Above-FHD display
IPS panel	“IPS” in screen description	Panel quality
Touchscreen	“Touchscreen” in description	Touch input
Large storage	$\text{Total storage} \geq 512 \text{ GB}$	Storage capacity
SSD	SSD or Flash storage present	Storage technology
Dedicated GPU	Discrete Nvidia/AMD GPU	Graphics capability
macOS	Operating system is macOS	Platform choice

Cars. We use the Cooper Union car dataset, which contains 11,914 vehicles with make, model, year, engine specifications, transmission, drivetrain, vehicle size and style, market-category tags, fuel economy, popularity, and MSRP.² Text fields are converted to lowercase before matching, and numeric columns are parsed as numbers. We exclude observations with MSRP at most \$1,000 or above \$500,000, non-positive or missing horsepower, or highway MPG outside (0, 80]. Table 2 gives the ten binary features.

Table 2: Binary car features.

Feature	Rule	Interpretation
High horsepower	$\text{HP} \geq 250$	Performance powertrain
Many cylinders	$\text{Cylinders} \geq 6$	Above economy baseline
AWD/4WD	Drivetrain is AWD or 4WD	Drivetrain capability
Automatic	Automatic, DCT, or CVT	Transmission preference
Large vehicle	Vehicle size is “Large”	Body size
Fuel efficient	$\text{Highway MPG} \geq 30$	Fuel efficiency
Luxury/high-performance	Market category contains “Luxury”	Market segment
SUV/crossover	Style contains “SUV” or “Crossover”	Body type
Electric/hybrid	Fuel type contains “electric” or “hybrid”	Powertrain
Four or more doors	$\text{Doors} \geq 4$	Sedan/SUV versus coupe

D.3 Per-feature price model

In both domains, we estimate an approximate per-feature price vector from the full dataset, rather than from the selected catalog. We regress log price on the ten binary features using Ridge regression:

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X} + \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}, \quad \mathbf{y} = \log(\text{Price}). \quad (15)$$

The regularization coefficient is one. Ridge regularization reduces instability from correlated product features, such as high horsepower and many cylinders or fast CPUs and dedicated GPUs. The platform nevertheless ranks products using their *true* prices. The estimated feature contributions are shown only to the careful user as an approximation to how product features affect price.

²<https://www.kaggle.com/datasets/CooperUnion/cardataset>

Table 3: Feature-price regressions for laptops and cars. The normalized magnitude divides the absolute coefficient by the largest absolute coefficient within the domain.

Domain	Feature	Coef.	% Δ price	Normalized magnitude	Avg. price if 1	Avg. price if 0
Laptops	High RAM	+0.293	+34.0%	0.693	\$2,270	\$1,025
	Fast CPU	+0.423	+52.6%	1.000	\$1,772	\$868
	Large screen	-0.154	-14.2%	0.363	\$1,172	\$1,351
	High-resolution display	+0.236	+26.6%	0.557	\$2,141	\$1,138
	IPS panel	+0.076	+7.9%	0.180	\$1,547	\$1,115
	Touchscreen	+0.094	+9.8%	0.221	\$1,590	\$1,175
	Large storage	+0.091	+9.5%	0.216	\$1,434	\$1,072
	SSD	+0.329	+39.0%	0.779	\$1,450	\$725
	Dedicated GPU	+0.232	+26.1%	0.549	\$1,639	\$1,047
	macOS	+0.248	+28.2%	0.587	\$1,721	\$1,228
Cars	High horsepower	+1.200	+232.2%	1.000	\$63,283	\$20,756
	Many cylinders	-0.224	-20.1%	0.187	\$51,357	\$23,572
	AWD/4WD	+0.131	+14.0%	0.109	\$47,688	\$35,795
	Automatic	+0.599	+82.1%	0.499	\$43,903	\$26,512
	Large vehicle	-0.073	-7.0%	0.060	\$52,577	\$35,609
	Fuel efficient	+0.428	+53.5%	0.357	\$26,347	\$44,847
	Luxury/high-performance	+0.487	+62.7%	0.405	\$71,069	\$24,303
	SUV/crossover	+0.101	+10.7%	0.085	\$38,137	\$40,024
	Electric/hybrid	+0.381	+46.4%	0.317	\$49,800	\$39,553
	Four or more doors	+0.004	+0.4%	0.003	\$35,842	\$48,210

Table 3 reports the raw log-price coefficient, the associated percentage price change, its normalized magnitude, and the average price conditional on the feature. When a raw coefficient is negative, we complement that binary feature before presenting price contributions to the user. For example, “large screen” is replaced by “not a large screen.” This keeps the displayed feature-price contributions nonnegative without discarding the information in the regression.

