

# When to Ask a Question: Understanding Communication Strategies in Generative AI Tools

Charlotte Park<sup>1</sup> ✉ 

Massachusetts Institute of Technology, Cambridge, MA, USA

Kate Donahue ✉ 

Massachusetts Institute of Technology, Cambridge, MA, USA

Manish Raghavan ✉ 

Massachusetts Institute of Technology, Cambridge, MA, USA

---

## Abstract

Generative AI models differ from traditional machine learning tools in that they allow users to provide as much or as little information as they choose in their inputs. This flexibility often leads users to omit certain details, relying on the models to infer and fill in under-specified information based on distributional knowledge of user preferences. Such inferences may privilege majority viewpoints and disadvantage users with atypical preferences, raising concerns about fairness. Unlike more traditional recommender systems, LLMs can explicitly solicit more information from users through natural language. However, while directly eliciting user preferences could increase personalization and mitigate inequality, excessive querying places a burden on users who value efficiency. We develop a stylized model of user-LLM interaction and develop an objective that captures tradeoff between user burden and preference representation. Building on the observation that individual preferences are often correlated, we analyze how AI systems should balance inference and elicitation, characterizing the optimal amount of information to solicit before content generation. Ultimately, we show that information elicitation can mitigate the systematic biases of preference inference, enabling the design of generative tools that better incorporate diverse user perspectives while maintaining efficiency. We complement this theoretical analysis with an empirical evaluation illustrating the model’s predictions and exploring their practical implications.

**2012 ACM Subject Classification** Human-centered computing → User models; Human-centered computing → Interaction paradigms; Computing methodologies → Artificial intelligence

**Keywords and phrases** human-AI interaction, user modeling, personalization

**Digital Object Identifier** 10.4230/LIPIcs.FORC.2026.7

## 1 Introduction

Generative AI models present a new frontier for personalization. Across a wide range of applications and modalities, users with heterogeneous preferences interact with generative models in the hopes of producing content they find desirable. Compared to more traditional recommender systems, generative AI models offer new opportunities: instead of having to learn from user behavior, conversational interfaces enable direct user feedback. A user might, for example, instruct a tool to add more color to an image or make an email less formal. On the other hand, their flexibility introduces new challenges as well. Because they support such a broad set of downstream use cases, generative AI models perpetually face “cold-start” problems. Based on limited relevant information about a user’s preferences, a model must produce content that the user finds desirable.<sup>2</sup>

---

<sup>1</sup> Corresponding author

<sup>2</sup> In recognition of this lack of data, providers of AI chatbots have begun to save user data or connect to external sources to improve personalization.



## 7:2 When to Ask a Question

Consider an image-generation model tasked with the prompt: “Make a photorealistic image of a robot relaxing under a tree.” While the prompt encodes the broad strokes of what the user wants, it leaves far more questions unanswered (What kind of tree? What color is the robot? Is it reading a book?). In the absence of concrete information from the user, the model must fill in the blanks by (perhaps implicitly) making inferences about the user’s ideal image.

In a sense, these implicit inferences about preferences offer convenience to users. A user may hold weak preferences about the type of tree in the image or the color of the robot, but it may not be worth their time to communicate each of their weak preferences to a model. Viewed this way, generative models serve as (noisy and biased) decompression algorithms, taking in a compressed description of the user’s desired end product and inflating it by making guesses about all of the attributes that the user left unspecified [30]. Despite the fact that they may not produce a user’s ideal output, they provide utility by saving users time. A user might be able to, say, write an email without any AI assistance that perfectly aligns with their preferences, but they would be willing to accept a slightly worse-aligned email if a model could generate one with minimal supervision. In an idealized setting, a user would communicate precisely the preferences for which the marginal time required to do so is outweighed by the marginal utility gain from having those preferences satisfied.

Viewed this way, a rational user can simultaneously benefit from the use of generative models and still end up with suboptimal content. The user does not get exactly what they want, but AI assistance allows them to produce something “good enough” with dramatically less effort than they would otherwise need. While the user could continue iterating towards their ideal content, it is simply not worth their time to do so.

### Motivating examples

Throughout this paper, we will use several motivating examples to illustrate settings where such user-LLM interactions may occur.

► **Example 1 (Vacation planning).** A user is planning a trip to a city, and queries an LLM to ask about a suggested itinerary. Based on its training data, the LLM “knows” that visitors to this city tend to fall in a few rough patterns, either preferring to visit locations  $\{A, B, C, \dots\}$  (e.g., museums and historical sites) or a different set of locations  $\{X, Y, Z, \dots\}$  (e.g., restaurants and cafes). Because a user doesn’t know the city, she doesn’t know her general “type” beforehand. An LLM can learn more about her relative preferences by asking direct queries (“Would you prefer location  $A$  or  $X$ ?”) and then inferring the rest of her preferences.

► **Example 2 (Email writing).** A user is relying on an LLM to help her write an email. While each email is unique, they do tend to fall into specific patterns, such as “long, polite” emails or “short, direct” emails. While users may not respond well to being asked “Is this a polite email or not?”, they may be willing to respond to proxies, such as “Should this email begin with a salutation or directly jump into subject matter?”.

► **Example 3 (Image generation).** Consider a user who is relying on an LLM to help generate an image. Image generation is a very high-dimensional problem, but one with strong degrees of correlation (e.g. pixel-to-pixel correlation in output, but also common themes in image styles). While a user may not directly know or be able to articulate her preferences, she may be able to answer queries about style aspects or examples that she wishes to include.

In all of these examples, user preferences may be idiosyncratic, or may be tied to sensitive features, such as socioeconomic status, race, or gender. We seek to model how preference underspecification impacts user utility, with a particular focus on implications for fairness.

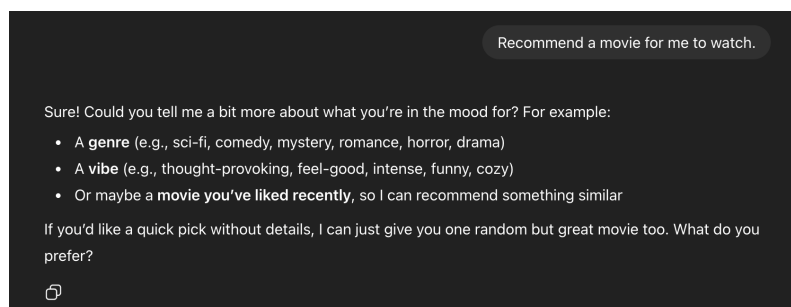
### System-level effects of personalization under uncertainty

We consider an idealized AI system that, given partial information about the user’s preferences, makes inferences based on its knowledge of the population distribution. A model might infer, for example, that “a flag next to the Statue of Liberty” refers to the American flag, while “a flag next to the Eiffel Tower” refers to the French flag.

When left to “fill in the blanks,” models can have subtle but profound implications on the utility, fairness, and overall diversity of a system. Inferred preferences may skew towards majority populations and tastes [9, 60, 26, 50, 37]. As a result, outputs will become homogeneous precisely in those preference dimensions left unspecified. Thus, the underspecification of preferences serves as a microfoundation for algorithmic homogenization [27, 6, 7].

### The impacts of user interaction

The challenges described – incomplete information, user fairness, and content homogeneity – are not unique to generative AI models. Similar concerns arise in more traditional recommender systems when dealing with limited user information. While this is true to some extent, a key differentiator of generative AI lies in its inherent interactivity. Unlike many traditional recommendation systems where user feedback is often implicit (e.g., clicks, purchases), content generation can involve multiple rounds of explicit interaction between the user and the model. A user can iteratively refine the generated output through natural language commands or other interactive modalities, providing direct feedback and steering the model towards their desired outcome. Moreover, a model can *actively solicit* the user’s preferences by asking additional questions. In fact, many state-of-the-art chatbots now exhibit this behavior (see Figure 1).



■ **Figure 1** A screenshot of ChatGPT prompting the user for additional information when an under-specified prompt is provided by the user. A decision to solicit information trades off additional user effort (to respond to the LLM’s question) for the benefits of better personalization.

### The present work: Modeling interactive content generation

We seek to understand how inferences about preferences impact the utility, fairness, and content diversity of AI systems. In our model, an AI system has two levers at its disposal. It can (1) solicit more information about a user’s preferences and (2) generate content by making inferences about underspecified preferences. At a high level, we might expect that soliciting information should increase both fairness and output diversity simply by more

faithfully representing user preferences. While this is true, additional information is not free; providing more information is costly for users (otherwise they would specify their complete preferences up-front). Thus, we build a formal framework under which we can pose and answer questions about when information-solicitation is “worth” the cost.

Our paper is organized as follows. After surveying related work in Section 2, we present our formal model in Section 3. In Section 4 we study this problem through the lens of maximizing the user’s expected welfare, in Section 5 we turn to broader notions of user welfare (e.g. relating to fairness), and in Section 6 we briefly discuss implications for output homogenization. In Section 7, we validate the qualitative predictions of our model with an empirical exercise. We conclude in Section 8 with a discussion of our results and future work.

