STEER-ME: Evaluating LLMs in Information Economics

NARUN K. RAMAN, TAYLOR LUNDY, KEVIN LEYTON-BROWN, and JESSE PERLA, University of British Columbia, Canada

1 INTRODUCTION

Recently, Large Language Models (LLMs) have increasingly been deployed as decision-making engines, either directly acting as economic agents [Cai et al., 2023, Horton, 2023, Wang et al., 2023a] or as essential components within broader systems designed for economic decision-making [Shen et al., 2023, Wang et al., 2023b, Zhuge et al., 2023]. Despite constituting promising demonstrations of LLM capabilities, such systems have also exposed significant brittleness in LLM performance: models succeeding in one scenario often fail unpredictably in closely related contexts, suggesting at least some reliance on superficial pattern matching vs robust economic reasoning [e.g., Hendrycks et al., 2020, Ribeiro et al., 2020]. Despite this behavior, most existing economic benchmarks narrowly evaluate specific applications and do not rigorously assess the foundational strategic and computational reasoning skills necessary for reliable economic decision-making. We argue that before LLMs can be meaningfully evaluated or deployed in information economics, their capacities for these foundational reasoning capabilities must be systematically assessed.

To address this need, we developed the STEER benchmark [Raman et al., 2024], providing a comprehensive assessment of strategic reasoning foundational to economics. STEER was constructed by taxonomizing distinct "elements of economic rationality," ultimately comprising 64 strategic reasoning elements, including core concepts from game theory and foundational decision theory. Leveraging state-of-the-art LLMs, we systematically generated diverse questions across multiple domains (e.g., finance, medicine, public policy) and varying difficulty levels, creating an extensive and continually expandable dataset for benchmarking LLM economic reasoning. Building upon this methodology, we recently expanded the scope of our benchmark with STEER-ME [Raman et al., 2025], introducing 58 computational microeconomic reasoning elements—such as competitive market analysis, optimal consumption, and utility maximization. STEER-ME is significantly more challenging than STEER, not only because it demands precise mathematical computation but also because the evaluated concepts frequently require careful sequential reasoning, which directly underpins many critical scenarios in information economics.

In particular, STEER-ME evaluates elements essential for decision-making under uncertainty, such as correctly computing expected utility, managing state-contingent consumption, and evaluating the prices and risks inherent in uncertain economic environments. It further includes elements explicitly testing models' abilities to systematically update beliefs in response to new information, such as precisely applying Bayes' rule and optimally adapting decisions based on revised probabilities. These reasoning capabilities are foundational building blocks for canonical information economics scenarios—including costly information acquisition, adverse selection, Bayesian persuasion, and rational information disclosure—where economic agents must balance the expected value of information with its cost and dynamically update their beliefs to make rational decisions.

A natural way to see these building blocks working in concert is the element we call Dynamic Profit Maximization. A firm starts with capital K_1 and tomorrow faces a price p_2 that may be deterministic, uncertain, or inspectable at cost c. Before any uncertainty is resolved the agent chooses (i) whether to pay the inspection fee $I \in \{0, 1\}$ and (ii) a decision rule ϕ that maps the realized price into tomorrow's capital. Its objective can be written once for all three flavors as

$$\max_{I \in \{0,1\}, \phi} p_1 \texttt{output}(K_1) - \texttt{AdjCost}\left(\phi(p_2) - K_1\right) + \delta \mathbb{E}\left[p_2 \texttt{output}(\phi(p_2)) - Ic\right].$$

Manuscript submitted for review to the 25th ACM Conference on Economics & Computation (EC'24).

Deterministic:

I run a company that produces handmade wooden toys. The amount I produce is a function of the amount of capital (K) I put in, described by the function $4.73K^{0.99}$. In today's market, my toys sell for a price of 2.6 and I currently have capital $K_1 = 3.07$. I am trying to decide how much to grow my capital for tomorrow's market. I know the price my toys will sell at tomorrow is 2.25. I also know that I will incur a cost of growing my capital equal to $(K_1 - K_2)^2$. Lastly, my discount factor for the revenue acquired tomorrow is 0.17. If I want to maximize my profit how much should I increase my capital?