D.4 Catalog construction

From each full dataset, we construct a diversified catalog of $32 \approx \lceil \sqrt{2^{10}} \rceil$ products.

1. **Deduplication.** Among products sharing the same ten-bit feature vector, retain a product from the most frequently occurring manufacturer. Break remaining ties by the most recent year and then the lowest price.
2. **Furthest-first diversification.** Initialize with the product whose continuous feature vector is closest in L^1 distance to the coordinate-wise median. Repeatedly add the product maximizing its minimum L^1 distance from the products already selected, until the catalog contains 32 items. This is the standard maximin diversification heuristic of Gonzalez [21]. Continuous product attributes are used for this step so that, for example, a 13-inch and a 15.6-inch laptop contribute proportionally to the diversity criterion even if both later map to binary features.

The selected catalog and each product’s true price are hidden from the user.

D.5 User preference distribution

We calibrate the marginal distribution of preference coordinate s_i from the empirical frequency $\hat{q}_i$ of the corresponding feature in the full product dataset. The interpretation is that the supply of a feature provides a rough proxy for its population-level desirability. We set

$$s_i = 2Z_i - 1, \quad Z_i \sim \text{Beta}(\hat{q}_i h, (1 - \hat{q}_i)h), \quad h = 2.5, \quad (16)$$

independently across coordinates. Thus $\mathbb{E}[s_i] = 2\hat{q}_i - 1$. The concentration parameter h controls heterogeneity: larger values concentrate users more tightly around the population mean, while smaller values place more mass near strong likes and dislikes. Table 4 records the feature frequencies used for calibration. The displayed frequencies are rounded; sampling uses the underlying empirical frequencies.

We draw 100 user preference vectors in each domain. The same 100 vectors are reused across all reporting interfaces, strategy families, and platform scores within that domain, so comparisons are paired at the user-type level. The independence assumption is a simplification: real preferences for features such as horsepower and cylinder count, or screen size and resolution, are likely correlated.

Table 4: Empirical feature frequencies used to calibrate preference distributions.

Laptop feature	Frequency	Car feature	Frequency
High RAM	17.0%	High horsepower	44.2%
Fast CPU	40.8%	Many cylinders	57.5%
Large screen	64.4%	AWD/4WD	31.6%
High-resolution display	9.7%	Automatic	75.0%
IPS panel	28.0%	Large vehicle	23.2%
Touchscreen	14.7%	Fuel efficient	28.6%
Large storage	45.4%	Luxury/high-performance	32.6%
SSD	70.5%	SUV/crossover	24.9%
Dedicated GPU	31.9%	Electric/hybrid	0.0%
macOS	1.6%	Four or more doors	70.0%

D.6 Reporting strategies

Truthful baselines. The truthful strategies are deterministic and require no language model.

- **Truthful filter-only.** Select the $k = 3$ coordinates with largest $|s_i|$. If $s_i > 0$, require the feature to be present; if $s_i < 0$, require it to be absent. Set the soft report to $\hat{s} = \mathbf{0}$.
- **Truthful binary.** Submit no hard filters and report $\hat{s}_i = \text{sign}(s_i)$ for every nonzero coordinate. Sign rounding exaggerates weak preferences and can therefore reduce the platform’s ability to trade a small loss in fit for a price increase.
- **Truthful expressive.** Submit no hard filters and report the full preference vector, $\hat{s} = s$.

Careful LLM user. Gemini 2.5 Flash Lite acts as a careful user. For each user type, interface, and platform score, the model receives:

- the true preference vector s together with natural-language feature descriptions;
- the platform’s exact product-selection rule;
- the approximate normalized per-feature price contributions from the Ridge regression;
- the approximate catalog size of 32 products; and
- an instruction to choose a report that maximizes the user’s true match, knowing that the platform evaluates the reported rather than the true preferences.

The LLM does not observe the catalog or the products’ true prices. It returns a structured JSON report $(\mathcal{F}_0, \mathcal{F}_1, \hat{s})$ satisfying the chosen interface: $\hat{s} = \mathbf{0}$ for filter-only, $\hat{s} \in \{-1, 0, 1\}^d$ for ternary, and $\hat{s} \in [-1, 1]^d$ for expressive. A separate report is generated for the two experimental values of w , allowing the careful user to account for the platform’s stated rule.