## **2** Related Work

There has been a rich body of literature, both empirical and theoretical, on preference elicitation and social welfare in content creation.

First, the emerging area of pluralistic alignment (e.g. [48]) aims to align LLMs with a range of human preferences over outcomes (see related summary papers [54, 17, 25]). Most closely related to our work are papers that connect with voting and social choice literature to aggregate diverse user preferences (e.g. [10, 12, 46, 1, 16, 40, 8]). One key difference between most of this literature and our work is that it tends to focus on a one-shot setting where the goal of an algorithm is to represent multiple diverse viewpoints, whereas our focus is when the algorithm has the ability to interact with the human (through querying or critique of content) to better suit their preferences.

Another related area of literature is that of algorithmic monoculture or homogenization (e.g. [27, 6]), which studies the impact of a “monoculture” of algorithmic tools on societal welfare. Our paper relates to this body of work by demonstrating how an attempt to satisfy multiple diverse users with only partial information about preferences can lead to homogenization of outcomes. A closely related area of research studies human-LLM collaboration in creative tasks [18, 45, 58, 14, 49, 53, 39, 10]. Creative tasks, such as generating images or stories, are especially relevant given that there they are high-dimensional outputs that often no “right” answer. For example, users probably have much wider variability over their most-preferred art image, as compared to a factual query (e.g. “What is the capital of New York State?”).

Some empirical work focuses on when it may be appropriate to ask questions to clarify user intent: this has been studied both in traditional web search tools (e.g. [56, 24]) as well as more recently in the context of LLMs [52, 2, 57, 43, 34, 31, 38, 15, 33, 28, 51, 23], as well as summary papers [55, 13].

Finally, our work is also closely connected with both the areas of feature acquisition and of fairness in recommender systems. Feature acquisition (e.g. see survey [44]) studies ways to acquire features in order to improve predictions: it differs from our work in that it does not consider generation and critique as an alternative framework, even for papers focusing on feature acquisition in generative AI (e.g. [35]). For fairness in recommender systems, literature in this space studies tradeoffs related to fairness and utility in suggesting content to users (e.g. [29, 36, 47]), or ensuring sufficient diversity among suggested items (e.g. [32, 41, 59, 20]). Our paper differs from this body of work primarily in the mode of interaction: rather than suggesting content to users in a ranked list, in our setting users are allowed to interact directly with the algorithm (here, a LLM) and answer queries or critique generated content. Another difference is that recommender systems are typically selecting

items from a pre-existing set of content to suggest to users, while our content creation setting implicitly assumes that the LLM can return every possible type of output. The main goal of this paper is to study how such modes of interaction affect the fairness and utility guarantees that we can give users.

## 3 Model

### 3.1 Overview

We consider the interaction between two different agents – a user and an LLM. Given an initial prompt from the user, the user has preferences over the LLM’s response space. In general, this preference distribution can be high-dimensional and domain-specific. The preference space for emails is quite different from the preference space for movie recommendations. For simplicity, we will take the user’s initial query as given. Moreover, we assume that the LLM knows the distribution of user preferences over the response space, while the user merely knows their own preferences.

As an example of this communication asymmetry, consider the example of a user using an LLM to help plan their vacation given in Example 1. The user has a sense of their own travel preferences (i.e. what locations they might want to visit), but not exact information about what’s available. The LLM, on the other hand, has access to the full space of possible vacations but does not know the user’s specific desires. This mismatch creates a challenge for effective co-creation, as each party holds complementary but incomplete information.

In this model, the goal of the LLM is to generate outputs that are maximally aligned with a user’s preferences. A key ingredient of our model is interactivity: We allow for the LLM to solicit additional information about the user by asking additional questions. While this is costly – users do not like to answer questions – it allows the LLM to make better inferences about the user’s preferences. The LLM must therefore trade off the cost of information acquisition with the benefits it brings to personalization.

### 3.2 Preference Generation

Here, we introduce a stylized model of preferences to ground our study. Because the space of possible human preferences is vast, we focus on a simple, illustrative instantiation that captures the key properties of interest. For a given query, we represent each user’s preferences as a vector of  $n$  binary features, which we will refer to as their *preference vector*. A key property that our model captures is correlation between preferences: Information about the user’s preferences along one dimension enables better inferences about their preferences along another. In Example 1, for instance, users who prefer visiting location  $A$  may also enjoy visiting location  $B$ , while users who like location  $X$  may be more interested in also visiting location  $Y$ . In what follows, we will provide a minimal model that captures these relationships.

We assume that preference vectors are generated according to a particular Bernoulli mixture model e.g., [4]:

1. Users are randomly assigned to either cluster 0 with probability  $\alpha$  or cluster 1 with probability  $1 - \alpha$ . Users assigned to cluster 0 begin with all 0 preference vectors, and users assigned to cluster 1 begin with all 1 preference vectors.
2. Each feature preference is flipped independently with probability  $p < 0.5$ .

From this process, we can fully describe the probability that any preference vector occurs in the general population.

When thinking about the practicality of such a preference model, consider the vacation planning example. The different sets of locations given in Example 1 might correspond to high-cost vs. low-cost activities. Travelers might generally plan their trips on a tight budget or seek luxury experiences. However, those on a tighter budget might be open to splurging on one or two activities of interest, and those willing to spend more overall may still be looking to save in a few areas. Such behavior naturally fits into our model of preference generation, where cluster 0 and cluster 1 correspond to  $n$  high-cost and low-cost activities, and any individual deviates their standard budgetary habits with some probability. Although incorporating correlated deviations across preferences could provide a richer model of user variation, assuming independent randomness adequately captures heterogeneity within preference clusters for the purposes of our analysis.

More generally, this model is intended as a first step toward understanding the relationship between preference inference, information acquisition, and fairness; a full treatment of the richer preference structures that arise in specific application domains is left to future work.

### 3.3 User-LLM Interaction

Now that we have described the setup of feature preferences, we must define how the user and LLM exchange information. We consider an interaction model in which the user approaches the LLM with an initial task, but where the LLM does not know the human’s full preferences over outcomes for that specific task (e.g. the full binary string). The  $n$  features considered in the previous section are assumed not to be included in the initial task, and are unknown conditional on the task being defined.

To gain information about a user’s preference, the LLM will directly query the user for their preference values of  $k$  features, where  $0 \leq k < n$ .

Based on the information learned through querying, the LLM then generates some output to give to the user. All  $k$  of the feature values revealed by the user are assumed to match the LLM’s output, leaving  $n - k$  for the LLM to “guess” in its output.

We assume that the only way an LLM can obtain information about the user’s preferences is through querying individual features rather than directly querying a user’s cluster identity. There are several possible reasons this assumption can make sense, including when clusters do not have well-defined names or due to user privacy concerns (i.e. a user may not want to reveal their exact financial status, as in Example 1).

The user’s utility depends on (1) how many questions they had to answer, and (2) how close the generated output is to their true preferences. We assume that the LLM has full knowledge of the preference distribution (i.e., parameters  $p$  and  $\alpha$ ). Based on this assumption, we assume that the LLM’s decision problem is comprised of both of the following:

► **Definition 4** (Optimal response policy). *Given a specific set of  $k$  revealed features, the optimal response policy specifies how should the remaining features be filled in.*

► **Definition 5** (Optimal querying policy). *For a given  $n$ , the optimal querying policy determines the optimal number of features  $k$  to query (assuming the optimal response policy fills in the remaining features).*

### 3.4 Utility

Having guessed the user’s preferences along  $n - k$  dimensions, the LLM produces an output. We’ll use  $B$  to denote the disagreement between the output and the user’s true preferences – that is,  $B$  is the number of dimensions along which the LLM “guesses wrong.” Based on this quantity, we define a user’s utility as

$$U(k, B) = n - k - B. \tag{1}$$

In this notion of utility, the cost of a single query is taken to be the same as the benefit of identifying an additional preference correctly. Practically, the cost/benefit ratio does not have to be 1, and we will explore other ratios empirically in Section 7.

From this definition, the expected utility of a user is therefore a function only on the expected number of mismatched features in the output compared to their feature vector. Abusing notation slightly,

$$\mathbb{E}[U(k)] = n - k - \mathbb{E}[B]. \quad (2)$$

The expected value of  $B$  above is taken over both the possible sets of  $k$  revealed bits and all possible user preference vectors. As mentioned in the previous section, one of the LLM’s decision problem when maximizing utility is to determine which  $n - k$  bits to generate given the revealed bits and preference generation parameters. As mentioned previously, the policy that achieves the highest expected utility will be referred to as the optimal response.

We include a complete formalization in Appendix A.

### 3.5 Model Summary and Overview of Results

To summarize, our model consists of two clusters of users with noisy and opposing preferences. The LLM can choose to solicit information from the user, at some cost to user satisfaction. Given the user’s response, the LLM must infer the remainder of the user’s preferences.