A. 1.68
B. 0.89 *[Correct Answer.]*
C. 1.11
D. 0.87

Uncertain Price:

I manage a company that produces eco-friendly packaging materials. The amount of packaging we produce depends on our level of capital (K), represented by the function $4.85K^{0.44}$. Currently, our products sell for a price of 9.18, and we have capital $K_1 = 1.74$. As we look towards tomorrow's market, I need to decide on the optimal increase in capital to maximize our profits. The price of our products tomorrow will follow the distribution price 3.37 with probability 0.31, price 1.92 with probability 0.11, price 9.89 with probability 0.57. Growing our capital will incur a cost given by $(K_1 - K_2)^2$, and any revenue earned tomorrow will be discounted by the factor 0.06. To maximize profit, how much should I grow our capital?

A. 0.31 *[Correct Answer.]*
B. 1.24
C. 0.27
D. 0.22

Costly Inspection:

As the lead engineer of a tech startup focusing on cutting-edge software solutions, I am evaluating our investment strategy in our development infrastructure, which directly influences our software capability described by $4.81K^{0.9}$. Our solutions are currently priced at 2.94, and we have a development capital of $K_1 = 7.36$ today. The pricing for our software projects tomorrow follows the distribution price $p_2 = 6.9$ with probability 0.24), price $p_2 = 3.14$ with probability 0.46), price $p_2 = 5.61$ with probability 0.3). We can choose to perform a market analysis costing 0.72, allowing us to predict tomorrow's exact market pricing. Without this analysis, our investment choices must rely on the price distribution. Expanding our development capital today incurs costs as defined by $(K_1 - K_2)^2$, and tomorrow's expected revenues will be affected by a discount factor 0.69. How should I proceed to optimize our expected profitability?

A. Skip the fee and set a single $K_2 = 1.53$
B. Pay the fee. If $p_2 = 1.3$ set $K_2 = 1.58$; if $p_2 = 2.5$, $K_2 = 1.89$; if $p_2 = 3.69$, $K_2 = 2.18$ *[Correct Answer.]*
C. Pay the fee. If $p_1 = 1.3$ set $K_2 = 1.9$; if $p_2 = 2.5$, $K_2 = 1.51$; if $p_2 = 3.69$, $K_2 = 2.61$
D. Skip the fee and set a single $K_2 = 1.91$

Fig. 1. This figure depicts three example questions in the Dynamic Profit Maximization element. In the top left is the deterministic flavor, where the tomorrow's price is given; the top right is the uncertain-price flavor, where the question gives a distribution over tomorrow's price; and lastly on the bottom is the costly inspection flavor, where the question allows the agent to pay some cost to identify tomorrow's price.

If tomorrow's price is revealed in advance (the deterministic flavor), p_2 is a constant, $c = 0$, and ϕ collapses to a single choice of K_2. If the price is uncertain but inspection is not offered, set $c = \infty$; the rule ϕ must again be constant, so the model must form an expectation over the three-point distribution before optimizing. In the costly-inspection flavor c is finite and ϕ may depend on p_2 whenever $I = 1$, turning the task into a value-of-information problem: the agent compares the expected profit from the price-contingent rule to that from a single prior-based K_2 and inspects only if the difference exceeds c. See Figure 1 for examples of each flavor as an instantiated question.

