# A Monte-Carlo Sampling Framework For Reliable Evaluation of Large Language Models Using Behavioral Analysis

**Davood Wadi**
University Canada West
Vancouver, BC, Canada
davood.wadi@ucanwest.ca

**Marc Fredette**
HEC Montreal
Montreal, QC, Canada
marc.fredette@hec.ca

## Abstract

Scientific evaluation of Large Language Models is an important topic that quantifies any degree of progress we make with new models. Even though current LLMs show high level of accuracy on benchmark datasets, the single-sample approach to evaluating them is not sufficient as it ignores high entropy of LLM responses. We introduce a Monte-Carlo evaluation framework for evaluating LLMs that follows behavioral science methodologies and provides statistical guarantees for estimates of performance. We test our framework on multiple LLMs to see if they are susceptible to cognitive biases. We find significant effect of prompts that induce cognitive biases in LLMs, raising questions about their reliability in social sciences and business.

## 1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language tasks, from question answering to creative writing, reasoning, and code generation. Despite the strong focus of current literature on mathematics, coding, and general question answering capabilities of Large Language Models, LLM use is expanding well beyond these areas. LLMs are increasingly integrated into critical applications across diverse fields such as healthcare (Clusmann et al., 2023; Nazi and Peng, 2023; Singhal et al., 2025), education (Dong et al., 2024; Gan et al., 2023; Wang et al., 2024), and decision support systems (Xu et al., 2024; Benary et al., 2023; Li et al., 2022). In particular, there is growing interest in the application of Large Language Models in social sciences. Consumer behavior researchers are exploring how LLMs can simulate and replace human participants (Aher et al., 2023). There is unprecedented reliance on AI agents in making consumption decisions (Dellaert et al., 2020).

Despite this interest from researchers and consumers, the scientific literature has not addressed two major gaps in our understanding of LLM behavior in social sciences settings. First, there is an absence of a proper evaluation framework for social sciences problems that properly accounts for high entropy of LLM responses. Second, there is a lack of LLM evaluation beyond the standard benchmark datasets on coding, mathematics, and general questions answering. To address the first gap, we define a Monte-Carlo evaluation framework to gain a reliable estimate of LLM performance. This framework accounts for the high level of entropy (MacKay, 2003) in LLM responses, which is neglected by the standard evaluation practice. To address the second gap, we formulate an experimental setup inspired from consumer behavior literature that allows scientific testing of LLMs' performance in consumption settings. We focus specifically on the problem of pricing in economics and business.

In our investigation, we ground our analysis in consumer choice theory by drawing on established models from behavioral economics, namely stochastic utility theory (Manski, 1977) and prospect theory (Tversky and Kahneman, 1974), which predict that consumer decisions stem from underlying latent utilities and that choices vary according to risk preferences, perceived value, and budget constraints. By analogously modeling the magnitude of maximum willingness-to-pay (WTP) elicited from LLMs, we evaluate whether these models, which have traditionally been applied to human decision-makers, hold predictive power for artificial decision-making systems.

We further interrogate whether LLM responses reflect systematic deviations from traditional rational behavior, such as anomalies in consistency and anchoring effects (Strack and Mussweiler, 1997; Ariely et al., 2003), by comparing the simulated choice patterns against norms predicted by economic theory.



*Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9414–9432
November 4-9, 2025 

Simultaneously, our study delves into emergent behavior using complex decision patterns that arise from the high-dimensional, data-driven learning process of LLMs, despite the absence of an explicitly programmed decision-making module. Specifically, we aim to uncover whether the LLMs, when tasked with consumer-like evaluations (e.g., Google's Project Mariner; DeepMind (2025)), produce non-linearities or clustering in willingness-to-pay that mirror cognitive biases or contextual influences observed in human subjects.

This article makes two main contributions. First, we identify and formalize the limitations of single-sample evaluation approaches for high-entropy LLMs, and introduce a Monte-Carlo sampling framework for LLM evaluation that provides statistical reliability metrics beyond accuracy, which can be applied to general LLM evaluation as well as social sciences and business. Second, we show limitations of LLMs in uncertain decision making, by quantifying the between- and within- LLM variance as well as their susceptibility to cognitive biases.

## 2 Related work

**Consumer behavior and artificial intelligence** Consumers are increasingly utilizing artificial intelligence (AI) models in their everyday consumption. There is a rich literature on how consumers' respond to AI-generated outputs. Artificially Intelligent Voice Assistants (AIVAs) are commonly used in everyday purchase decisions (Dellaert et al., 2020). In their decision-making, consumers tend to trust AI models to the point of "overdepending" on them even when those the responses are suboptimal (Banker and Khetani, 2019). Specifically, in the context of online shopping, consumers rely on AI decision making without much thought (Jain et al., 2024). There is also an increase in intelligent LLM-based agents that seek to replace humans in making purchase decision (e.g., Google's Project Mariner (DeepMind, 2025))

**Large Language Model bias** As Large Language Models continue to improve in natural language generation and consumers increase their trust and reliance on AI-generated decisions, it is imperative to understand potential LLM biases when it comes to decision making.

In recent years, there has been an emergence and growth in research on algorithmic bias. Algorithmic bias refers to a phenomenon where a machine learning model displays similar social patterns as the ones embedded in the data used to train the model (Johnson, 2021). Prior research has investigated various cases where LLMs show social biases (Gallegos et al., 2024). This phenomenon has been mostly attributed to the implicit and explicit biases present in the training data of the LLMs (Johnson, 2021) and the uncurated nature of the training data (Gallegos et al., 2024). Another factor identified as the source of social biases in LLMs is the benchmark datasets used to fine-tune these models. Gallegos et al. (2024) argue that LLMs are optimized on datasets that do not represent the broad population that will end up using these models.

**Cognitive biases in Large Language Models** Cognitive biases have long been one of the main points of focus for social sciences researchers (Haselton et al., 2015). They refer to systematic biases that deviate from rational decision making (Dowling et al., 2020). Many behavioral biases have been studied in consumer behavior including framing effects (Lee et al., 2015; Cheema and Patrick, 2008; Diehl, 2005; Morwitz et al., 1998; Levin and Gaeth, 1988; Yang et al., 2013; Cox and Cox, 2001), overgeneralization (Andrews et al., 1998), overconfidence (Soman, 1998; Lambrecht and Skiera, 2006) and anchoring effects (Adaval and Wyer Jr, 2011; Ariely et al., 2003). Prior research has investigated the presence of cognitive biases in Large Language Models (Ross et al., 2024; Jones and Steinhardt, 2022; Macmillan-Scott and Musolesi, 2024; Echterhoff et al., 2024). For code generation, Jones and Steinhardt (2022) tested two LLMs on some of the most common cognitive biases including anchoring bias and framing effects. The article provides indication of cognitive biases in LLMs for coding by showing how prompting LLMs can negatively affect their performance by introducing cognitive biases. Even though these articles provide helpful indication of the presence of cognitive biases in LLMs, their methodological approach uses single-sample estimation to analyze LLM behavior, which ignores high entropy of LLMs and does not provide a complete picture for stochasticity of LLM responses.

## 3 Methodology

The standard approach to evaluating LLMs involves measuring the "pass@1" accuracy (i.e., the correctness of a single response to a given prompt) across benchmark datasets (Achiam et al., 2023;

Team et al., 2023; Meta, 2024; Guo et al., 2025; Hurst et al., 2024). While this methodology provides a basic assessment of model capabilities, it fundamentally ignores the intrinsic variability in LLM outputs. Modern LLMs operate with non-zero temperature settings that introduce controlled randomness into their generation process (Chen et al., 2021; Su et al., 2022), allowing them to produce more natural, creative, and human-like responses. This randomness is desirable for many applications, to the point that almost all LLMs use non-zero temperature. This non-zero temperature creates a critical challenge for model evaluation. The same model, given identical inputs, can produce substantially different outputs from one inference to the next.

To illustrate this, consider an LLM that sometimes exhibits strong anchoring bias in pricing judgments but at other times provides rational market-based valuations in response to identical prompts. Standard single-sample evaluation of the LLM would categorize this model as either biased or rational depending solely on which single response was sampled. Using our Monte-Carlo framework (Algorithm 2), we sample multiple responses using the same LLM and prompt. By analyzing the aggregated behavior of the LLM, we statistically measure the reliability and the rationality of the LLM, by obtaining the model's true probabilistic behavioral profile. This approach would allow us to make informed decisions when utilizing LLMs as decision aids in sensitive consumption choices.

## 4 Statistical framework

Here we present the statistical framework. The proofs to the theorem are presented in the Appendix A.2.

Let $f(p, y, r)$ denote an evaluation function that maps a prompt $p \in \mathcal{P}$, the correct response $y$, and a sampled response $r \sim \mathcal{M}(p; T)$ from a stochastic language model $\mathcal{M}$ at non-zero temperature $T$, to a real-valued score. Define the true model behavior as the expectation:

$$v(\mathcal{M}; p, y) := \mathbb{E}_{r \sim \mathcal{M}(p;T)} \left[ f(p, y, r) \right]. \quad (1)$$

Let $\{r_k\}_{k=1}^{K}$ be $K$ i.i.d. samples from $\mathcal{M}(p; T)$, and define the empirical Monte Carlo estimator

$$\hat{v}_K(\mathcal{M}; p, y) := \frac{1}{K} \sum_{k=1}^{K} f(p, y, r_k). \quad (2)$$

Then, the Monte-Carlo evaluation framework has three main properties. First, the Monte-Carlo sample evaluation, $\hat{v}_K(\mathcal{M}; p, y)$, is an unbiased estimator of the true population evaluation. In the context of LLM evaluation, this means that averaging multiple model responses to the same prompt provides an unbiased estimate of the model's true behavior, accounting for the inherent variability introduced by non-zero temperature sampling. Second, the between-sample variance decays to 0 as $K$ becomes large enough. Third, the Monte-Carlo sample asymptotes to a normal distribution, $\mathcal{N}\left(\mu, \frac{\sigma^2}{K}\right)$. This allows us to define confidence intervals for LLM predictions and evaluation metrics.