With this model, we begin by characterizing the LLM’s welfare-maximizing policy in Section 4. How much information should it solicit from the user, and with that information, how should it make inferences about the user’s unknown preferences? The welfare-maximizing policy can lead to disparities in utility across users. In Section 5, we show that the LLM’s optimal response policy depends on the planner’s aversion to inequality: By asking more queries, the LLM can personalize better and reduce inequality. We consider the implications for content diversity in Section 6, showing that inequality-aversion can also lead to more diverse LLM outputs, reducing output homogeneity. Before we conclude, we conduct an empirical validation of our model in Section 7.

## 4 When Asking a Question is Helpful

We begin by asking two natural questions: what is the optimal response policy; and how do we choose  $k$ , the optimal querying policy? In the first part of this section, we will investigate how the optimal response policy is determined from the set of revealed observations. Then, we move onto answering the second question about determining the optimal number of queries to ask. We investigate how initial cluster sizes, determined by parameter  $\alpha$  in the preference generation function, affect the optimal  $k$  value. We observe that depending on the probability  $p$  with which each bit flips from its initial assignment, the optimal strategy can vary: in some settings, it is best to ask no questions at all, while in others, asking one or more queries yields higher utility. Intuitively, more queries are helpful if they allow the LLM to make better inferences about the remainder of the user’s preferences.

### 4.1 Optimal Response Policy

First, we note that the optimal response policy is actually quite straightforward.

► **Lemma 6.** *The optimal response policy first identifies the user’s cluster of origin, and then makes the most accurate prediction for the remaining features based on this cluster.*

Intuitively, because we consider a linear utility function, the utility-maximizing choice is to guess the most likely element for each of the user’s preferences. In the special case where  $\alpha = 1/2$  and users are equally likely to start out with an all-0s or all-1s preference vector, the optimal response policy is simply to guess the bit value that is more frequently revealed during querying.

### 4.2 Optimal Querying Policy

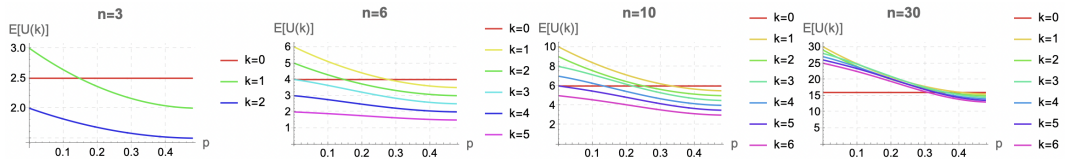
Now, let us move on to considering the optimal  $k$ , the number of questions to ask. This is determined by a trade-off between the value of acquiring information and its cost to user experience.

On the one hand, increasing  $k$  allows the LLM to learn more about a user’s preferences, enabling better personalization. On the other hand, soliciting information from the user requires effort on their part, which we model as costly.

#### 4.2.1 Equal-Sized Clusters

To analyze  $k$ , we will begin by considering a simplified case of  $\alpha$  (the probability of being one cluster) =  $1/2$ . In this setup, users are equally likely to come from cluster 0 or 1.

► **Lemma 7.** *When  $\alpha = 1/2$ , for all  $n > 2$ , there exists a noise rate  $p$  when it is optimal to ask a single query and a noise rate  $p$  where it is optimal to ask no queries.*



■ **Figure 2**  $\mathbb{E}[U(k)]$  vs  $p$  in the case of  $\alpha = 1/2$  for  $n = 3, 6, 10,$  and  $30$ . For each  $n$ , as the noise rate  $p$  increases, there comes a point where it optimizes expected utility to not ask any questions. In other words, as there is more uncertainty to the user’s cluster of origin based on the information, there is less value to asking questions. Additional figures are provided in Appendix C.

When the noise rate  $p$  is low, a single query often suffices to identify the user’s preference cluster with high probability. Once the cluster is known, the remaining bits are likely aligned with it, making subsequent predictions more accurate and the expected utility higher.

Conversely, as  $p$  approaches  $1/2$ , each bit in a user’s preference vector becomes nearly random. Recovering the user’s original cluster is thus both *harder* and *less informative*. Any query yields mostly noise, so the LLM’s guesses are no better than random while still incurring a utility cost for  $k > 0$ . Hence, when  $p$  is near  $1/2$ , it is often better to set  $k = 0$ .

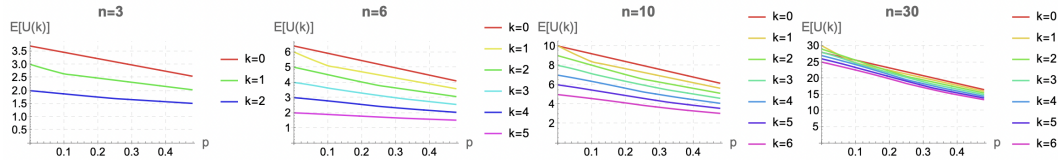
Together, these cases illustrate that while soliciting information can substantially improve inference when the signal is reliable, it becomes counterproductive when the signal is dominated by noise. In general, querying is valuable only when the information obtained is sufficiently strong to offset its cost.

#### 4.2.2 Unequal Clusters

Next, we consider the more general case of clusters occurring with probability  $\alpha$  and  $1 - \alpha$ . In light of Example 1, assume that there’s a minority group that is more cost-sensitive. Without additional information, the LLM’s “best guess” would be to satisfy the majority

group’s utility, which would lead to a more expensive itinerary. Soliciting more information in terms of preferences could make it more likely to satisfy the minority group’s preferences. However, depending on the noise rate, the additional cost in terms of querying may not be “worth it” when the dominant group is large.

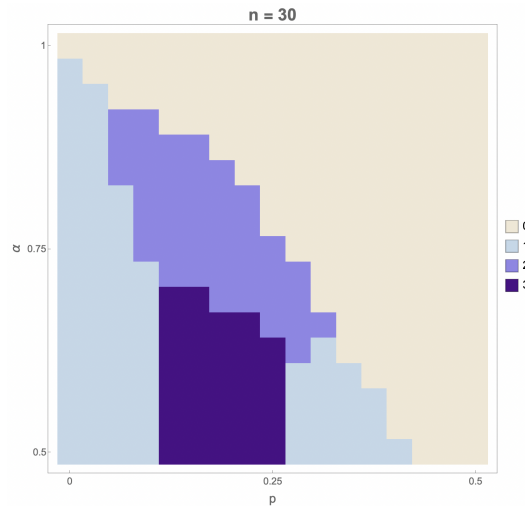
► **Lemma 8.** *There exist values of  $n$  where it maximizes expected utility to ask no questions for all values of  $p$ .*



■ **Figure 3**  $\mathbb{E}[U(k)]$  vs  $p$  in the case of  $\alpha = 0.9$  for  $n = 3, 6, 10,$  and  $30$ . When  $n$  is smaller,  $k = 0$  is the optimal querying policy regardless of  $p$ . When  $n$  is larger however, it can sometimes be better to ask a single question.

When  $\alpha$  is either close to 0 or 1, the niche set of users is significantly smaller than those with majority preferences, leading the model to have a much stronger prior that any user is part of the majority. When this prior belief is sufficiently strong, the utility cost paid from asking a single query might outweigh any information gained from the query. This can even be true for small noise rate  $p$  if the group of users with niche tastes is sufficiently small.

Figure 4 provides an overview of the landscape of optimal  $k$  values for a range of choices for  $\alpha \in [0.5, 1]$  and  $p \in [0, 0.5]$  for  $n = 30$ . We see here that for increased values of  $n$ , it can be useful to ask more than a single question, particularly for intermediate values of  $p$  and  $\alpha$ . In these cases, the prior belief about cluster membership is strong but not decisive, so the additional clarification provided by a second query meaningfully increases confidence in the remaining preference inferences.



■ **Figure 4** Optimal number of queries  $k$  for various values of  $p$  and  $\alpha$  and  $n = 30$ . The plot is symmetric for smaller values of  $\alpha$  from 0 to 0.5.

However, these results also make clear that under expected utility maximization, information acquisition is driven entirely by aggregate gains. When one preference cluster is sufficiently small, the model optimally forgoes clarification because the expected benefit is

too low. As a result, minority users are systematically under-served: their preferences are unlikely to be inferred correctly, even when the noise level is small. This raises concerns of fairness, as minority users have little to no chance of receiving outputs that align with their preferences. In the following section, we explore alternative objective functions that explicitly account for this aversion to inequality.

## 5 Fairness and Welfare Considerations

The prior section studied how to optimize for expected utility over a population. In this section, we turn to questions of fairness across a population. We study how both the optimal response policy and optimal querying policy change if we are optimizing for different notions of welfare.

### 5.1 Welfare

We model welfare using a single-parameter family of social welfare functions for any output and where the parameter  $\gamma$  represents the level of aversion to inequality. Within the context of this paper, we will be focusing on welfare functions that marginalize over all possible outputs given possible sets of  $k$  revealed features.

$$W(k) = \frac{1}{1-\gamma} \mathbb{E} \left[ \sum_{\text{users}} \text{Utility}(\text{user})^{1-\gamma} \right]. \quad (3)$$

Note that our social welfare function captures expected ex-post welfare – i.e., measured over *realized* outcomes, as opposed to ex ante expected outcomes. Intuitively, this is the difference between  $E[U^{1-\gamma}]$  vs.  $E[U]^{1-\gamma}$ , where  $U$  represents a user’s utility.