The following typeset template reproduces the substantive prompt; bracketed fields are filled separately for each user and treatment.

```

You are a {domain} user. There are {D} binary features (0 or 1).

Features and your preferences:
f0 ({feature description 0}): s0 = {s[0]}
f1 ({feature description 1}): s1 = {s[1]}
...
f{D-1} ({feature description D-1}): s{D-1} = {s[D-1]}

Your utility is  $\sum_i s_i * x_i$ . Maximize this.
You do NOT know the {domain} catalog. Submit ONE report.

You know the platform selects the product  $x$  maximizing

 $\log(\text{Price}(x)) + \{\text{match weight}\} * \langle \text{reported preferences}, x \rangle$ .

The approximate per-feature price contributions are:
f0: p0 = {p[0]}
f1: p1 = {p[1]}
...
f{D-1}: p{D-1} = {p[D-1]}.
```

Each one-unit increase in reported match has the same effect on the platform score as multiplying price by approximately {price multiplier}.

If the filters leave the catalog empty, the outcome receives the lowest user utility in the catalog.

Strategically choose your report to maximize YOUR true utility, knowing the platform uses your REPORTED preferences, not your true preferences.

[Interface-specific output instructions follow.]

The interface-specific instructions require filtered coordinates to have zero soft weight and remind the LLM that excessive filtering may leave no feasible products.

D.7 Platform response and evaluation

For a report $(\mathcal{F}_0, \mathcal{F}_1, \hat{s})$ and a coefficient $w \in \{1/3, 3\}$, the experimental platform returns

$$\hat{x} \in \arg \max_{x \in \mathcal{X}(\mathcal{F}_0, \mathcal{F}_1)} \{\log(\text{Price}(x)) + w\langle \hat{s}, x \rangle\}, \quad (17)$$

with ties broken uniformly. If the hard filters leave no feasible product, the trial is assigned the utility of the lowest-utility product in the realized catalog. This convention penalizes over-filtering and makes the feasibility rate directly interpretable.

For every trial, we record:

- the recommended product’s true user match $\langle s, \hat{x} \rangle$;
- the oracle match $\max_{x \in \mathcal{X}} \langle s, x \rangle$;
- the gap between the oracle and recommended match;
- the true price of the recommended product; and
- whether the submitted hard filters leave a nonempty feasible set.

The full design crosses three interfaces, two strategy families, and two platform scores, yielding $3 \times 2 \times 2 = 12$ cells per domain, plus the oracle benchmark. All reported means, standard deviations, medians, and feasibility rates are computed across the same 100 user types within each domain.

D.8 Results

Table 5 reports the full results. Smaller gaps are better for the user. The filter-only truthful baseline is identical under both platform scores because it sends no soft report. Figures 1 and 2 show the distribution of utility gaps.

Truthful reporting. When $w = 3$, truthful expressive reporting performs best in both domains: the mean gap is 0.009 for laptops and 0.147 for cars, compared with 0.234 and 0.348 for truthful binary reporting. This is the benchmark in which preserving cardinal preference information is valuable. When $w = 1/3$, the car results reverse. Truthful binary and truthful expressive reports both have mean gap 1.953, whereas truthful filter-only reporting has mean gap 1.124. The platform can use the soft reports to select higher-priced products that are worse matches; the coarser filter-only report reveals less information about which tradeoffs the user will tolerate.

Careful reporting. For cars when $w = 1/3$, the careful ternary and expressive users reduce the mean gap to 1.053 and 1.209, respectively, relative to 1.953 under either truthful soft report. Thus, even a single strategic report can partially correct for price incentives without observing the catalog. When $w = 3$, careful expressive reports retain a small median gap—zero for laptops and 0.016 for cars—but are somewhat worse on average than truthful expressive reports. The LLM occasionally over-corrects or over-filters when truthful reporting already closely aligns the platform’s recommendation with the user.

Differences across domains. The car catalog has larger price differences across feature combinations, and the contrast between truthful and careful reporting is correspondingly sharper. Laptop outcomes are more similar across strategy families when $w = 1/3$. The feasibility rates also show

Table 5: Car and laptop recommendation experiments. The outcome is the user’s match gap relative to the best product in the hidden catalog. Feasibility is the fraction of reports whose hard filters leave a nonempty feasible set.