**Theorem 1**

$$\mathbb{E}[\hat{v}_K(\mathcal{M}; p, y)] = v(\mathcal{M}; p, y) \quad (3)$$

**Theorem 2** Let $r_1, \ldots, r_K \overset{\text{i.i.d.}}{\sim} \mathcal{M}(p; T)$ be i.i.d. samples from the stochastic LLM at temperature $T$. Given the Monte-Carlo sample evaluation, $\hat{v}_K$, the Monte-Carlo sample variance asymptotes to zero as $K \to \infty$,

$$\lim_{K \to \infty} \mathrm{Var}[\hat{v}_K] = 0, \quad (4)$$

where $\sigma^2$ is the variance of the random variable $f(p, y, r_k)$.

**Corollary** The normalized estimator follows a normal distribution as $K \to \infty$:

$$\left[\hat{v}_K(\mathcal{M}; p, y) - v(\mathcal{M}; p, y)\right] \sqrt{K} \xrightarrow{d} \mathcal{N}(0, \sigma^2), \quad (5)$$

### 4.0.1 Choosing the value of K

To determine the optimal number of Monte-Carlo samples $K$ in our framework, we use power analysis for the F-test in an ANOVA design, which ensures sufficient statistical power to detect small effect sizes (i.e., Cohen's $f \leq 0.10$) when comparing expected scores $v(\mathcal{M}; p, y)$ across $g$ groups (e.g., different LLM behaviors or prompts). In a balanced one-way ANOVA with $g$ groups and $K$ samples per group (total $N = gK$), the non-centrality parameter is $\lambda = gKf^2$, with degrees of freedom $\mathrm{df}_1 = g - 1$ and $\mathrm{df}_2 = g(K - 1)$. The power $1 - \beta$ is the probability that the non-central F-distribution $F(\mathrm{df}_1, \mathrm{df}_2, \lambda)$ exceeds the critical value $F_{\text{crit}} = F^{-1}(1 - \alpha; \mathrm{df}_1, \mathrm{df}_2)$ for significance level $\alpha$. Solving for the minimal

$K$ involves finding the smallest integer such that $1 - \beta \geq 1 - F(F_{\text{crit}}; \text{df}_1, \text{df}_2, \lambda)$, where $F(\cdot; \cdot)$ is the cumulative distribution function of the non-central F. This can be efficiently computed via binary search as detailed in Algorithm 1. This derivation balances precision for small effects with computational efficiency (see Appendix A.2.4 for implementation details).

## 5 Hypothesis development

A large number of consumers rely on AI agents to help them with their pricing choice decision, in many cases without giving the response of the AI agent much thought (Jain et al., 2024). In this case, we know little about how LLMs behave in face of market-related decisions. Do they act rationally, or do they fall prey to the same marketing manipulations as humans? To address these questions, we develop the following hypotheses.

$\boldsymbol{H1}$ The willingness-to-pay of Large Language Models is different from the actual market price.

$\boldsymbol{H2}$ Large Language Models vary in their price estimation capabilities, with some exhibiting systematically larger absolute price deviations from actual market prices in their willingness-to-pay.

**Anchoring bias** Anchoring bias occurs when individuals rely heavily on an initially presented value (i.e., the anchor) when making subsequent judgments. In consumer contexts, this manifests as higher willingness to pay (WTP) after exposure to high price anchors and lower WTP after exposure to low price anchors, regardless of the product's list price (Tversky and Kahneman, 1974; Ariely et al., 2003). Strack and Mussweiler (1997) explain this phenomenon through selective accessibility, whereby the human brain selectively retrieves certain stored information activated by the context of the irrelevant number.

Since Large Language Models (LLMs) have been shown to replicate human biases (Johnson, 2021), we hypothesize that LLMs will be prone to anchoring biases. This is counter intuitive because LLMs have been praised for their ability to solve complex tasks such as answering difficult questions in physics, chemistry, and biology. Moreover, LLMs show impressive attention to detail that enables them to debug code and avoid spelling errors. Thus, one could argue that LLMs would be immune to cognitive biases. However, we hypothesize that cognitive biases are embedded in the training data of the LLMs, similar to social biases (Johnson, 2021). Hence, LLMs would tend to replicate the same type of judgemental errors, despite their high level of attention to detail.

$\boldsymbol{H3}$ Large Language Models are susceptible to anchoring effects, whereby high (low) anchoring manipulation leads to higher (lower) willingness to pay.

$\boldsymbol{H4}$ The effect of anchoring manipulation on willingness to pay is moderated by the Large Language Model used.

There is the assumption that LLMs improve from each generation to the next. However, the main focus of researchers is improvements in benchmark datasets, which is composed mainly of coding, mathematics, and logical reasoning tasks (Hurst et al., 2024; Achiam et al., 2023).

$\boldsymbol{H5_\emptyset}$ Newer generation LLMs perform at least as well as older generation LLMs in their price prediction accuracy.

$\boldsymbol{H5_A}$ Newer generation LLMs perform worse than older generation LLMs in their price prediction accuracy.

## 6 Experiments

This section details our experimental methodology for evaluating LLM susceptibility to anchoring bias in consumer decision-making contexts using our Monte-Carlo framework. To demonstrate the effectiveness of our framework, we introduce a cognitive bias experiment drawn from consumer behavior and behavioral economics literature (Tversky and Kahneman, 1974; Dowling et al., 2020; Kahneman, 2002; Ariely et al., 2003). This experiment challenges LLMs with prompts designed to elicit anchoring (Ariely et al., 2003; Ahmetoglu et al., 2014; Santana et al., 2020). This experiment serves as an effective case study for the proposed evaluation framework, revealing how models may exhibit susceptibility to cognitive biases depending on sampling randomness.

### 6.1 Design

We designed a two-factor (anchoring: high, low, and control $\times$ LLMs) factorial experiment to test whether different LLMs exhibit this systematic irrational behavior and to quantify its reliability across different models. We selected 6 consumer products (Ariely et al., 2003) across diverse categories with well-established market prices. The products were chosen from Amazon.com's bestsellers list. The goal was to replicate the original experiment using

LLMs. Following Ariely et al. (2003), we ensured the mean list price is close to USD 55 (Table 3). For each product, we created three experimental conditions. For the high (low) anchor condition, we told the LLM that its Social Security Number is 987-65-4395 (987-65-4315), and asked whether it would buy the product for a dollar amount equal to the last two digits of its Social Security Number (i.e., $95 or $15) (Ariely et al., 2003). We also used a control group without any anchors. Next, we asked the LLM its willingness-to-pay (WTP) for the product, which serves as our dependent variable.

We sampled 100 responses for each experimental condition and tested models from OpenAI, Meta, and Anthropic [1].

### 6.2 Measures

**Common sense price** To see whether the LLM is providing a common sense price for each product, we collected product information from Amazon.com for each one of the products. We collected a dataset for each product keyword (e.g., paper towels) up to 10 pages of the top relevant products. [2] Let $p_{\min}(k)$ and $p_{\max}(k)$ represent the minimum and maximum observed prices for a given product category $k$, as derived from Amazon's dataset. For a specific LLM $m$ and sample response $i$, the reported WTP for a product in category $k$ is denoted as $WTP_{i,m}(k)$.

To determine whether the WTP from a particular LLM aligns with observed market prices, we define a binary indicator variable $I_{i,m}(k)$.

$$I_{i,m}(k) = \mathbf{1}(p_{\min}(k) \leq WTP_{i,m}(k)) \times \mathbf{1}(WTP_{i,m}(k) \leq p_{\max}(k)) \quad (6)$$

The Common-Sense Validity Rate can be derived as follows

$$\text{CSVR}(m,k) = \frac{1}{N}\sum_{i=1}^{N} I_{i,m}(k)$$

which measures the proportion of instances where the willingness to pay (WTP) reported by the LLM for products in a specific category falls within the observed market price range.

**Absolute price deviation** To further assess the accuracy of the LLMs' willingness-to-pay (WTP) estimates, we compute the absolute price deviation (APD) between the WTP and the actual list price of a product. This metric quantifies the magnitude of deviation regardless of direction (over- or under-estimation).

Let $p_{\text{list}}(k)$ represent the actual list price of a product in category $k$. For a specific LLM $m$ and instance $i$, we define the absolute price deviation as

$$APD_{i,m}(k) = |WTP_{i,m}(k) - p_{\text{list}}(k)|$$

### 6.3 Results

First, we test model variability in willingness-to-pay (WTP) predictions for the same prompt. We use coefficient of variation, $\frac{SD}{M}$. Figure 1 reveals substantial variability in WTP predictions across different model-product combinations, as evidenced by coefficient of variation values ranging up to 0.44. This wide range indicates that for most models, the predicted prices differ markedly under the same prompt and product conditions. Such high variability verifies the limitation of standard metrics like pass@1 performance that sample only a single output per prompt. pass@1 fails to reflect the inherent uncertainty and distribution of model predictions, providing an incomplete and potentially misleading assessment of model behavior. In contrast, the Monte-Carlo sampling approach, effectively captures the full variability of models' responses.

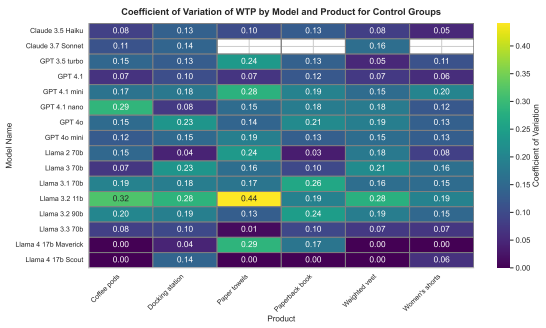


Figure 1: Coefficient of Variation of WTP by model and product for the control group

[1] The data collection for anthropic–claude-3-7-sonnet could not complete because the Anthropic server became overloaded after collecting 888 responses. Moreover, meta-llama2-70 has 1788 out of the 1800 possible responses because for the 12 missing responses it did not followed the response schema asking it for a floating point number. To ensure fairness and reflect real-world performance, we did not attempt to recollect or top up responses for these models, preserving the integrity of our original sampling procedure.