The exact formulation of the social welfare function we consider in this setup can be found in Appendix A. Overall, this family of social welfare functions are widely used in related literature in computer science (e.g. [40, 11, 42, 19]) and economics (e.g. [3, 5]). Choosing such an objective function is desirable because it directly encodes trade-offs between equity and efficiency, which also provides a parameter that can enable alignment with different ethical or policy priorities.

There are several natural choices of  $\gamma$  with such a welfare function:  $\gamma = 0$  yields a utilitarian welfare function (as was covered in the previous section),  $\gamma = 1$  yields the Nash welfare (the product of utilities), and  $\gamma = \infty$  implies maximin fairness.<sup>3</sup>

### 5.2 Optimal Response Policy

In principle, the LLM’s optimal response might be randomized. Given information provided by the user, the LLM’s welfare-maximizing response could be randomized. In fact, we show that this is not the case in our setting. Deterministic responses suffice:

► **Lemma 9.** *In order to maximize ex-post social welfare, the best course of action for an LLM is to deterministically return a single response given the revealed features.*

The intuition is straightforward: assume by contrast that the LLM used a randomized strategy. Users experience welfare based on the specific guesses made given revealed features. Because we are using ex-post social welfare, our utility over a randomized set of responses is

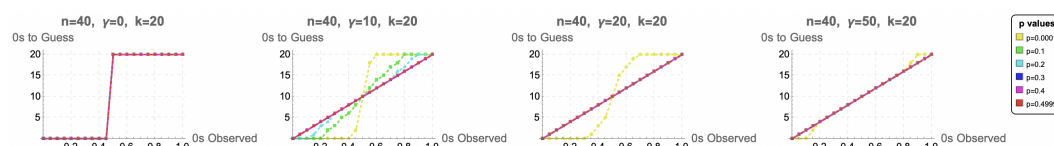
<sup>3</sup> More specifically, the Nash welfare is given by the limit of (3) as  $\gamma$  approaches 1.

simply equal to the expected utility, with randomness taken over our responses. Out of all possible guesses given revealed features, there will be a single one that maximizes welfare. By linearity of expectation, using a deterministic strategy of only returning those guesses given  $k$  revealed features has welfare at least as high as the original randomized strategy, and strictly higher in every case without ties in welfare between items.

Next, we consider the optimal response policy. Beyond the  $\gamma = 0$  case discussed in Section 4, interesting things start to happen to the optimal response policy as we increase  $\gamma$ . The optimal response policy for the purely utilitarian setting is no longer relevant as we prioritize fairness, and is now a function of  $\gamma$  as well. Let  $a$  be the number of 1's in response to the LLM's  $k$  initial queries. Then, denote the optimal number of 1's in the LLM's final output as  $f_\gamma(a)$ .

► **Lemma 10.** *The optimal response policy may produce a mixed output of 0s and 1s.*

In contrast the results given in Lemma 6, the optimal inequality-aware policy is to hedge, rather than simply cater to the majority tastes. As a result, the LLM will make a combination of choices that favor each cluster.



■ **Figure 5** The number of 0s to output based on the optimal response policy  $n - k - f_\gamma(a)$  as a function of  $a$  for various  $p$  values when  $\alpha = 1/2$ ,  $n = 40$ , and  $k = 20$ . As  $\gamma$  increases, the optimal policy for all noise rates gets closer to being linear in the composition of the observations.

As  $\gamma$  becomes increasingly large, however, the policy does not simply become outputting half 0s and half 1s to hedge its guesses towards both clusters. Rather, it reflects a trade-off between guessing equal numbers of 0s and 1s and making use of the information learned from the revealed bits. In contrast to the result of Lemma 6 where much of the solicited information may not directly impact the optimal response policy, the LLM actually makes use of all information learned from the revealed bits.

► **Theorem 11.** *For every fixed  $k \leq n/2$ , each set of revealed preferences  $a$  results in a distinct optimal response policy  $f_\gamma(a)$  for a maximin fairness objective ( $\gamma \rightarrow \infty$ ).*

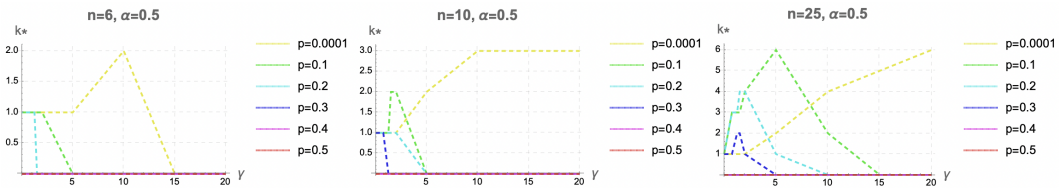
For some intuition about why this is true, consider that maximin utility is maximized in the setting when there are the fewest number of users who are maximally worst-off (i.e. receive utility 0). Moreover, users with the lowest utility are those that have exact opposite values for every term guessed by the LLM. As a result, this tells us that an LLM should guess the remaining bits to be the opposite of the least-likely user, given the bits we've observed. We thus arrive at the somewhat surprising result that an LLM should sometimes produce responses that vary sharply with the responses given when the goal is maximin fairness, depending on the preferences of the least-likely user, as we can notice in the rightmost panel of Figure 5. Importantly, this also clarifies why greater inequality aversion increases the value of information: the more sensitive the objective is to worst-case outcomes, the more the model directly benefits from information in queries. While this explains how the model should use information once it is obtained, it does not yet address the separate question of how much information should optimally be elicited in the first place, which we will consider next.

### 5.3 Optimal Querying Policy

Another interesting question to consider when generalizing our setup to prioritize fairness is how this affects the optimal number of queries  $k$  that the LLM should ask the user before generating content.

Intuitively, one might think that it is best to ask  $n$  queries and uniquely identify each user’s preferences, making sure everyone is equally happy. However, while this minimizes inequality, it does not maximize the inequality-averse social welfare we consider, as all users receive utility 0 in this plan. Under an inequality-averse social welfare objective, the goal is to balance minimizing inequality with maximizing total utility, leading to the adoption of different strategies.

► **Lemma 12.** *The optimal number of queries is non-monotonic as a function of  $\gamma$  for  $p > 0$ .*



■ **Figure 6** Optimal choice of  $k$  vs  $\gamma$  for various values of  $p$  and  $\alpha = 1/2$ . As  $\gamma$  initially increases, the optimal querying policy grows for small noise rates  $p$ . As  $\gamma$  continues to grow however, the optimal querying policy begins to decrease again.

This observation is not immediately clear. As mentioned above, one might expect that asking more questions would always be beneficial, as it provides the LLM with additional information and therefore a higher likelihood of correctly inferring the missing bits. However, this potential information gain must be weighed against the cost associated with each query. When the noise rate  $p$  is very small, increasing the number of queries initially improves performance as  $\gamma$  increases, as the LLM can more reliably identify the user’s cluster of origin. In Figure 6, this can be seen for the choices of flipping probabilities  $p = 0.1$  and  $p = 0.0001$ , as the choice of  $k$  initially increases for  $n = 10$  and  $n = 25$ . However, for any nonzero  $p$ , there comes a value of  $\gamma$  where the marginal benefit of additional queries is outweighed by their cost. This trend can be seen in Figure 6 when for noise rate  $p = 0.1$ , we see the optimal  $k$  fall back down to 0 as  $\gamma$  continues to grow for  $n = 10$  and  $n = 25$ . As  $\gamma \rightarrow \infty$ , the optimal strategy switches to be simply guessing all bits of a user’s preference vector. Intuitively, for a maximin welfare objective, information does not help.

## 6 Diversity and Homogeneity in Outputs

Finally, we turn to the connection between fairness and output diversity. We might intuitively expect a positive relationship between these: A preference for fairness should lead to a higher emphasis placed on personalization, thus reducing output homogenization. While this can be true in specific instances, our main result in this section will be to show that the intuition given above does not hold in general.

Recall that by Lemma 9, the LLM’s optimal response policy is a deterministic function of the strength of its belief that the user is from cluster 0 vs. cluster 1. As defined above,  $f_\gamma(a)$  specifies how many 1’s are in the LLM’s inferred preference vector, given the user’s response  $a$  to the LLM’s initial queries. To quantify content heterogeneity, we will characterize how

the entropy over  $f_\gamma(a)$  varies as  $\gamma$  increases. Formally, this is  $\mathbb{E}[H(f_\gamma(a))]$ , where  $H$  is the standard Shannon entropy and the randomness is over the user's response  $a$ . Note that the distribution of  $a$  depends on the number of queries  $k$  asked by the LLM. Here, we assume the LLM follows the optimal query policy for that value of  $\gamma$ . With this, we can state our final result:

► **Lemma 13.**  $\mathbb{E}[H(f_\gamma(a))]$  (the entropy of responses generated by the optimal query and response policies, given a particular  $\gamma$ ) is non-monotonic in  $\gamma$ .