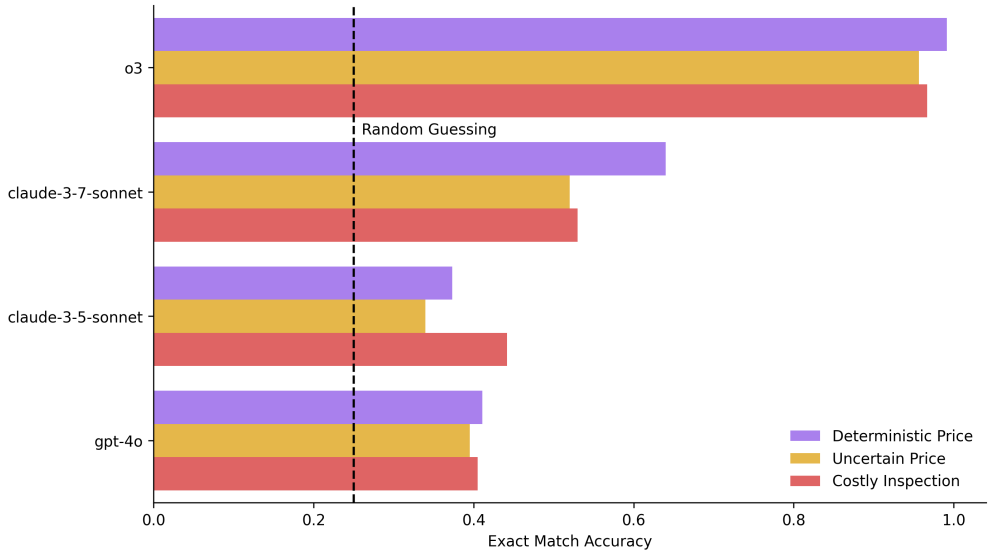

Fig. 2. Exact-match accuracy of each of the evaluated models across the Dynamic Profit Maximization element. Each colored bar denotes the performance of each model on the specific flavor of the element.

2 METHODS

We evaluated four state-of-the-art models: Anthropic's Claude 3.5 Sonnet and Claude 3.7 Sonnet and OpenAI's GPT 4o and o3. While Claude 3.5 Sonnet and GPT 4o are standard language models, Claude 3.7 Sonnet and o3 are reasoning models. Reasoning models are fine-tuned to conduct better chain-of-thought reasoning and o3 even has the ability to interleave code execution during reasoning. To standardize the evaluation, we present each model with the same set of 500 questions per flavor and allow each model to reason before coming to an answer. For the standard LLMs we decode with temperature 0 as that is considered optimal for high-fidelity reasoning tasks. For the reasoning models we sample with temperature 0.6 as is recommended by each provider. Each prompt lists the scenario and four labelled options; the model's task is to output the letter of its chosen option. We score responses with exact-match accuracy, counting an answer correct only when the returned letter matches the unique correct choice.

3 RESULTS

As can be seen in Figure 2, o3 was nearly flawless in the baseline deterministic-price flavor and loses only three to five percentage points as uncertainty and information acquisition are layered on. This robustness is unsurprising: o3 is the only model in the cohort with native code-execution, so once it forms the correct objective it can offload the calculus to the built-in Python sandbox—turning what is conceptually challenging but mechanically straightforward optimization into a trivial call to a solver. Claude 3.7 Sonnet Thinking ranks a clear second. Although it lacks code execution, the fine-tuned reasoning offers the model noticeably higher accuracy than both GPT-4o and Claude 3.5 Sonnet. However, we observed that it had the largest drop from deterministic to uncertain price out of all the evaluated models. That large drop suggests that Claude 3.7 Sonnet's boost in performance is susceptible to uncertainty in the information set. GPT-4o's performance mostly clusters in the low 0.4 range and exhibits little systematic difference between flavors. The flat profile implies that the step from deterministic optimisation to value-of-information reasoning did not make much

difference because it was already near random-guessing on the underlying calculus. Surprisingly, Claude 3.5 Sonnet showed an increase in performance for the costly inspection flavor.

This does not necessarily mean that Claude 3.5 Sonnet was particularly good at value-of-information reasoning, however. Because the costly-inspection question contains two "pay" menus and two "skip" menus, the evaluation can—and should—distinguish between two very different cognitive hurdles. First, a model must decide whether paying the fee is worthwhile. This is the harder piece of reasoning: it requires forming an expectation over tomorrow's prices, computing the marginal value of information, and comparing that value to the fee. Only after the correct branch is chosen does the model face the comparatively mechanical task of selecting the correct option for that branch. To tease these skills apart we re-graded every response in two layers: (1) Strategy accuracy: Did the model pick the correct branch (pay vs. skip)? (2) Conditional-K_2 accuracy: Given that branch, did the model pick the menu whose K_2 values satisfy the conditions within that branch?

Model	Strategy acc.	K_2 acc. | strategy
o3	0.97	0.98
Claude 3.7 Sonnet Thinking	0.70	0.82
Claude 3.5 Sonnet	0.54	0.80
GPT-4o	0.51	0.83

Table 1. Decomposing accuracy on the costly-inspection flavor of Dynamic Profit Maximization. A random guesser achieves 0.50 strategy accuracy and 0.25 overall accuracy (two correct branches × two correct menus).