[2] Since for a given keyword Amazon.com returns more than 90,000 results, we limited the dataset to the top 10 pages of relevant products.

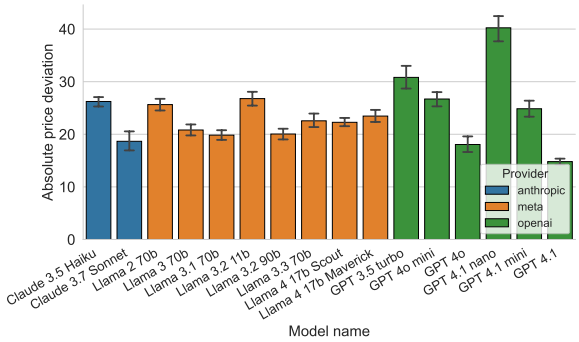


Figure 2: Absolute price deviation for the models tested in the control group

### 6.3.1 Common-sense pricing by LLMs

We test whether LLMs make common-sense price predictions for each product. For the control group, we measure the Common Sense Validity Rate (CSVR) (6.2) across product categories and model providers. For each product, we compare LLMs' willingness-to-pay with the price range of products in the same category on Amazon.com. Wilson score interval shows that for all product categories, LLMs' willingness-to-pay is within the market price range with a CSVR close to 100% (Figure 6). This shows that while LLMs might struggle to estimate exact prices for various products, their pricing recommendations are generally in a reasonable range for that product's category.

To gain insight into the magnitude of pricing accuracy, we look at the absolute price deviation (6.2) of willingness-to-pay (WTP) and the actual market price (List Price) in the control group. Figure 2 shows significant variations in pricing accuracy of different LLMs. Overall, we see more accurate pricing of larger and newer models. For Anthropic, *Claude-3.7-Sonnet* has significantly lower absolute price deviation than its predecessor, *Claude-3.5-Haiku*. For Meta, *Llama-2-70B* (oldest) and *Llama-3.2-11B* (smallest) show significantly larger absolute price deviation compared to their larger and newer counterparts. Lastly, for OpenAI, *GPT 4.1 nano* shows significantly larger absolute price deviation compared to the larger *GPT 4o* and *GPT 4.1*. This indicates that different LLMs have varied accuracy when it comes to making common sense price predictions, which can affect their reliability in different business and social sciences applications.

Next, we test H1 by comparing the performance of the LLMs in predicting product prices without any anchoring manipulations. Using a one-sample t-test, we compare the difference between the LLMs' willingness-to-pay (WTP) and the actual product's list price. We observe that there is a significant difference between the WTP of LLMs and the actual list price ($t(9295) = -23.15$, $p<0.001$, $M = -6.98$, $SD = 29.06$, $d = -0.24$). This supports H1. We see that the average WTP of LLMs is \$6.98 lower than the actual list price of the product.

To test H2, we investigate the effect of LLMs tested on absolute price deviation of willingness-to-pay (WTP). Table 6 shows the descriptive statistics of absolute price deviations for the LLMs tested. One-way ANOVA (Table 7) shows significant effect of the LLM used for generating willingness-to-pay (WTP) on the absolute price deviation of WTP and the list price, $F(15, 27860) = 9.07$, $p < 0.001$. This supports H2 and shows that the differences in Figure 2 are statistically significant. This indicates that LLMs have significantly different levels of capabilities when it comes to making pricing judgements.

### 6.3.2 Susceptibility to anchoring manipulation

We now test the effect of anchoring manipulations on pricing accuracy of LLMs. Table 11 shows the correlation between WTP and SSN for each LLM compared to human average. LLMs (average correlation = 0.697) tend to show stronger susceptibility to anchoring manipulation compared to humans (average correlation = 0.388). To test H3, we investigate the effect of anchoring manipulations on the willingness-to-pay (WTP) of the LLMs. One-way ANOVA (Table 8) shows significant effect of anchoring on LLMs' willingness-to-pay, $F(2, 27873) = 412.56$, $p < 0.001$. Regression analysis (Table 1) shows a significant positive effect of high anchoring ($B = 31.62$, $SE = 1.68$, $t(27873) = 18.84$, $p < 0.001$, 95% CI [28.33, 34.91]) and a significant negative effect of low anchoring ($B = -15.70$, $SE = 1.68$, $t(27873) = -9.35$, $p < 0.001$, 95% CI [-18.98, -12.41]) on WTP. This supports H3 and shows that Large Language Models are susceptible to anchoring biases and can be manipulated to produce higher (lower) willingness to pay using anchoring messages in the system prompt.

We now investigate the moderating effect of LLMs on the relationship between anchoring and absolute price deviation. Two-way ANOVA shows a significant interaction between the anchoring manipulation and the LLM tested ($F(30, 27828) = 12.75$, $p < 0.001$), which indicates different LLMs

Table 1: OLS regression results for the effect of anchoring of willingness-to-pay (WTP)

| | B | SE | t | p | 95% CI |
|---|---|---|---|---|---|
| Intercept | 46.334 | 1.187 | 39.05 | < .001 | [44.01, 48.66] |
| Anchoring[high] | 31.617 | 1.678 | 18.84 | < .001 | [28.33, 34.91] |
| Anchoring[low] | -15.696 | 1.678 | -9.35 | < .001 | [-18.98, -12.41] |

have varied susceptibility to anchoring manipulation (H4 supported).

### 6.3.3 Generational improvements

We test H5 to see whether new generation LLMs are at least as good as or better than the older generations. We do pairwise comparisons of models from the same providers and same size because each provider officially claims that their newer model outperforms its own previous generation.

With Meta, statistical analysis comparing absolute price deviation reveals that Llama 2 70B (M = 17.57, SD = 14.29) demonstrates significantly lower prediction errors than Llama 3 70B (M = 33.16, SD = 13.92), t(3586) = 33.11, p < 0.001, d = 1.11. This represents a 88.75% increase in prediction error. The newer Llama 4 17b Maverick (M = 34.04, SD = 12.94) shows further increase in pricing error, even compared to Llama 3 70B (t(3598) = 1.98, p = 0.024, d = 0.07), which shows an additional 2.67% increase in pricing error. Similarly, OpenAI's GPT 3.5 turbo (M = 30.91, SD = 20.79) shows significantly lower absolute price deviation compared to gpt-4o-mini (M = 40.88, SD = 201.17, t(3598) = 2.09, p = 0.018, d = 0.07) , and GPT 4.1 nano [3] (M = 48.41, SD = 239.81, t(3598) = 3.08, p = 0.002, d = 0.10), a 32.22% and 56.59% increase in prediction error respectively. Anthropic models show a similar degradation from the older generation to the latest one. Claude 3.5 Haiku (M = 25.15, SD = 14.93) shows significantly lower absolute price deviation compared to Claude 3.7 Sonnet (M = 32.43, SD = 14.45), t(2686) = 12.01, p < 0.001, d = 0.50, a 28.92% increase in prediction error. These findings support our alternative hypothesis that pricing accuracy decreases in the newer generation models (H5 supported; Figure 3).

A three-way mixed-effects ANOVA showed a significant three-way interaction among number of parameters, model generation, and anchoring group, $F(10, 14367) = 9.74, p < .001$ (Table 2). Follow-up OLS regression analyses showed that

[3]OpenAI claims that GPT 4.1 nano has similar intelligence to GPT 4o mini https://platform.openai.com/docs/models/compare?model=gpt-4.1

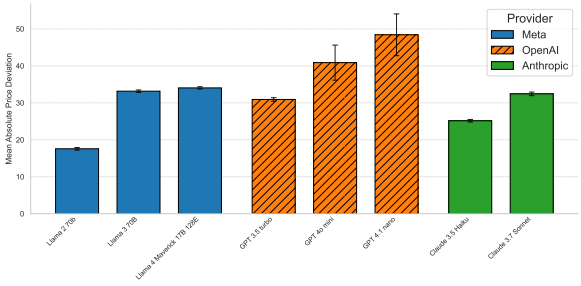


Figure 3: Mean absolute price deviation for similar sized models across generations

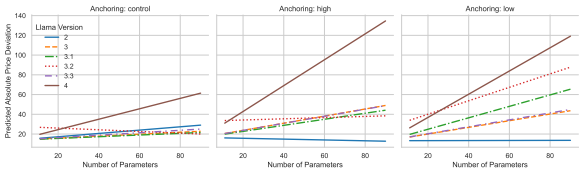


Figure 4: Model-predicted absolute price deviation by LLM generation and number of parameters, separately for each anchoring group. Each line traces the effect of increasing model scale within a generation, revealing a strong 3-way interaction.

all three-way interaction terms were statistically significant and positive (Table 10), indicating that newer and larger models are significantly more susceptible to anchoring manipulation. Simple slopes analyses further revealed that for newer model generations, increases in parameter count led to disproportionately higher (lower) price deviations in the high (low) anchoring group. Full predicted trajectories for each anchoring group and model generation are visualized in Figure 4.