To provide some intuition, consider what happens when  $\gamma$  grows from 0 to  $\infty$ . For  $\gamma = 0$ , by Lemma 6, the LLM infers the user's cluster of origin and caters exclusively to that cluster. This effectively yields two possible outputs, based on the inferred cluster of origin, and low output entropy. As  $\gamma$  increases, following the optimal query policy, the LLM may ask more queries and receive more information from the user. Because the LLM has more information to work with, it can produce a broader set of possible outputs. For example, when asking  $k = 2$  queries, there are 3 categories of responses that the LLM can get: all 0's, all 1's, or one 0 and one 1. By providing different responses in each of these cases, the LLM's output distribution spans a larger set of potential outputs. And as we show in our proof, there are instances where this initially leads to higher output diversity. However, as shown in Figure 6, the optimal number of queries eventually drops for sufficiently large  $\gamma$ . As a result, the LLM no longer solicits much information from the user, meaning that the diversity of its output distribution must go down.

In a sense, this result is quite related to the data processing inequality: the entropy of the output distribution of a deterministic function is upper-bounded by the entropy of its input distribution. With little information about an individual user's preferences, the LLM cannot offer much by way of output diversity (given that the optimal response policy is a deterministic function of the inputs.) Given more queries, as long as the optimal response policy is sufficiently sensitive to user-supplied information, output diversity increases.

These findings show that there is no simple relationship between optimizing for fairness and optimizing for output diversity. The relationship between fairness and output homogenization is mediated by the information-soliciting behavior of the LLM. If fairness leads to an increase in more information solicited from users, we should expect to see a corresponding increase in output diversity.

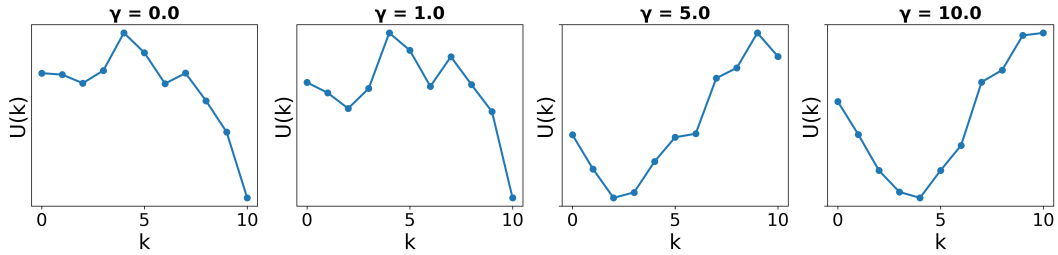
## 7 Empirics

The theoretical framework developed above relies on a highly stylized preference model. A natural question is whether the insights carry over to real preference data. Our primary goal with this section is to demonstrate that the optimal number of queries to ask depends on one's level of inequity aversion.

### 7.1 Experiment Setup

We consider the sushi preference dataset introduced in [21, 22], which contains the complete rankings of 10 different sushi by 5000 users. To convert ranking data to binary preference data, we encode the preference for any sushi ranked in the top 2 as a 1, and any sushi ranked in the bottom 8 as a 0.

In order to map to our model, the goal of our experiment is to identify a user's preferences on each of the 10 pieces of sushi rather than to just recommend sushi they'd like. To simulate preference elicitation, we began by training a decision tree classifier where each split is chosen



■ **Figure 7** Mean welfare vs  $k$  for increasing values of  $\gamma$  for per-query cost  $c = 0.3$ . The  $y$ -axis scale is omitted, as welfare is not comparable across  $\gamma$ ; we consider only within- $\gamma$  comparisons over  $k$ .

to maximize information gain about user preferences over the 10 sushi types. For a given  $k$ , a user’s preferences for the first  $k$  sushi types following tree order are revealed. The remaining  $10 - k$  preferences are selected to be the most likely values conditional on a user’s revealed preferences.

In a general setting, it makes sense to model differing costs of queries. Hence, to allow for more flexibility, we modify our utility function to include an explicit per-query cost  $c$ , leading to a revised function of

$$U(k, B) = n - c \cdot k - B, \quad (4)$$

where  $B$  is the number of preferences guessed incorrectly,  $k$  is the number of queries, and  $n = 10$ . We will use this generalized utility function in our definition of welfare and then analyze the optimal  $k$  for different values of  $\gamma$ .

## 7.2 Fairness-Utility Tradeoff

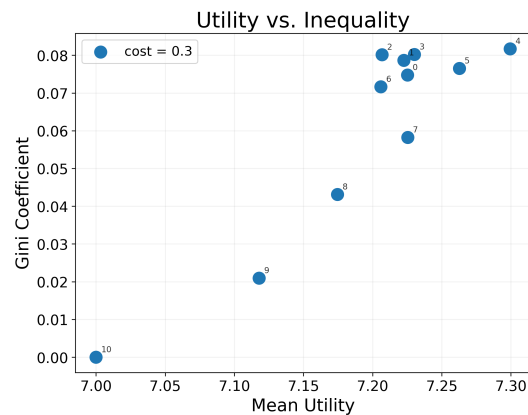
Figure 7 demonstrates how welfare changes as the optimal number of queries  $k$  increases for different values of the inequality aversion parameter  $\gamma$ . For purely utilitarian welfare, we can see that it is not until  $k = 3$  queries that the preference information learned is sufficiently valuable to increase welfare. Moreover, welfare becomes optimal at  $k = 4$  and quickly begins to trend down from its base level as the number of queries increases further.

When we become more inequality-averse and increase  $\gamma$ , we see that the optimal  $k$  changes. When  $\gamma$  is small, we see that the optimal  $k$  remains at 4. However, when  $\gamma = 5$ , we see that the optimal  $k$  increases to 9. As  $\gamma$  further increases to 10, we see that it becomes optimal for welfare to directly elicit a user’s entire set of preferences. As in our theoretical setup, the optimal information elicitation behavior depends on our level of inequality-aversion  $\gamma$ .

Figure 8 illustrates the underlying fairness–utility tradeoff. Setting  $k = 4$  maximizes mean utility but generates substantial inequality by favoring majority preferences. At the other extreme, fully eliciting preferences minimizes inequality but reduces utility due to query costs. Notably, because the marginal benefit of querying exceeds the cost in this setting, utility at  $k = 10$  remains strictly positive. Consequently, as  $\gamma \rightarrow \infty$ , higher levels of querying are always preferred.

## 8 Discussion

Our work introduces a mathematical framework for interaction between humans and LLMs, formalized under a particular model of preference generation. Within this framework, we assume that the LLM is able to obtain information about user preferences by directly querying



■ **Figure 8** Mean utility vs Gini coefficient for different values of  $k$  and  $c = 0.3$ .

users for their preferences – a behavior that is starting to be utilized by LLM chatbots in practice. We also consider an objective function that maximizes social welfare, with a parameter  $\gamma$  that controls the trade-off between fairness and utility. Despite its simplicity, our model reveals subtle relationships between fairness, utility, and output diversity – relationships that may practically emerge even when model assumptions are violated.

One key limitation of our model is its reliance on a highly stylized distribution of user preferences. The Bernoulli mixture model fits some data well but may not generalize to other domains. While our experiments in Section 7 validate some of our findings, further empirical testing is needed to assess when querying strategies are valuable for a specific domain. Moreover, the model assumes that an LLM has perfect knowledge of the full distribution of user preferences – however, this assumption may be imperfect, given that LLMs may rely on incomplete or biased data (our experiments in Section 7 explore relaxations of this assumption). Finally, the model treats all users as facing the same cost when answering queries, though in reality this burden varies with each individual’s circumstances and the specificity of the question.

Future work could generalize our model in a number of ways. One could consider more general preference models, differential costs for information, multi-round conversations, and stateful personalization. More generally, information solicitation by LLMs is fruitful direction for future research, and we seek to contribute to the literature with theoretical grounding that might help guide and motivate future empirical work.

## References

- 1 Parand A Alamdari, Soroush Ebadian, and Ariel D Procaccia. Policy aggregation. *Advances in Neural Information Processing Systems*, 37:68308–68329, 2024.
- 2 Chinmaya Andukuri, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah D Goodman. Star-gate: Teaching language models to ask clarifying questions. *arXiv preprint arXiv:2403.19154*, 2024. doi:10.48550/arXiv.2403.19154.
- 3 Anthony B Atkinson et al. On the measurement of inequality. *Journal of economic theory*, 2(3):244–263, 1970.
- 4 Christopher M Bishop. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- 5 Charles Blackorby and David Donaldson. Measures of relative equality and their meaning in terms of social welfare. *Journal of economic theory*, 18(1):59–80, 1978.