Table 1 shows that o3's near-perfect score came from both layers: it almost always chooses the correct branch and almost never mis-computes K_2. Claude 3.7 Sonnet Thinking got the branch decision right four times out of five and, when it did, chose the correct capital level 82 % of the time. It trailed o3 but clearly outperforms non-reasoning models, suggesting fine-tuning for chain-of-thought improved both its value-of-information heuristic and its raw optimisation arithmetic despite lacking tool use. Claude 3.5 Sonnet and GPT-4o hovered between 51.26 to 54.10 % on strategy accuracy—essentially indistinguishable from random guessing—but still achieved mid-80 % conditional accuracy. This means they could solve the calculus when the branch is handed to them yet lacked a reliable rule for judging when information is worth its cost.

4 CONCLUSIONS

Our evaluations provide concrete evidence that weak proficiency on seemingly "low-level" elements—expectation formation, Bayes updates—manifests downstream as brittle, surface-level behavior on more elaborate economic tasks. Even competitive models such as GPT-4o and Claude 3.5 Sonnet were no better than guessing whether paying for information was worthwhile. In contrast, o3's native code-execution combined with fine-tuned reasoning delivered near-perfect performance across deterministic, uncertain, and costly-inspection variants, illustrating how tool integration can compensate for gaps in symbolic reasoning.

These findings underscore three broader points. First, information-economics problems expose a dimension of reasoning—valuing information and acting on contingent states—that is not well tested by classic optimisation benchmarks. Second, decomposing accuracy is essential for diagnosing where models actually fail; headline scores alone can be misleading when the multiple-choice structure embeds partial credit. Third, method matters: models that can off-load algebra to external solvers enjoy a large advantage, suggesting that benchmark design should explicitly distinguish conceptual errors from computational ones.

REFERENCES

Tianle Cai, Xuezhi Wang, Tengyu Ma, Xinyun Chen, and Denny Zhou. Large Language Models as Tool Makers. *CoRR*, abs/2305.17126, 2023. doi: 10.48550/ARXIV.2305.17126. URL https://doi.org/10.48550/arXiv.2305.17126.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding, 2020. URL https://arxiv.org/abs/2009.03300.

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.

Narun Krishnamurthi Raman, Taylor Lundy, Samuel Joseph Amouyal, Yoav Levine, Kevin Leyton-Brown, and Moshe Tennenholtz. STEER: Assessing the Economic Rationality of Large Language Models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Vienna, Austria, 2024. OpenReview.net. URL https://openreview.net/forum?id=nU1mtFDtMX.

Narun Krishnamurthi Raman, Taylor Lundy, Thiago Amin, Jesse Perla, and Kevin Leyton-Brown. Steer-me: Assessing the microeconomic reasoning of large language models, 2025. URL https://arxiv.org/abs/2502.13119.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/77c33e6a367922d003ff102ffb92b658-Abstract-Conference.html.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An Open-ended Embodied Agent with Large Language Models. *CoRR*, abs/2305.16291, 2023a. doi: 10.48550/ARXIV.2305.16291. URL https://doi.org/10.48550/arXiv.2305.16291.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona selfcollaboration. *arXiv preprint arXiv:2307.05300*, 1(2):3, 2023b.

Mingchen Zhuge, Haozhe Liu, Francesco Faccio, Dylan R. Ashley, Róbert Csordás, Anand Gopalakrishnan, Abdullah Hamdi, Hasan Abed Al Kader Hammoud, Vincent Herrmann, Kazuki Irie, Louis Kirsch, Bing Li, Guohao Li, Shuming Liu, Jinjie Mai, Piotr Piekos, Aditya A. Ramesh, Imanol Schlag, Weimin Shi, Aleksandar Stanic, Wenyi Wang, Yuhui Wang, Mengmeng Xu, Deng-Ping Fan, Bernard Ghanem, and Jürgen Schmidhuber. Mindstorms in Natural Language-based Societies of Mind. *CoRR*, abs/2305.17066, 2023. doi: 10.48550/ARXIV.2305.17066. URL https://doi.org/10.48550/arXiv.2305.17066.