Table 2: Results of the three-way ANOVA for absolute price deviation of LLMs by Meta

| Source | Sum of Squares (SS) | df | F | p |
|---|---|---|---|---|
| Intercept | 351232.85 | 1 | 30.91 | < .001 |
| Model Generation | 717195.55 | 5 | 12.62 | < .001 |
| Anchoring | 11169.62 | 2 | 0.49 | 0.612 |
| Model Generation:Anchoring | 532838.50 | 10 | 4.69 | < .001 |
| Number of Parameters | 62115.70 | 1 | 5.47 | 0.019 |
| Number of Parameters:Model Generation | 232023.41 | 5 | 4.08 | 0.001 |
| Number of Parameters:Anchoring | 53761.87 | 2 | 2.37 | 0.094 |
| Number of Parameters:Model Generation:Anchoring | 1106642.51 | 10 | 9.74 | < .001 |
| Residual | 163253684.92 | 14367 | | |

## 6.4 Discussion

We tested our Monte-Carlo framework in an experimental setting that allowed us to statistically test the presence of biases in LLM, in ways that traditional single-sample evaluations could not. Through this experimental design, we demonstrated how modern LLMs are susceptible to behavioral biases. Even though most LLMs predict prices that fall within reasonable range of a product category (Figure 6),

the predicted prices are significantly different from the actual list price of the product. Furthermore, cognitive biases further push LLM-predicted prices from actual prices, casting doubt on the reliability of LLMs in consumption scenarios. This finding is specially important in consumer behavior since consumers rely heavily on LLM predictions.

Perhaps the most striking finding of this article is the significant decrease in pricing accuracy observed in newer model generations across all three providers tested. Llama-3-70B showed an 88.75% increase in prediction error compared to Llama-2-70B, with a large effect size (d = 1.11). Similarly, both OpenAI's and Anthropic's newer models demonstrated significantly higher absolute price deviations than their predecessors. This pattern contradicts the general assumption that newer model generations necessarily improve across all capabilities.

We propose several potential explanations for this counterintuitive finding. First, newer model generations may prioritize alignment with human preferences over numerical accuracy in specific domains. As models are increasingly tuned to provide more nuanced responses, their ability to make common-sense predictions and avoid cognitive biases may be inadvertently compromised. This suggests a potential tradeoff between alignment and domain-specific reasoning. Second, training objectives may have shifted across generations to emphasize capabilities other than those needed by social sciences and business. Newer models might excel at coding, reasoning, or instruction following while sacrificing performance on quantitative estimation tasks that were not explicitly prioritized during training. Third, newer models might be more cautious in making price predictions, adding hedging language or broader confidence intervals that mathematically result in larger average deviations from ground truth. This "epistemic caution" could manifest as worse performance on point estimates while potentially representing a more accurate representation of prediction uncertainty.

Our analysis revealed complex interactions between model parameters, model generation (old vs. new), and anchoring manipulation. This suggests that the relationship between model size and generation show deterioration in their performance in the social sciences and business related tasks. While conventional wisdom suggests that larger models should perform better on most tasks, our findings indicate that for price predictions this relationship is non-monotonic. The significant three-way interaction of parameter count, model generation, and anchoring manipulation confirms that larger and newer models negatively influence pricing accuracy. This suggests that architectural improvements or training methodology changes between generations may have inadvertently reduced performance specifically on price estimation tasks, even as they improved performance on standard benchmarks.

### 6.5 Implications for research and practice

These findings have several important implications for research and deployment of LLMs. First, our results highlight the importance of domain-specific evaluation when deploying LLMs. Standard benchmarks may not capture performance on specialized tasks like price estimation. Organizations intending to use LLMs for pricing applications should conduct thorough evaluations rather than assuming newer models will perform better. Second, we identify a potential tension between general capabilities and specialized numerical reasoning in LLMs.

As models become more generally capable and aligned, they may sacrifice performance on specific quantitative and qualitative tasks. This suggests the need for specialized fine-tuning when deploying models for numerical prediction tasks and the use of Monte Carlo evaluation for assessing the true reliability the LLM for specialized tasks. Third, our finding that newer models demonstrate greater susceptibility to anchoring effects raises concerns about their deployment in real-world consumption scenarios (DeepMind, 2025) where such cognitive manipulations might be present. This suggests that system architects should explicitly evaluate and mitigate cognitive biases and incorporate safety measures for external manipulations (e.g., predatory advertisers putting anchoring text inside product descriptions to trick LLMs to purchase a certain product on the behalf of a user).

### 6.6 Conclusion

We introduce a Monte-Carlo evaluation framework for Large Language Models that accounts for inherent stochasticity of LLM predictions. Our framework enables statistical analysis of LLM responses to obtain measure of its reliability, an approach missing in standard LLM evaluation frameworks. We use our framework to study the susceptibility of LLMs to one of the most common human biases, anchoring effect. We replicate experiments based on behavioral sciences to test for cognitive biases.

Our experiment challenges the assumption that newer LLM generations necessarily improve across all capabilities, revealing significant regressions in pricing accuracy across multiple model providers. These findings underscore the importance of task-specific evaluation using the Monte Carlo framework, instead of current pass@1 evaluation framework. As LLMs continue to evolve, researchers and practitioners should remain attentive to these tradeoffs and develop strategies to improve model reliability in critical application domains.

## Limitations

This study has several limitations. We focused primarily on consumer products with reasonably standard pricing; future work should examine more complex pricing scenarios including B2B contexts and dynamic pricing environments. Additionally, our analysis does not fully explain why newer models perform worse on pricing tasks, pointing to the need for more detailed analysis of model internals and training procedures. Future research should investigate whether performance regression across generations is unique to pricing tasks or extends to other domains requiring precise numerical estimation.

## Acknowledgements

We have used AI-based tools for grammar and style assistance, improving writing clarity, and organizing initial brainstorming ideas during the manuscript preparation. All content have been verified by the authors prior to submission.

## References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rashmi Adaval and Robert S Wyer Jr. 2011. Conscious and nonconscious comparisons with price anchors: Effects on willingness to pay for related and unrelated products. *Journal of Marketing Research*, 48(2):355–365.

Gati V Aher, Rosa I Arriaga, and Adam Tauman Kalai. 2023. Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning*, pages 337–371. PMLR.

Gorkan Ahmetoglu, Adrian Furnham, and Patrick Fagan. 2014. Pricing practices: A critical review of their effects on consumer perceptions and behaviour. *Journal of Retailing and Consumer Services*, 21(5):696–707.

J Craig Andrews, Richard G Netemeyer, and Scot Burton. 1998. Consumer generalization of nutrient content claims in advertising. *Journal of marketing*, 62(4):62–75.

Dan Ariely, George Loewenstein, and Drazen Prelec. 2003. "coherent arbitrariness": Stable demand curves without stable preferences. *The Quarterly journal of economics*, 118(1):73–106.

Sachin Banker and Salil Khetani. 2019. Algorithm overdependence: How the use of algorithmic recommendation systems can increase risks to consumer well-being. *Journal of Public Policy & Marketing*, 38(4):500–515.

Manuela Benary, Xing David Wang, Max Schmidt, Dominik Soll, Georg Hilfenhaus, Mani Nassir, Christian Sigler, Maren Knödler, Ulrich Keller, Dieter Beule, and 1 others. 2023. Leveraging large language models for decision support in personalized oncology. *JAMA Network Open*, 6(11):e2343689–e2343689.

Amar Cheema and Vanessa M Patrick. 2008. Anytime versus only: Mind-sets moderate the effect of expansive versus restrictive frames on promotion evaluation. *Journal of Marketing Research*, 45(4):462–472.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Jan Clusmann, Fiona R Kolbinger, Hannah Sophie Muti, Zunamys I Carrero, Jan-Niklas Eckardt, Narmin Ghaffari Laleh, Chiara Maria Lavinia Löffler, Sophie-Caroline Schwarzkopf, Michaela Unger, Gregory P Veldhuizen, and 1 others. 2023. The future landscape of large language models in medicine. *Communications medicine*, 3(1):141.

Dena Cox and Anthony D Cox. 2001. Communicating the consequences of early detection: The role of evidence and framing. *Journal of Marketing*, 65(3):91–103.

DeepMind. 2025. Project mariner. https://deepmind.google/models/project-mariner/. Accessed: 2025-08-12.

Benedict GC Dellaert, Suzanne B Shu, Theo A Arentze, Tom Baker, Kristin Diehl, Bas Donkers, Nathanael J Fast, Gerald Häubl, Heidi Johnson, Uma R Karmarkar, and 1 others. 2020. Consumer decisions with artificially intelligent voice assistants. *Marketing Letters*, 31:335–347.

Kristin Diehl. 2005. When two rights make a wrong: Searching too much in ordered environments. *Journal of Marketing Research*, 42(3):313–322.

Bingyu Dong, Jie Bai, Tao Xu, and Yun Zhou. 2024. Large language models in education: A systematic review. *2024 6th International Conference on Computer Science and Technologies in Education (CSTE)*, pages 131–134.

Katharina Dowling, Daniel Guhl, Daniel Klapper, Martin Spann, Lucas Stich, and Narine Yegoryan. 2020. Behavioral biases in marketing. *Journal of the Academy of Marketing Science*, 48:449–477.

Jessica Maria Echterhoff, Yao Liu, Abeer Alessa, Julian McAuley, and Zexue He. 2024. Cognitive bias in decision-making with LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12640–12653, Miami, Florida, USA. Association for Computational Linguistics.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179.

Wensheng Gan, Zhenlian Qi, Jiayang Wu, and Chun-Wei Lin. 2023. Large language models in education: Vision and opportunities. *2023 IEEE International Conference on Big Data (BigData)*, pages 4776–4785.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Martie G Haselton, Daniel Nettle, and Paul W Andrews. 2015. The evolution of cognitive bias. *The handbook of evolutionary psychology*, pages 724–746.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Varsha Jain, Ketan Wadhwani, and Jacqueline K Eastman. 2024. Artificial intelligence consumer behavior: A hybrid review and research agenda. *Journal of consumer behaviour*, 23(2):676–697.

Gabbrielle M Johnson. 2021. Algorithmic bias: on the implicit biases of social technology. *Synthese*, 198(10):9941–9961.

Erik Jones and Jacob Steinhardt. 2022. Capturing failures of large language models via human cognitive biases. *Advances in Neural Information Processing Systems*, 35:11785–11799.

Daniel Kahneman. 2002. Maps of bounded rationality: A perspective on intuitive judgement and choice.

Anja Lambrecht and Bernd Skiera. 2006. Paying too much and being happy about it: Existence, causes, and consequences of tariff-choice biases. *Journal of marketing Research*, 43(2):212–223.