- 6 Rishi Bommasani, Kathleen A Creel, Ananya Kumar, Dan Jurafsky, and Percy S Liang. Picking on the same person: Does algorithmic monoculture lead to outcome homogenization? *Advances in Neural Information Processing Systems*, 35:3663–3678, 2022.
- 7 Francisco Castro, Jian Gao, and Sébastien Martin. Human-ai interactions and societal pitfalls. *arXiv preprint arXiv:2309.10448*, 2023. doi:10.48550/arXiv.2309.10448.
- 8 Daiwei Chen, Yi Chen, Aniket Rege, and Ramya Korlakai Vinayak. Pal: Pluralistic alignment framework for learning from heterogeneous preferences. *arXiv preprint arXiv:2406.08469*, 2024. doi:10.48550/arXiv.2406.08469.
- 9 Irene Chen, Fredrik D Johansson, and David Sontag. Why is my classifier discriminatory? *Advances in neural information processing systems*, 31, 2018.
- 10 Vincent Conitzer, Rachel Freedman, Jobst Heitzig, Wesley H Holliday, Bob M Jacobs, Nathan Lambert, Milan Mossé, Eric Pacuit, Stuart Russell, Hailey Schoelkopf, et al. Social choice should guide ai alignment in dealing with diverse human feedback. *arXiv preprint*, 2024. arXiv:2404.10271.
- 11 Cyrus Cousins. Revisiting fair-pac learning and the axioms of cardinal welfare. In *International Conference on Artificial Intelligence and Statistics*, pages 6422–6442. PMLR, 2023. URL: <https://proceedings.mlr.press/v206/cousins23a.html>.
- 12 Jessica Dai and Eve Fleisig. Mapping social choice theory to rlhf. *arXiv preprint arXiv:2404.13038*, 2024. doi:10.48550/arXiv.2404.13038.
- 13 Yang Deng, Lizi Liao, Wenqiang Lei, Grace Hui Yang, Wai Lam, and Tat-Seng Chua. Pro-active conversational ai: A comprehensive survey of advancements and opportunities. *ACM Transactions on Information Systems*, 43(3):1–45, 2025. doi:10.1145/3715097.
- 14 Anil R Doshi and Oliver P Hauser. Generative ai enhances individual creativity but reduces the collective diversity of novel content. *Science Advances*, 10(28):eadn5290, 2024.
- 15 Subhabrata Dutta, Timo Kaufmann, Goran Glavaš, Ivan Habernal, Kristian Kersting, Frauke Kreuter, Mira Mezini, Iryna Gurevych, Eyke Hüllermeier, and Hinrich Schuetze. Problem solving through human-ai preference-based cooperation. *arXiv preprint arXiv:2408.07461*, 2024. doi:10.48550/arXiv.2408.07461.
- 16 Luise Ge, Daniel Halpern, Evi Micha, Ariel D Procaccia, Itai Shapira, Yevgeniy Vorobeychik, and Junlin Wu. Axioms for ai alignment from human feedback. *arXiv preprint arXiv:2405.14758*, 2024. doi:10.48550/arXiv.2405.14758.
- 17 Jian Guan, Junfei Wu, Jia-Nan Li, Chuanqi Cheng, and Wei Wu. A survey on personalized alignment—the missing piece for large language models in real-world applications. *arXiv preprint arXiv:2503.17003*, 2025. doi:10.48550/arXiv.2503.17003.
- 18 Nava Haghighi, Sunny Yu, James Landay, and Daniela Rosner. Ontologies in design: How imagining a tree reveals possibilities and assumptions in large language models. *arXiv preprint arXiv:2504.03029*, 2025. doi:10.48550/arXiv.2504.03029.
- 19 Dan Hendrycks. *Introduction to AI safety, ethics, and society*. Taylor & Francis, 2025.
- 20 Amin Javari and Mahdi Jalili. A probabilistic model to resolve diversity–accuracy challenge of recommendation systems. *Knowledge and Information Systems*, 44:609–627, 2015. doi:10.1007/S10115-014-0779-2.
- 21 Toshihiro Kamishima. Nantonac collaborative filtering: recommendation based on order responses. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 583–588, 2003. doi:10.1145/956750.956823.
- 22 Toshihiro Kamishima and Shotaro Akaho. Efficient clustering for orders. In *Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06)*, pages 274–278. IEEE, 2006. doi:10.1109/ICDMW.2006.66.
- 23 Aum Kendapadi, Kerem Zaman, Rakesh R Menon, and Shashank Srivastava. Interact: Enabling interactive, question-driven learning in large language models. *arXiv preprint arXiv:2412.11388*, 2024. doi:10.48550/arXiv.2412.11388.

- 24 Kimiya Keyvan and Jimmy Xiangji Huang. How to approach ambiguous queries in conversational search: A survey of techniques, approaches, tools, and challenges. *ACM Computing Surveys*, 55(6):1–40, 2022. doi:10.1145/3534965.
- 25 Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A Hale. Personalisation within bounds: A risk taxonomy and policy framework for the alignment of large language models with personalised feedback. *arXiv preprint arXiv:2303.05453*, 2023. doi:10.48550/arXiv.2303.05453.
- 26 Jon Kleinberg and Sendhil Mullainathan. Simplicity creates inequity: implications for fairness, stereotypes, and interpretability. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 807–808, 2019. doi:10.1145/3328526.3329621.
- 27 Jon Kleinberg and Manish Raghavan. Algorithmic monoculture and social welfare. *Proceedings of the National Academy of Sciences*, 118(22):e2018340118, 2021. doi:10.1073/PNAS.2018340118.
- 28 Katarzyna Kobalczyk, Nicolas Astorga, Tension Liu, and Mihaela van der Schaar. Active task disambiguation with llms. *arXiv preprint arXiv:2502.04485*, 2025. doi:10.48550/arXiv.2502.04485.
- 29 Hannah Korevaar, Chris McConnell, Edmund Tong, Erik Brinkman, Alana Shine, Misam Abbas, Blossom Metevier, Sam Corbett-Davies, and Khalid El-Arini. Matched pair calibration for ranking fairness. *arXiv preprint arXiv:2306.03775*, 2023. doi:10.48550/arXiv.2306.03775.
- 30 Max Kreminski. Endless forms most similar: the dearth of the author in ai-supported art. *AI & SOCIETY*, pages 1–15, 2025.
- 31 Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Clam: Selective clarification for ambiguous questions with generative language models. *arXiv preprint*, 2022. arXiv:2212.07769.
- 32 Matevž Kunaver and Tomaž Požrl. Diversity in recommender systems—a survey. *Knowledge-based systems*, 123:154–162, 2017. doi:10.1016/J.KNOSYS.2017.02.009.
- 33 Belinda Z Li, Been Kim, and Zi Wang. Questbench: Can llms ask the right question to acquire information in reasoning tasks? *arXiv preprint arXiv:2503.22674*, 2025. doi:10.48550/arXiv.2503.22674.
- 34 Belinda Z Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. Eliciting human preferences with language models. *arXiv preprint arXiv:2310.11589*, 2023. doi:10.48550/arXiv.2310.11589.
- 35 Yang Li and Junier Oliva. Active feature acquisition with generative surrogate models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6450–6459. PMLR, 18–24 July 2021. URL: <https://proceedings.mlr.press/v139/li21p.html>.
- 36 Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. Fairness in recommendation: Foundations, methods, and applications. *ACM Transactions on Intelligent Systems and Technology*, 14(5):1–48, 2023. doi:10.1145/3610302.
- 37 Yunqi Li, Dingxian Wang, Hanxiong Chen, and Yongfeng Zhang. Transferable fairness for cold-start recommendation. *arXiv preprint arXiv:2301.10665*, 2023. doi:10.48550/arXiv.2301.10665.
- 38 Qianou Ma, Weirui Peng, Hua Shen, Kenneth Koedinger, and Tongshuang Wu. What you say= what you want? teaching humans to articulate requirements for llms. *arXiv preprint arXiv:2409.08775*, 2024. doi:10.48550/arXiv.2409.08775.
- 39 Kibum Moon, Adam Green, and Kostadin Kushlev. Homogenizing effect of large language model (llm) on creative diversity: An empirical comparison, 2024.
- 40 Kanad Pardeshi, Itai Shapira, Ariel D Procaccia, and Aarti Singh. Learning social welfare functions. *Advances in Neural Information Processing Systems*, 37:41733–41766, 2024.
- 41 Kenny Peng, Manish Raghavan, Emma Pierson, Jon Kleinberg, and Nikhil Garg. Reconciling the accuracy-diversity trade-off in recommendations. In *Proceedings of the ACM Web Conference 2024*, pages 1318–1329, 2024. doi:10.1145/3589334.3645625.