w	Strategy	Interface	Laptops				Cars			
			Mean	Std.	Median	Feas.	Mean	Std.	Median	Feas.
1/3	Truthful	Filter-only	1.388	0.745	1.355	100%	1.124	0.767	0.985	100%
		Binary	1.065	0.633	1.034	100%	1.953	0.798	2.029	100%
		Expressive	1.271	0.649	1.383	100%	1.953	0.798	2.029	100%
	Careful	Filter-only	1.654	1.480	1.185	87%	1.117	1.215	0.737	87%
		Ternary	1.014	1.087	0.778	96%	1.053	0.983	0.855	98%
		Expressive	1.149	1.035	1.067	95%	1.209	0.951	1.024	99%
3	Truthful	Filter-only	1.388	0.745	1.355	100%	1.124	0.767	0.985	100%
		Binary	0.234	0.311	0.019	100%	0.348	0.379	0.247	100%
		Expressive	0.009	0.028	0.000	100%	0.147	0.265	0.000	100%
	Careful	Filter-only	1.682	1.416	1.263	88%	1.522	1.431	0.982	77%
		Ternary	0.480	0.932	0.099	96%	0.464	0.841	0.100	96%
		Expressive	0.221	0.864	0.000	96%	0.317	0.718	0.016	96%

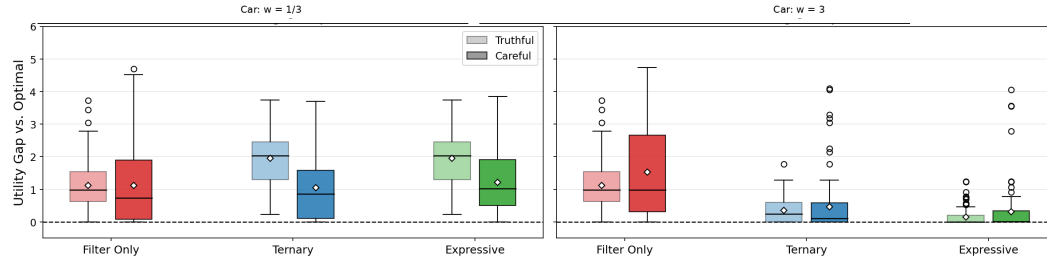


Figure 1: Car experiments. Each interface shows the distribution of the user’s match gap under truthful and careful reports across 100 user types. Smaller values are better.

785 a cost of strategic filtering: all no-filter truthful reports are feasible, while careful reports with hard
786 filters are feasible between 77% and 99% of the time. The lowest rate occurs for careful filter-only
787 reporting on cars when $w = 3$.

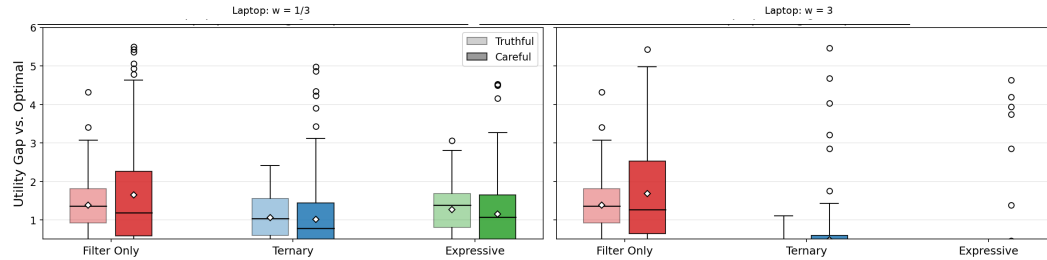


Figure 2: Laptop experiments. Each interface shows the distribution of the user’s match gap under truthful and careful reports across 100 user types. Smaller values are better.

788 D.9 Scope and limitations

789 The experiments are designed to illustrate the economic mechanism rather than to measure the multi-
790 round interaction bound of Algorithm 1. Each user submits one report, so the results show what a
791 one-shot careful agent can accomplish but do not evaluate the grid of reports used by the positive
792 theorem.

793 Several additional limitations are worth noting.

- 794 1. **Approximate price model.** True product prices are not additive in binary features and need
795 not be monotone in the feature set. The platform uses true prices, while the careful user sees
796 only a Ridge approximation. The reported improvement therefore depends on the approximation
797 preserving the product comparisons that matter for the user.
- 798 2. **Linear and independent preferences.** User match is linear in features, and preference co-
799 ordinates are sampled independently. Real product preferences can exhibit complementarities,
800 substitution, and correlation; for example, high horsepower and many cylinders or large screens
801 and high resolution may be jointly valued.
- 802 3. **Static datasets.** The data contain a single price and product snapshot. Promotions, time trends,
803 macroeconomic conditions, manufacturer strategy, and changes in product popularity are not
804 modeled.
- 805 4. **Limited catalog information.** The careful LLM receives neither the realized catalog nor a prior
806 over catalogs. Additional catalog information could improve its report, but would require a sep-
807 arate method for communicating that information compactly.
- 808 5. **Hard-filter penalty.** Empty feasible sets receive the lowest utility in the catalog. This makes
809 over-filtering costly and is intentionally conservative; alternative platform behavior after an in-
810 feasible query could change the performance of filter-heavy strategies.