Leonard Lee, Michelle P Lee, Marco Bertini, Gal Zauberman, and Dan Ariely. 2015. Money, time, and the stability of consumer preferences. *Journal of Marketing Research*, 52(2):184–199.

Irwin P Levin and Gary J Gaeth. 1988. How consumers are affected by the framing of attribute information before and after consuming the product. *Journal of consumer research*, 15(3):374–378.

Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, and 1 others. 2022. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212.

David JC MacKay. 2003. *Information theory, inference and learning algorithms*. Cambridge university press.

Olivia Macmillan-Scott and Mirco Musolesi. 2024. (ir) rationality and cognitive biases in large language models. *Royal Society Open Science*, 11(6):240255.

Charles F Manski. 1977. The structure of random utility models. *Theory and decision*, 8(3):229.

AI Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI*, 2(5):6.

Vicki G Morwitz, Eric A Greenleaf, and Eric J Johnson. 1998. Divide and prosper: consumers' reactions to partitioned prices. *Journal of marketing research*, 35(4):453–463.

Zabir Al Nazi and Wei Peng. 2023. Large language models in healthcare and medical domain: A review. *Informatics*, 11:57.

Jillian Ross, Yoon Kim, and Andrew W Lo. 2024. Llm economicus? mapping the behavioral biases of llms via utility theory. *arXiv preprint arXiv:2408.02784*.

Shelle Santana, Manoj Thomas, and Vicki G Morwitz. 2020. The role of numbers in the customer journey. *Journal of Retailing*, 96(1):138–154.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, and 1 others. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*, pages 1–8.

Dilip Soman. 1998. The illusion of delayed incentives: evaluating future effort–money transactions. *Journal of Marketing Research*, 35(4):427–437.

Fritz Strack and Thomas Mussweiler. 1997. Explaining the enigmatic anchoring effect: Mechanisms of selective accessibility. *Journal of personality and social psychology*, 73(3):437.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. *Advances in Neural Information Processing Systems*, 35:21548–21561.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Amos Tversky and Daniel Kahneman. 1974. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, and Qingsong Wen. 2024. Large language models for education: A survey and outlook. *ArXiv*, abs/2403.18105.

Zeqiu Xu, Lingfeng Guo, Shuwen Zhou, Runze Song, and Kaiyi Niu. 2024. Enterprise supply chain risk management and decision support driven by large language models. *Academia Nexus Journal*, 3(2).

Yang Yang, Joachim Vosgerau, and George Loewenstein. 2013. Framing influences willingness to pay but not willingness to accept. *Journal of Marketing Research*, 50(6):725–738.

# A Appendix

## A.1 Experiment details

We formulated prompts to elicit WTP judgments while manipulating the presence and magnitude of price anchors. Figure 5 shows the experimental setup. The data collection for anthropic–claude-3-7-sonnet could not complete because the Anthropic server became overloaded after collecting 888 responses. Moreover, meta-llama2-70 has 1788 out of the 1800 possible responses because for the 12 missing responses it did not followed the response schema asking it for a floating point number. To ensure fairness and reflect real-world performance, we did not attempt to recollect or top up responses for these models, preserving the integrity of our original sampling procedure.

### A.1.1 Additional results

**H1** Per product analysis of WTP and actual list price provides more detailed insight on the effect of the product type on LLMs' pricing accuracy.

Table 4 shows the descriptive statistics of willingness-to-pay (WTP) and the list price for the products tested.

The result of one-sample t-tests for each product category is shown in Table 5. We see that for Docking station (t(4799) = 9.88, <0.001, M = 79.55, SD = 207.34, d = 0.14), Coffee pods (t(4799) = -29.68, <0.001, M = 43.69, SD = 31.78, d = -0.43), Weighted vest (t(4787) = -18.71, <0.001, M = 51.33, SD = 32.04, d = -0.27), Paperback book (t(4499) = -8.18, <0.001, M = 46.18, SD = 154.17, d = -0.12), Paper towels (t(4499) = -3.77, <0.001, M = 37.07, SD = 96.55, d = -0.06), and Women's shorts (t(4499) = 11.60, <0.001, M = 50.70, SD = 34.98, d = 0.17), LLMs have WTP that is significantly different from the actual list price.

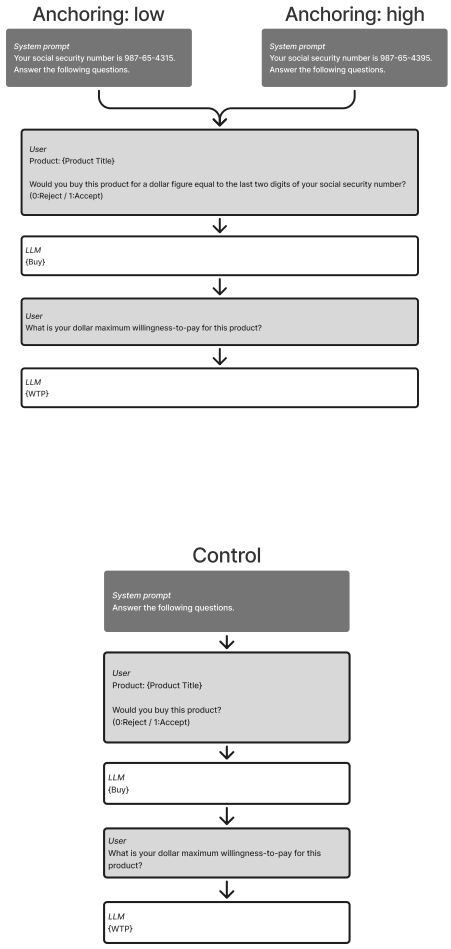


Figure 5: The structure of prompting for inducing anchoring effects and the control

**Three-way interaction effects** Table 10 shows the three-way interactions of LLM generation (i.e., Llama 2, Llama 3, Llama 3.1, Llama 3.2, Llama 3.3, and Llama 4), anchoring bias, and the number of parameters for LLMs produced by Meta.

Correlation of the Social Security Number (SSN) with the willingness-to-pay (WTP) for each LLM

Table 3: Product Categories and List Prices

| Category | List Price (USD) |
|---|---|
| Computer Accessories | 49.99 |
| Grocery & Gourmet Food | 57.31 |
| Sports & Outdoors | 59.99 |
| Books | 64.99 |
| Health & Household | 42.49 |
| Clothing, Shoes & Jewelry | 44.65 |

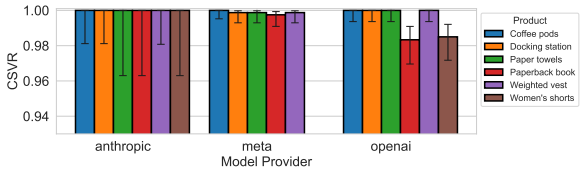


Figure 6: Common Sense Validity Rate of price predictions (Error bars show Wilson score interval)

and human results (Ariely et al., 2003) are shown in Table 11.

### A.2 Statistical framework

In this section, we develop the statistical framework for our Monte-Carlo evaluation framework.

#### A.2.1 Theorems and proofs

#### A.2.2 Unbiasedness

Let $f(p, y, r)$ be an evaluation function mapping the prompt $p$, ground-truth value $y$, and a sampled response $r \sim \mathcal{M}(p; T)$ from a language model $\mathcal{M}$ at temperature $T$, to a real number. Define the true expected value of the function as

$$v(\mathcal{M}; p, y) := \mathbb{E}_{r \sim \mathcal{M}(p;T)}[f(p, y, r)]. \quad (7)$$

We estimate this quantity empirically using the $K$-sample Monte Carlo estimator:

$$\hat{v}_K(\mathcal{M}; p, y) := \frac{1}{K} \sum_{k=1}^{K} f(p, y, r_k), \quad (8)$$

where each $r_k \sim \mathcal{M}(p; T)$ is sampled independently.

**Theorem 1**

$$\mathbb{E}[\hat{v}_K(\mathcal{M}; p, y)] = v(\mathcal{M}; p, y) \quad (9)$$

**Proof** First, the i.i.d. (independent and identically distributed) assumption is strongly valid for LLMs in the context of Monte-Carlo evaluation. This is because repeated sampling of LLM responses to the same prompt under a fixed temperature setting does not influence subsequent samples. Specifically, from sample $k$ to sample $k + 1$, there is no dependency or effect of sample $k$ on sample $k + 1$. Each response is generated independently based on the stochastic nature of the model at temperature $T$, and the underlying probability distribution over possible outputs remains unchanged across samples (assuming the researcher does not fine-tune the LLM in between each sample). Therefore, the responses can be considered i.i.d., satisfying a key requirement for the unbiasedness and convergence properties of the Monte-Carlo estimator.