- 42 Feng Qi, Jia-Qiang Mei, Da-Feng Xia, and Sen-Lin Xu. New proofs of weighted power mean inequalities and monotonicity for generalized weighted mean values. *Math. Inequal. Appl.*, 3(3):377–383, 2000.
- 43 Linlu Qiu, Fei Sha, Kelsey R Allen, Yoon Kim, Tal Linzen, and Sjoerd van Steenkiste. Can language models perform implicit bayesian inference over user preference states? In *The First Workshop on System-2 Reasoning at Scale, NeurIPS’24*, 2024.
- 44 Arman Rahbar, Linus Aronsson, and Morteza Haghiri Chehreghani. A survey on active feature acquisition strategies. *arXiv preprint arXiv:2502.11067*, 2025. doi:10.48550/arXiv.2502.11067.
- 45 Judy Hanwen Shen and Carlos Guestrin. Societal impacts research requires benchmarks for creative composition tasks. *arXiv preprint arXiv:2504.06549*, 2025. doi:10.48550/arXiv.2504.06549.
- 46 Ali Shirali, Arash Nasr-Esfahany, Abdullah Alomar, Parsa Mirtaheri, Rediet Abebe, and Ariel Procaccia. Direct alignment with heterogeneous preferences. *arXiv preprint arXiv:2502.16320*, 2025. doi:10.48550/arXiv.2502.16320.
- 47 Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2219–2228, 2018. doi:10.1145/3219819.3220088.
- 48 Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, et al. A roadmap to pluralistic alignment. *arXiv preprint*, 2024. arXiv:2402.05070.
- 49 Kailas Vodrahalli and James Zou. Artwhisperer: A dataset for characterizing human-ai interactions in artistic creations. *arXiv preprint arXiv:2306.08141*, 2023. doi:10.48550/arXiv.2306.08141.
- 50 Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. Are big recommendation models fair to cold users? *arXiv preprint arXiv:2202.13607*, 2022. arXiv:2202.13607.
- 51 Shujin Wu, May Fung, Cheng Qian, Jeonghwan Kim, Dilek Hakkani-Tur, and Heng Ji. Aligning llms with individual preferences via interaction. *arXiv preprint arXiv:2410.03642*, 2024. doi:10.48550/arXiv.2410.03642.
- 52 Siwei Wu, Xiangqing Shen, and Rui Xia. A new dialogue response generation agent for large language models by asking questions to detect user’s intentions. *arXiv preprint arXiv:2310.03293*, 2023. doi:10.48550/arXiv.2310.03293.
- 53 Yueqi Xie, Tao Qi, Jingwei Yi, Xiyuan Yang, Ryan Whalen, Junming Huang, Qian Ding, Yu Xie, Xing Xie, and Fangzhao Wu. Measuring human contribution in ai-assisted content generation. *arXiv preprint*, 2024. arXiv:2408.14792.
- 54 Zhouhang Xie, Junda Wu, Yiran Shen, Yu Xia, Xintong Li, Aaron Chang, Ryan Rossi, Sachin Kumar, Bodhisattwa Prasad Majumder, Jingbo Shang, et al. A survey on personalized and pluralistic preference alignment in large language models. *arXiv preprint*, 2025. arXiv:2504.07070.
- 55 Yifei Yuan, Zahra Abbasiantaeb, Yang Deng, and Mohammad Aliannejadi. Query understanding in llm-based conversational information seeking. *arXiv preprint arXiv:2504.06356*, 2025. doi:10.48550/arXiv.2504.06356.
- 56 Hamed Zamani, Bhaskar Mitra, Everest Chen, Gord Lueck, Fernando Diaz, Paul N Bennett, Nick Craswell, and Susan T Dumais. Analyzing and learning from user interactions for search clarification. In *Proceedings of the 43rd international acm sigir conference on research and development in information retrieval*, pages 1181–1190, 2020. doi:10.1145/3397271.3401160.
- 57 Michael JQ Zhang, W Bradley Knox, and Eunsol Choi. Modeling future conversation turns to teach llms to ask clarifying questions. *arXiv preprint*, 2024. arXiv:2410.13788.
- 58 Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. Noveltybench: Evaluating creativity and diversity in language models. *arXiv preprint*, 2025. arXiv:2504.05228.

- 59 Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515, 2010.
- 60 Ziwei Zhu, Jingu Kim, Trung Nguyen, Aish Fenton, and James Caverlee. Fairness among new items in cold start recommender systems. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 767–776, 2021. doi:10.1145/3404835.3462948.

## A Derivations

### A.1 Utility

The full derivation conditions on cluster membership and marginalizes over the  $k$  revealed bits. By symmetry, the cluster 1 case follows by exchanging  $p$  and  $1 - p$ . This yields:

$$\begin{aligned} \mathbb{E}[U(k)] = n - k - \alpha \cdot \sum_{a=0}^k \binom{k}{a} (1-p)^{k-a} p^a \cdot (f(a) \cdot p + (n - k - f(a))(1-p)) \\ - (1 - \alpha) \cdot \sum_{a=0}^k \binom{k}{a} (1-p)^a p^{k-a} \cdot (f(a) \cdot (1-p) + (n - k - f(a)) \cdot p). \end{aligned}$$

### A.2 Welfare

To obtain an expression for welfare based on the general form given in Equation 3, we can use much of the same logic as in the previous section. The main practical differences between the expression for expected utility and the expression for welfare is the inclusion of the  $\gamma$  parameter which the utility is raised to. Due to the presence of this parameter, we can no longer apply linearity of expectation to pull out the  $n - k$  in the expression for  $W(k)$ . Hence, we must directly plug in the full expression for utility into our summation.

As in the utility derivation, conditioning on  $a$  revealed 1s for a cluster 0 user, define

$$\begin{aligned} G_{0,\gamma}(z, p) \triangleq \sum_{\substack{j \leq z \\ \ell \leq n-k-z}} \binom{z}{j} (1-p)^{z-j} p^j \cdot \binom{n-k-z}{\ell} (1-p)^\ell p^{n-k-z-\ell} \\ \cdot (n - k - j - \ell)^{1-\gamma}. \end{aligned}$$

From here, let  $W_0(k, \gamma)$  denote the welfare given that a user is from cluster 0. With these definitions and conditioned on seeing  $a$  1s, we arrive at the following welfare taking the form of our family of welfare functions

$$W_0(k, \gamma \mid a \text{ revealed 1s}) = \frac{1}{1-\gamma} \cdot G_{0,\gamma}(f_\gamma(a), p).$$

Marginalizing over all possible sets of  $k$  revealed features, we are able to write conditional welfare taken over the randomness of  $k$  as

$$W_0(k, \gamma) = \frac{1}{1-\gamma} \cdot \sum_{a=0}^k \binom{k}{a} (1-p)^{k-a} p^a \cdot G_{0,\gamma}(f_\gamma(a), p).$$

Similarly to before, we can exploit the symmetry of the preference generation process for the case where a user originates from cluster 1, and express the conditional welfare for cluster 1 as

$$W_1(k, \gamma) = \frac{1}{1-\gamma} \cdot \sum_{a=0}^k \binom{k}{a} (1-p)^a p^{k-a} \cdot G_{1,\gamma}(f_\gamma(a), 1-p).$$

## 7:20 When to Ask a Question

This expression is nearly identical to  $W_0(k, \gamma)$ , except for the  $p$  and  $1 - p$  values are flipped, in line with the different starting clusters. Finally, marginalizing once more over cluster of origin, we are able to get a full expression for welfare as

$$W(k, \gamma) = \frac{1}{1 - \gamma} (\alpha \cdot W_0(k, \gamma) + (1 - \alpha) \cdot W_1(k, \gamma)). \quad (5)$$

### **B** Proofs

#### B.1 Proof of Lemma 6

► **Lemma 6.** *The optimal response policy first identifies the user's cluster of origin, and then makes the most accurate prediction for the remaining features based on this cluster.*

**Proof.** In order to prove the optimal response policy in the  $\gamma = 0$  case, consider the basic definition of expected utility given in Equation 3 as

$$\mathbb{E}[U(k)] = n - k - \mathbb{E}[b],$$

where  $b$  is the number of bits that were guessed incorrectly. The expected number of mistakes can be written as the sum of the probabilities of guessing each bit incorrectly. Moreover, as all bits are flipped during our preference generation process independently with the same probability  $p$ , we can express the expected number of mistakes as simply

$$\mathbb{E}[b] = (n - k) \cdot \Pr(\text{single bit is guessed incorrectly}).$$

Thus, maximizing expected utility is equivalent to minimizing the probability of an incorrect guess for each bit.

Because all bits are identically distributed and independent, the same optimal decision rule applies to each bit. The optimal guess for a bit is therefore the value (either 0 or 1) that maximizes its posterior probability given the revealed information. Using Bayes' rule, this corresponds to choosing the more likely bit value conditional on the observed revealed bits. Hence, we can claim that the optimal response policy in the  $\gamma = 0$  case is to always guess the same value based on revealed bits. ◀

#### B.2 Proof of Lemma 7

► **Lemma 7.** *When  $\alpha = 1/2$ , for all  $n > 2$ , there exists a noise rate  $p$  when it is optimal to ask a single query and a noise rate  $p$  where it is optimal to ask no queries.*

**Proof.** Consider the plot given in Figure 2. We can see that for  $n = 3$ , it is best to ask a single query when  $p = 0$ . However, in that same  $n = 3$  plot when  $p = 0.5$ , it is best to ask no queries in order to maximize utility. ◀

#### B.3 Proof of Lemma 8

► **Lemma 8.** *There exist values of  $n$  where it maximizes expected utility to ask no questions for all values of  $p$ .*

**Proof.** Consider the plot given in Figure 3. We see that both for  $n = 3$  and  $n = 6$ , it always maximizes expected utility to ask no questions and just guess on all bits. ◀

## B.4 Proof of Lemma 9

► **Lemma 9.** *In order to maximize ex-post social welfare, the best course of action for an LLM is to deterministically return a single response given the revealed features.*