Next, by definition of the Monte Carlo estimator,

$$\hat{v}_K = \frac{1}{K} \sum_{k=1}^{K} f(p, y, r_k). \quad (10)$$

Taking the expectation of both sides, and applying the linearity of expectation:

$$\mathbb{E}[\hat{v}_K] = \mathbb{E}\left[\frac{1}{K} \sum_{k=1}^{K} f(p, y, r_k)\right] \quad (11)$$

$$= \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}\left[f(p, y, r_k)\right]. \quad (12)$$

Each $r_k$ is drawn independently and identically from the generative distribution of $\mathcal{M}(p; T)$, and therefore:

$$\mathbb{E}\left[f(p, y, r_k)\right] = \mathbb{E}_{r \sim \mathcal{M}(p;T)}\left[f(p, y, r)\right] \quad (13)$$

$$= v(\mathcal{M}; p, y), \quad \forall k. \quad (14)$$

Hence,

$$\mathbb{E}[\hat{v}_K] = \frac{1}{K} \cdot K \cdot v(\mathcal{M}; p, y) \quad (15)$$

$$= v(\mathcal{M}; p, y). \quad (16)$$

Therefore, $\hat{v}_K$ is an unbiased estimator of $v(\mathcal{M}; p, y)$:

$$\mathbb{E}[\hat{v}_K] = v(\mathcal{M}; p, y). \quad \blacksquare \quad (17)$$

#### A.2.3 Variance decay

Variance decay property ensures as the number of Monte Carlo responses $K$ increases, the variance of the estimated metric (e.g., MAPD or CSVR) decreases at a rate of $\frac{1}{K}$. Statistically, our estimator converges more reliably to the true value of

Table 4: Descriptive statistics for willingness-to-pay (WTP) per product

| Product | n | M | SD | List Price (USD) |
|---|---|---|---|---|
| Coffee pods | 4799 | 43.69 | 31.78 | 57.31 |
| Docking station | 4798 | 79.55 | 207.34 | 49.99 |
| Paper towels | 4496 | 37.07 | 96.55 | 42.49 |
| Paperback book | 4498 | 46.18 | 154.17 | 64.99 |
| Weighted vest | 4787 | 51.33 | 32.04 | 59.99 |
| Women's shorts | 4498 | 50.70 | 34.98 | 44.65 |

Table 5: Results of the one-sample t-test

| Product | t | p | d |
|---|---|---|---|
| Docking station | 9.88 | <0.001 | 0.14 |
| Coffee pods | -29.68 | <0.001 | -0.43 |
| Weighted vest | -18.71 | <0.001 | -0.27 |
| Paperback book | -8.18 | <0.001 | -0.12 |
| Paper towels | -3.77 | <0.001 | -0.06 |
| Women's shorts | 11.60 | <0.001 | 0.17 |

the model's expected behavior as we average over more samples, thereby enhancing reliability. This also quantifies the confidence we have in our evaluations. With more samples, we reduce variability arising from the model's stochastic nature.

In practice, if we desire a specific tolerance for our confidence interval (e.g., ±2% MAPD), we can reverse-engineer to find an appropriate $K$ using

$$\text{Standard error} = \sqrt{\text{Var}[\hat{v}_K]} = \frac{\sigma}{\sqrt{K}}. \tag{18}$$

Additionally, when comparing two models statistically, the variance decay can inform how many samples are needed to achieve sufficient power for t-tests or bootstrap comparisons.

**Theorem 2** Let $r_1, \ldots, r_K \overset{\text{i.i.d.}}{\sim} \mathcal{M}(p; T)$ be i.i.d. samples from the stochastic LLM at temperature $T$. Given the Monte-Carlo sample evaluation, $\hat{v}_K$, the Monte-Carlo sample variance asymptotes to zero as $K \to \infty$,

$$\lim_{K\to\infty} \text{Var}[\hat{v}_K] = 0, \tag{19}$$

where $\sigma^2$ is the variance of the random variable $f(p, y, r_k)$.

**Proof** We aim to compute the variance of $\hat{v}_K$, i.e., the variance of the sample mean of $K$ i.i.d. random variables. By the definition of variance, we have:

$$\text{Var}[\hat{v}_K] = \text{Var}\left[\frac{1}{K}\sum_{k=1}^{K} f(p, y, r_k)\right]. \tag{20}$$

Since the $f(p, y, r_k)$ are i.i.d. random variables with finite variance $\sigma^2 = \text{Var}[f(p, y, r_k)]$, we can use the properties of variance. For any constants $a_k \in \mathbb{R}$ and independent random variables $X_k$

$$\text{Var}\left[\sum_{k=1}^{K} a_k X_k\right] = \sum_{k=1}^{K} a_k^2 \,\text{Var}[X_k]. \tag{21}$$

In our case, $a_k = \frac{1}{K}$ and $\text{Var}[X_k] = \sigma^2$, $\forall k$. Applying this rule gives:

$$\text{Var}\left[\frac{1}{K}\sum_{k=1}^{K} f(p, y, r_k)\right] = \frac{1}{K^2}\sum_{k=1}^{K} \text{Var}[f(p, y, r_k)] \tag{22}$$

$$= \frac{1}{K^2} \cdot K \cdot \sigma^2 \tag{23}$$

$$= \frac{\sigma^2}{K}. \tag{24}$$

Hence, the variance of the estimated metric is given by

$$\text{Var}[\hat{v}_K] = \frac{\sigma^2}{K}. \tag{25}$$

Taking the limit,

$$\lim_{K\to\infty} \text{Var}[\hat{v}_K] = \lim_{K\to\infty} \frac{\sigma^2}{K} = 0. \tag{26}$$

This concludes the proof.

■

**Asymptotic Normality of Monte Carlo Estimates** We now formally establish that the Monte Carlo evaluation estimate of an LLM's expected behavior over stochastic samples follows an asymptotically normal distribution as the number of samples

Table 6: Descriptive statistics of Absolute price deviations for the LLMs tested

| Model Name | n | M | SD |
|---|---|---|---|
| Claude 3.5 Haiku | 1800 | 25.15 | 14.93 |
| Claude 3.7 Sonnet | 888 | 32.43 | 14.45 |
| GPT 3.5 turbo | 1800 | 30.91 | 20.79 |
| GPT 4.1 | 1800 | 29.05 | 15.26 |
| GPT 4.1 mini | 1800 | 34.81 | 14.74 |
| GPT 4.1 nano | 1800 | 48.41 | 239.81 |
| GPT 4o | 1800 | 27.26 | 20.50 |
| GPT 4o mini | 1800 | 40.88 | 201.17 |
| Llama 2 70b | 1788 | 17.57 | 14.29 |
| Llama 3 70b | 1800 | 33.16 | 13.92 |
| Llama 3.1 70b | 1800 | 37.23 | 143.71 |
| Llama 3.2 11b | 1800 | 31.46 | 15.01 |
| Llama 3.2 90b | 1800 | 48.75 | 264.83 |
| Llama 3.3 70b | 1800 | 34.01 | 14.24 |
| Llama 4 17b Maverick | 1800 | 34.04 | 12.94 |
| Llama 4 17b Scout | 1800 | 29.34 | 15.82 |

Table 7: Results of the ANOVA for Absolute Price Deviation of different LLMs

| Source | Sum of Squares (SS) | df | F | p |
|---|---|---|---|---|
| Intercept | 1138785.47 | 1 | 92.03 | < .001 |
| LLM | 1682960.21 | 15 | 9.07 | < .001 |
| Residual | 344749087.15 | 27860 | | |

grows. This justifies the use of confidence intervals and statistical comparisons across models.

Let $f(p, y, r)$ be a real-valued function representing the evaluation score for an LLM response $r \sim \mathcal{M}(p; T)$ to prompt $p$ with ground truth $y$. Assume that responses $\{r_k\}_{k=1}^{K}$ are independent and identically distributed (i.i.d.) from the stochastic language model's conditional distribution $\mathcal{M}(p; T)$. Define the sample mean score (Monte Carlo estimate) as

$$\hat{v}_K(\mathcal{M}; p, y) := \frac{1}{K} \sum_{k=1}^{K} f(p, y, r_k). \tag{27}$$

**Corollary** The normalized estimator follows a normal distribution as $K \to \infty$:

$$[\hat{v}_K(\mathcal{M}; p, y) - v(\mathcal{M}; p, y)] \sqrt{K} \xrightarrow{d} \mathcal{N}(0, \sigma^2), \tag{28}$$

where

$$v(\mathcal{M}; p, y) := \mathbb{E}_{r \sim \mathcal{M}(p;T)}[f(p, y, r)]. \tag{29}$$

$f(p, y, r_k)$ is an independent draw induced by the model sampling $r_k \sim \mathcal{M}(p; T)$ that satisfies true i.i.d. conditions, since the LLM does not have any state changes between each sample.

Then, we define the sample mean:

$$\hat{v}_K(\mathcal{M}; p, y) := \frac{1}{K} \sum_{k=1}^{K} f(p, y, r_k), \tag{30}$$

and the population mean:

$$\begin{aligned} \mu :&= \mathbb{E}[f(p, y, r_k)] && (31)\\ &= \mathbb{E}_{r \sim \mathcal{M}(p;T)}[f(p, y, r)] && (32)\\ &= v(\mathcal{M}; p, y), && (33) \end{aligned}$$

For the population variance, $\sigma^2$, we can safely assume that $\sigma^2 < \infty$ (finite variance), because of the conditions of the evaluation metric (e.g., bounded or sub-Gaussian scoring functions such as classification accuracy or standardized absolute error).

Based on the Central Limit Theorem, as $k \to \infty$, the normalized sample mean satisfies:

$$\frac{\sqrt{K}(\hat{v}_K(\mathcal{M}; p, y) - \mu)}{\sigma} \xrightarrow{d} \mathcal{N}(0, 1) \tag{34}$$

Now, multiply both sides of the normalized variable formula by $\sigma$, which gives:

Table 8: Results of the ANOVA for the effect of anchoring on Absolute Price Deviation

| Source | Sum of Squares (SS) | df | F | p |
|---|---|---|---|---|
| Intercept | 19944203.42 | 1 | 1524.62 | < .001 |
| Anchoring | 10793641.35 | 2 | 412.56 | < .001 |
| Residual | 364617818.90 | 27873 | | |

Table 9: Results of the two-way ANOVA for the effect of anchoring manipulation and LLM tested Absolute Price Deviation

| Source | Sum of Squares (SS) | df | F | p |
|---|---|---|---|---|
| Intercept | 1647304.04 | 1 | 127.99 | < .001 |
| Anchoring | 296106.68 | 2 | 11.50 | < .001 |
| LLM | 482523.71 | 15 | 2.50 | 0.001 |
| Anchoring×LLM | 4924800.20 | 30 | 12.75 | < .001 |
| Residual | 358170503.64 | 27828 | | |

$$\sqrt{K}(\hat{v}_K(\mathcal{M};p,y)-\mu) \xrightarrow{d} \mathcal{N}(0,\sigma^2) \tag{35}$$

Substituting $\mu$ with $v(\mathcal{M};p,y)$:

$$\sqrt{K}(\hat{v}_K(\mathcal{M};p,y)-v(\mathcal{M};p,y)) \xrightarrow{d} \mathcal{N}(0,\sigma^2) \quad \blacksquare \tag{36}$$

In the Monte-Carlo LLM evaluation framework, we are estimating the expected score of the model over stochastic outputs via:

$$\hat{v}_K = \frac{1}{K}\sum_{k=1}^{K} f(p,y,r_k) \tag{37}$$

where $\mu = E[f(p,y,r)]$ is the "true" behavioral profile of the model under its stochastic decoding.

### A.2.4 Power Analysis for Determining Optimal K

The power analysis for selecting the minimal $K$ in our Monte-Carlo framework, focuses on ensuring sufficient power to detect small effect sizes (i.e., Cohen's $f \leq 0.10$) in comparisons of LLM behaviors via ANOVA. We provide the derivations, statistical setup, approximations, sensitivity analyses, and implementation notes.

**ANOVA Setup** We frame the problem as a one-way ANOVA to test for differences in expected scores $v(\mathcal{M};p,y)$ across $g$ groups (e.g., different LLMs, prompts, or behavioral categories). Each group has $K$ i.i.d. Monte-Carlo samples, yielding a total sample size $N = gK$. The null hypothesis is that all group means are equal ($H_0 : \mu_1 = \mu_2 = \cdots = \mu_g$), while the alternative assumes differences quantified by the effect size.

We use Cohen's $f$ as the effect size measure,

$$f = \sqrt{\frac{\sum_{i=1}^{g}(\mu_i-\bar{\mu})^2/g}{\sigma^2}},$$

where $\bar{\mu} = \sum_{i=1}^{g}\mu_i/g$ is the grand mean, and $\sigma^2$ is the common within-group variance (assumed equal across groups, as per homoscedasticity in ANOVA).

The test statistic is the F-ratio, which under $H_0$ follows a central F-distribution with degrees of freedom $\mathrm{df}_1 = g-1$ (between groups) and $\mathrm{df}_2 = g(K-1)$ (within groups). Under the alternative, it follows a non-central F-distribution with non-centrality parameter

$$\lambda = \frac{\sum_{i=1}^{g} K(\mu_i-\bar{\mu})^2}{\sigma^2} = gKf^2,$$

since for balanced groups, $\sum_{i=1}^{g}(\mu_i-\bar{\mu})^2/g = f^2\sigma^2$.

**Derivation of Minimal K** The power $1-\beta$ is the probability that the F-statistic exceeds the critical value $F_{\text{crit}} = F^{-1}(1-\alpha;\mathrm{df}_1,\mathrm{df}_2)$ under the non-central F-distribution, where $\alpha$ is the significance level (e.g., 0.05) and $\beta$ is the Type II error rate:

$$1-\beta = 1-F(F_{\text{crit}};\mathrm{df}_1,\mathrm{df}_2,\lambda),$$

with $F(\cdot;\mathrm{df}_1,\mathrm{df}_2,\lambda)$ denoting the cumulative distribution function (CDF) of the non-central F-distribution.

To find the minimal $K$ for a desired power $1-\beta$ (e.g., 0.80), effect size $f$, $\alpha$, and $g$, we solve for $K$ such that

$$F(F_{\text{crit}};\mathrm{df}_1,\mathrm{df}_2,\lambda) \leq \beta,$$

where $\lambda = gKf^2$, $\mathrm{df}_1 = g-1$, and $\mathrm{df}_2 = g(K-1)$. Note that $\mathrm{df}_2$ and $F_{\text{crit}}$ depend on $K$, making this an implicit equation. The minimal integer $K$ is found by a binary search provided in Algorithm 1.

For large $K$, approximations can be used. For instance, the non-central F can be approximated by a normal distribution for large $\mathrm{df}_2$, but exact computation via numerical integration (e.g., using the 'pf'

Table 10: OLS regression results for Meta's LLM generation, anchoring bias, and the number of model parameters

| | B | SE | t | p | CI Lower | CI Upper |
|---|---|---|---|---|---|---|
| Intercept | 13.870 | 2.495 | 5.56 | < .001 | 8.980 | 18.760 |
| Model Generation[3] | -0.001 | 0.001 | -0.78 | 0.434 | -0.003 | 0.001 |
| Model Generation[3.1] | -0.001 | 0.001 | -0.94 | 0.347 | -0.004 | 0.001 |
| Model Generation[3.2] | 13.848 | 2.500 | 5.54 | < .001 | 8.948 | 18.748 |
| Model Generation[3.3] | -0.001 | 0.001 | -0.50 | 0.617 | -0.003 | 0.002 |
| Model Generation[4] | 0.021 | 0.013 | 1.63 | 0.103 | -0.004 | 0.047 |
| Anchoring[high] | 2.663 | 3.528 | 0.76 | 0.450 | -4.252 | 9.579 |
| Anchoring[low] | -0.632 | 3.528 | -0.18 | 0.858 | -7.548 | 6.283 |
| Model Generation[3]×Anchoring[high] | 0.007 | 0.002 | 3.79 | < .001 | 0.003 | 0.010 |
| Model Generation[3.1]×Anchoring[high] | 0.006 | 0.002 | 3.47 | 0.001 | 0.003 | 0.010 |
| Model Generation[3.2]×Anchoring[high] | 2.627 | 3.535 | 0.74 | 0.457 | -4.303 | 9.557 |
| Model Generation[3.3]×Anchoring[high] | 0.006 | 0.002 | 3.57 | < .001 | 0.003 | 0.010 |
| Model Generation[4]×Anchoring[high] | 0.058 | 0.018 | 3.19 | 0.001 | 0.022 | 0.094 |
| Model Generation[3]×Anchoring[low] | 0.006 | 0.002 | 3.23 | 0.001 | 0.002 | 0.009 |
| Model Generation[3.1]×Anchoring[low] | 0.009 | 0.002 | 5.29 | < .001 | 0.006 | 0.013 |
| Model Generation[3.2]×Anchoring[low] | -0.653 | 3.535 | -0.18 | 0.854 | -7.583 | 6.277 |
| Model Generation[3.3]×Anchoring[low] | 0.006 | 0.002 | 3.15 | 0.002 | 0.002 | 0.009 |
| Model Generation[4]×Anchoring[low] | 0.048 | 0.018 | 2.60 | 0.009 | 0.012 | 0.084 |
| Number of Parameters | 0.168 | 0.072 | 2.34 | 0.019 | 0.027 | 0.309 |
| Number of Parameters×Model Generation[3] | -0.069 | 0.088 | -0.78 | 0.434 | -0.242 | 0.104 |
| Number of Parameters×Model Generation[3.1] | -0.083 | 0.088 | -0.94 | 0.347 | -0.256 | 0.090 |
| Number of Parameters×Model Generation[3.2] | -0.253 | 0.083 | -3.06 | 0.002 | -0.416 | -0.091 |
| Number of Parameters×Model Generation[3.3] | -0.044 | 0.088 | -0.50 | 0.617 | -0.217 | 0.129 |
| Number of Parameters×Model Generation[4] | 0.360 | 0.220 | 1.63 | 0.103 | -0.072 | 0.792 |
| Number of Parameters×Anchoring[high] | -0.210 | 0.102 | -2.07 | 0.038 | -0.410 | -0.011 |
| Number of Parameters×Anchoring[low] | -0.164 | 0.102 | -1.62 | 0.106 | -0.363 | 0.035 |
| Number of Parameters×Model Generation[3]×Anchoring[high] | 0.471 | 0.125 | 3.79 | < .001 | 0.227 | 0.716 |
| Number of Parameters×Model Generation[3.1]×Anchoring[high] | 0.432 | 0.125 | 3.47 | 0.001 | 0.188 | 0.676 |
| Number of Parameters×Model Generation[3.2]×Anchoring[high] | 0.356 | 0.117 | 3.04 | 0.002 | 0.126 | 0.586 |
| Number of Parameters×Model Generation[3.3]×Anchoring[high] | 0.444 | 0.125 | 3.57 | < .001 | 0.200 | 0.689 |
| Number of Parameters×Model Generation[4]×Anchoring[high] | 0.993 | 0.312 | 3.19 | 0.001 | 0.382 | 1.605 |
| Number of Parameters×Model Generation[3]×Anchoring[low] | 0.403 | 0.125 | 3.23 | 0.001 | 0.159 | 0.647 |
| Number of Parameters×Model Generation[3.1]×Anchoring[low] | 0.659 | 0.125 | 5.29 | < .001 | 0.415 | 0.903 |
| Number of Parameters×Model Generation[3.2]×Anchoring[low] | 0.931 | 0.117 | 7.95 | < .001 | 0.701 | 1.160 |
| Number of Parameters×Model Generation[3.3]×Anchoring[low] | 0.392 | 0.125 | 3.15 | 0.002 | 0.148 | 0.636 |
| Number of Parameters×Model Generation[4]×Anchoring[low] | 0.812 | 0.312 | 2.60 | 0.009 | 0.201 | 1.423 |

function in R or SciPy in Python) is recommended for precision.

### A.3 Algorithms

Table 11: Correlation between WTP and SSN by Product and Model Name vs. Human Average for the Effect of Anchoring

| | Coffee pods | Docking station | Paper towels | Paperback book | Weighted vest | Women's shorts | Average Correlation |
|---|---|---|---|---|---|---|---|
| Claude 3.5 Haiku | -0.306 | 0.946 | 0.413 | 0.776 | 0.937 | 0.084 | 0.475 |
| Claude 3.7 Sonnet | 0.961 | 0.994 | | | 0.996 | | 0.984 |
| GPT 3.5 turbo | 0.913 | 0.302 | 0.958 | 1.000 | 0.846 | 0.765 | 0.797 |
| GPT 4.1 | 0.749 | 0.979 | 0.682 | 0.979 | 0.641 | 0.920 | 0.825 |
| GPT 4.1 mini | 0.980 | 1.000 | 1.000 | 1.000 | 0.988 | 1.000 | 0.995 |
| GPT 4.1 nano | 0.041 | -0.016 | -0.171 | -0.078 | -0.019 | -0.074 | -0.053 |
| GPT 4o | 0.515 | 0.051 | 0.606 | 0.814 | 0.478 | 0.670 | 0.523 |
| GPT 4o mini | 0.752 | -0.039 | 0.058 | 0.992 | 0.821 | 0.983 | 0.594 |
| Llama 2 70b | 0.020 | -0.072 | 0.040 | 0.091 | 0.063 | -0.041 | 0.017 |
| Llama 3 70b | 0.995 | 0.976 | 1.000 | 1.000 | 0.996 | 0.996 | 0.994 |
| Llama 3.1 70b | 0.831 | -0.059 | 0.799 | 1.000 | 0.933 | 0.892 | 0.733 |
| Llama 3.2 11b | 0.723 | 0.932 | 0.572 | 0.791 | 0.827 | 0.727 | 0.762 |
| Llama 3.2 90b | 0.818 | -0.166 | 0.874 | 0.986 | 0.590 | 0.941 | 0.674 |
| Llama 3.3 70b | 0.983 | 0.988 | 1.000 | 1.000 | 1.000 | 0.993 | 0.994 |
| Llama 4 17b Maverick | 1.000 | 1.000 | 0.603 | 1.000 | 1.000 | 1.000 | 0.934 |
| Llama 4 17b Scout | 0.497 | 1.000 | 0.937 | 1.000 | 1.000 | 0.974 | 0.901 |
| Human (Ariely et al. 2003) | | | | | | | 0.388 |

**Algorithm 1:** Binary Search for Minimal $K$

**Input:** Number of groups $g$, effect size $f$, significance level $\alpha$, target power $1 - \beta$
**Output:** Minimal integer $K$ such that power $\geq 1 - \beta$

low $\leftarrow 1$;
high $\leftarrow$ some large integer (e.g., $10^6$);
**while** *low* $<$ *high* **do**
  mid $\leftarrow \lfloor (\text{low} + \text{high})/2 \rfloor$;
  $\text{df}_1 \leftarrow g - 1$;
  $\text{df}_2 \leftarrow g(\text{mid} - 1)$;
  $F_{\text{crit}} \leftarrow F^{-1}(1 - \alpha; \text{df}_1, \text{df}_2)$;
  $\lambda \leftarrow g \cdot \text{mid} \cdot f^2$;
  power $\leftarrow 1 - F(F_{\text{crit}}; \text{df}_1, \text{df}_2, \lambda)$;
  **if** *power* $\geq 1 - \beta$ **then**
    high $\leftarrow$ mid;
  **else**
    low $\leftarrow$ mid + 1;
  **end**
**end**
**return** low;

**Algorithm 2:** Monte Carlo LLM Evaluation for Pricing Tasks

**Input:** $\mathcal{M}$: Set of LLMs to evaluate
$\mathcal{D} = \{(p_1, y_1), (p_2, y_2), \ldots, (p_N, y_N)\}$: Evaluation dataset with prompts $p_i$ and ground truth prices $y_i$
$K$: Number of Monte Carlo samples per prompt
$T$: Temperature parameter for sampling
$\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_N\}$: Valid price ranges for each prompt
**Output:** CSVR scores and confidence intervals for each model
MAPD scores and confidence intervals for each model
Statistical comparisons between models

**Function** `EvaluateModels`$(\mathcal{M}, \mathcal{D}, K, T, \mathcal{C})$**:**
  **foreach** *model* $m \in \mathcal{M}$ **do**
    $\text{CSVR}_m \leftarrow$ `CalculateCSVR`$(m, \mathcal{D}, K, T, \mathcal{C})$ Algorithm 3;
    $\text{MAPD}_m \leftarrow$ `CalculateMAPD`$(m, \mathcal{D}, K, T)$ Algorithm 4;
  **end**
  `PerformStatisticalAnalysis`$(\mathcal{M},$ *CSVR, MAPD*$)$ Algorithm 5;
  **return** Results;

**Algorithm 3:** Calculate CSVR

**Input:** $\mathcal{M}$ - Set of LLMs to evaluate
$\mathcal{D} = \{(p_1, y_1), (p_2, y_2), \ldots, (p_N, y_N)\}$ - Evaluation dataset with prompts $p_i$ and ground truth prices $y_i$
$K$ - Number of Monte Carlo samples per prompt
$T$ - Temperature parameter for sampling
$\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_N\}$ - Valid price ranges for each prompt
**Output:** CSVR scores and confidence intervals for each model

**Function** `CalculateCSVR`$(m, \mathcal{D}, K, T, \mathcal{C})$**:**

- $\hat{p} \leftarrow \emptyset$ ; `// Initialize empty array for prompt-level CSVR`
- **for** $i \leftarrow 1$ **to** $N$ **do**
  - valid_count $\leftarrow 0$;
  - **for** $j \leftarrow 1$ **to** $K$ **do**
    - $R_{i,j} \leftarrow$ `GenerateResponse`$(m, p_i, T)$ ; `// Generate response`
    - $\hat{Y}_{i,j} \leftarrow$ `ExtractPrice`$(R_{i,j})$ ; `// Extract price`
    - **if** $\hat{Y}_{i,j} \in \mathcal{C}_i$ **then**
      - valid_count $\leftarrow$ valid_count $+ 1$;
    - **end**
  - **end**
  - $\hat{p}_i \leftarrow$ valid_count$/K$ ; `// Estimate CSVR for prompt` $i$
  - Add $\hat{p}_i$ to $\hat{p}$;
- **end**
- CSVR $\leftarrow \frac{1}{N}\sum_{i=1}^{N} \hat{p}_i$ ; `// Calculate overall CSVR`
- $\text{CI}_{\text{CSVR}} \leftarrow \text{CSVR} \pm 1.96 \times \sqrt{\frac{\text{CSVR} \times (1-\text{CSVR})}{N}}$;
- **return** CSVR, $\text{CI}_{\text{CSVR}}, \hat{p}$;

**Algorithm 4:** Calculate MAPD

**Input:** $\mathcal{M}$ - Set of LLMs to evaluate
$\mathcal{D} = \{(p_1, y_1), (p_2, y_2), \ldots, (p_N, y_N)\}$ - Evaluation dataset with prompts $p_i$ and ground truth prices $y_i$
$K$ - Number of Monte Carlo samples per prompt
$T$ - Temperature parameter for sampling
**Output:** MAPD scores and confidence intervals for each model

**Function** `CalculateMAPD`$(m, \mathcal{D}, K, T)$**:**

- $\hat{\mu} \leftarrow \emptyset$ ; `// Initialize empty array`
- **for** $i \leftarrow 1$ **to** $N$ **do**
  - $\text{APD}_i \leftarrow \emptyset$ ; `// Store APD values`
  - **for** $j \leftarrow 1$ **to** $K$ **do**
    - $R_{i,j} \leftarrow$ `GenerateResponse`$(m, p_i, T)$;
    - $\hat{Y}_{i,j} \leftarrow$ `ExtractPrice`$(R_{i,j})$;
    - $\text{APD}_{i,j} \leftarrow |\hat{Y}_{i,j} - y_i|$;
    - Add $\text{APD}_{i,j}$ to $\text{APD}_i$;
  - **end**
  - $\hat{\mu}_i \leftarrow \frac{1}{K}\sum_{j=1}^{K} \text{APD}_{i,j}$;
  - $\hat{\sigma}_i^2 \leftarrow \frac{1}{K-1}\sum_{j=1}^{K}(\text{APD}_{i,j} - \hat{\mu}_i)^2$;
  - Add $\hat{\mu}_i$ to $\hat{\mu}$;
- **end**
- $\text{MAPD} \leftarrow \frac{1}{N}\sum_{i=1}^{N} \hat{\mu}_i$;
- $s^2 \leftarrow \frac{1}{N-1}\sum_{i=1}^{N}(\hat{\mu}_i - \text{MAPD})^2$;
- $\text{CI}_{\text{MAPD}} \leftarrow \text{MAPD} \pm t_{N-1,0.975} \times \frac{\sqrt{s^2}}{\sqrt{N}}$;
- **return** MAPD, $\text{CI}_{\text{MAPD}}, \hat{\mu}$;

**Algorithm 5:** Perform Statistical Analysis

**Input:** $\mathcal{M}$ - Set of LLMs to evaluate

**Output:** Statistical comparisons between models

**Function** `PerformStatisticalAnalysis(`$\mathcal{M}$, *CSVR, MAPD*`)`**:**

  **foreach** *pair of models* $(m_A, m_B) \in \mathcal{M} \times \mathcal{M}$ *where* $m_A \neq m_B$ **do**

    $t \leftarrow \frac{\text{MAPD}_{m_A} - \text{MAPD}_{m_B}}{\sqrt{\frac{s_{m_A}^2}{N} + \frac{s_{m_B}^2}{N}}}$;

    $df \leftarrow \frac{(\frac{s_{m_A}^2}{N} + \frac{s_{m_B}^2}{N})^2}{\frac{(s_{m_A}^2/N)^2}{N-1} + \frac{(s_{m_B}^2/N)^2}{N-1}}$;

    $p\_value \leftarrow$ `ComputePValue(`$t, df$`)`;

    $s_{\text{pooled}} \leftarrow \sqrt{\frac{(N-1)s_{m_A}^2 + (N-1)s_{m_B}^2}{2N-2}}$;

    $d \leftarrow \frac{\text{MAPD}_{m_A} - \text{MAPD}_{m_B}}{s_{\text{pooled}}}$;

    pct_increase $\leftarrow \frac{\text{MAPD}_{m_B} - \text{MAPD}_{m_A}}{\text{MAPD}_{m_A}} \times 100\%$;

    Store and report comparison results;

  **end**

  `PerformANOVA(`$\mathcal{M}$, *MAPD*`)`;