**Proof.** Assume the LLM returns a randomized response. First, utilizing the definition of welfare given in Equation 5 and the derivation of this definition in the previous section, recall

$$G_{0,\gamma}(z,p) = \sum_{\substack{j \leq z \\ \ell \leq n-k-z}} \binom{z}{j} (1-p)^{z-j} p^j \cdot \binom{n-k-z}{\ell} (1-p)^\ell p^{n-k-z-\ell} \cdot (n-k-j-\ell)^{1-\gamma}.$$

This is the expected welfare for a cluster 0 member if we choose  $z$  of the remaining  $n-k$  to be 0s. From this definition, the deterministic welfare was given by

$$W_0(k,\gamma) = \sum_{a=0}^k \binom{k}{a} (1-p)^{k-a} p^a \sum_{z=0}^{n-k} 1[f_\gamma(a) = z] G_{0,\gamma}(z,p).$$

To adapt welfare to randomized policies, define  $q_\gamma(a,z) = \Pr[f_\gamma(a) = z]$ . In a deterministic policy, we would constrain  $q$ 's to be either 0 or 1. For randomized policies, welfare for users from cluster 0 becomes

$$W_0(k,\gamma) = \sum_{a=0}^k \binom{k}{a} (1-p)^{k-a} p^a \sum_{z=0}^{n-k} q_z(a,\gamma) G_{0,\gamma}(z,p).$$

Now, let  $z^*$  denote the number of 0s that is optimal to guess for the remaining bits given  $z \in [0, n-k]$ . Formally, choose  $z^* = \arg \max_{0 \leq z \leq n-k} G_{0,\gamma}(z,p)$ . Then for users from cluster 0, the policy that deterministically returns  $z^*$  0s has welfare at least as large as the original randomized distribution over  $q$ s and strictly larger in every situation except for the edge case where whenever the distribution of  $q$ s has support only over choices of  $z$  with  $G_{0,\gamma}(z,p) = G_{0,\gamma}(z^*,p)$ . The result for cluster 1 follows by a symmetrical argument with  $G_{1,\gamma}(z,p)$ . ◀

## B.5 Proof of Lemma 10

► **Lemma 10.** *The optimal response policy may produce a mixed output of 0s and 1s.*

**Proof.** Consider the plot of optimal response policy as a function of  $\gamma$  given in Figure 11. For  $\gamma = 10$  in the row second from the left, we can see in the case where  $n = 6$  and  $k = 3$ , it is best to output some number at least a single 0 *and* a single 1 when both a 1 and 0 are observed. ◀

## B.6 Proof of Theorem 11

► **Theorem 11.** *For every fixed  $k \leq n/2$ , each set of revealed preferences  $a$  results in a distinct optimal response policy  $f_\gamma(a)$  for a maximin fairness objective ( $\gamma \rightarrow \infty$ ).*

**Proof.** To prove that we always utilize the information learned from the revealed features as  $\gamma \rightarrow \infty$  and  $k \leq n/2$ , we solve for the precise optimal response policy and show that it depends on  $a$ , the number of 1s revealed. Without loss of generality, shift utility up by 1 to  $(n-k-j-\ell+1)$ . Now, consider what happens to the expression  $(n-k-j-\ell+1)^{1-\gamma}$  as

## 7:22 When to Ask a Question

$\gamma$  increases. As  $n - k - j - \ell > 1$ , the utility will go to 0. However, when  $n - k = j + \ell$ , then utility will be exactly 1. Thus, in the limit as  $\gamma \rightarrow \infty$ , only the terms in the double sum where  $j + \ell = n - k$  survive. Because by definition we know  $j + \ell$  can never exceed  $n - k$ , we do not need to worry about negative terms in our expression for welfare.

The only term where  $n - k = j + \ell$  is exactly the case where  $j = f_\gamma(a)$  and  $\ell = n - k - f_\gamma(a)$ . Simplifying our double sum to this single term, we can express welfare as

$$\begin{aligned} W_0(k, \infty) &= \frac{1}{1 - \gamma} \sum_{a=0}^k \binom{k}{a} (1 - p)^{k-a} p^a \\ &\quad \cdot \left( \binom{f_\gamma(a)}{j} (1 - p)^{f_\gamma(a)-j} p^j \cdot \binom{n-k-f_\gamma(a)}{\ell} (1 - p)^\ell p^{n-k-f_\gamma(a)-\ell} \right), \\ W_1(k, \infty) &= \frac{1}{1 - \gamma} \sum_{a=0}^k \binom{k}{a} (1 - p)^a p^{k-a} \\ &\quad \cdot \left( \binom{f_\gamma(a)}{j} (1 - p)^j p^{f_\gamma(a)-j} \cdot \binom{n-k-f_\gamma(a)}{\ell} (1 - p)^{n-k-f_\gamma(a)-\ell} p^\ell \right). \end{aligned}$$

This can be further simplified to

$$\begin{aligned} W_0(k, \infty) &= \frac{1}{1 - \gamma} \sum_{a=0}^k \binom{k}{a} (1 - p)^{k-a} p^a \left( p^{f_\gamma(a)} \cdot (1 - p)^{n-k-f_\gamma(a)} \right), \\ W_1(k, \infty) &= \frac{1}{1 - \gamma} \sum_{a=0}^k \binom{k}{a} (1 - p)^a p^{k-a} \left( (1 - p)^{f_\gamma(a)} \cdot p^{n-k-f_\gamma(a)} \right). \end{aligned}$$

Now, note that as  $\gamma \rightarrow \infty$ , then  $\frac{1}{1-\gamma}$  approaches 0 from the negative side. So if we just want to maximize welfare, we can replace  $\frac{1}{1-\gamma}$  with  $-1$ . With this and combining  $W_0$  and  $W_1$ , we can write the total expression we wish to maximize as

$$W(k, \infty) = - \sum_{a=0}^k \binom{k}{a} \left( \alpha \cdot p^{f_\gamma(a)} \cdot (1 - p)^{n-k-f_\gamma(a)} + (1 - \alpha) \cdot (1 - p)^{f_\gamma(a)} \cdot p^{n-k-f_\gamma(a)} \right).$$

This expression will be maximized by choosing a policy to minimize each term within the summation. As can be seen by the dependence of  $a$  in the exponent of the terms,  $a$  influences the choice of policy. To minimize each term, let  $r := \frac{p}{1-p}$  and let us express

$$\begin{aligned} g(f_\gamma(a)) &= \alpha \cdot p^{f_\gamma(a)} \cdot (1 - p)^{n-k-f_\gamma(a)} + (1 - \alpha) \cdot (1 - p)^{f_\gamma(a)} \cdot p^{n-k-f_\gamma(a)} \\ &= \alpha \cdot (1 - p)^n \cdot r^{f_\gamma(a)+a} + (1 - \alpha) \cdot p^n \cdot r^{-f_\gamma(a)-a} \\ &= (1 - p)^n \left( \alpha \cdot r^{f_\gamma(a)+a} + (1 - \alpha) \cdot r^{n-f_\gamma(a)-a} \right). \end{aligned}$$

To minimize this expression, we can take the derivative and set it to 0. Ignoring any part that does not depend on  $f_\gamma(a)$  gives us a derivative of

$$\begin{aligned} g'(f_\gamma(a)) &= \alpha \cdot \log(r) \cdot r^{f_\gamma(a)+a} + (1 - \alpha) \cdot (-1) \cdot \log(r) \cdot r^{n-f_\gamma(a)-a} \\ &= \log(r) \cdot \left( \alpha \cdot r^{f_\gamma(a)+a} - (1 - \alpha) \cdot r^{n-f_\gamma(a)-a} \right). \end{aligned}$$

Setting this equal to 0, we get

$$\log(r) \cdot \left( \alpha \cdot r^{f_\gamma(a)+a} - (1 - \alpha) \cdot r^{n-f_\gamma(a)-a} \right) = 0.$$

If  $p = 1/2$ ,  $r = 1$  and  $\log(r) = 0$ , then all points are stationary points. However, when  $p \neq 1 - p$ , then a single stationary point exists when

$$f_\gamma(a) = \frac{1}{2} \left( n - 2a + \log_r \left( \frac{1 - \alpha}{\alpha} \right) \right).$$

When  $\alpha = 1/2$ , this implies  $f_\gamma(a)$  should be  $n/2 - a$ . This is always a non-negative value, as we assume  $k \leq n/2$ . When  $\alpha \neq 1/2$ , this implies that as  $\alpha$  increases,  $f_\gamma(a)$  decreases. As the second derivative of  $g$  is positive for valid  $p$  and  $\alpha$  values, we know the inflection points described must be a minimum. If it is not possible to set  $f_\gamma(a)$  as described above within the constraint that  $f_\gamma(a) \in [0, n - k]$ , then  $f_\gamma(a)$  should be set as close to as possible to the calculated expression to minimize  $g$ .

To conclude, we have provided an exact formulation for the optimal response policy when  $\gamma \rightarrow \infty$ . As it depends on  $a$  in all cases, we can claim that the optimal response policy for maximin fairness will never ignore observations. ◀

## B.7 Proof of Lemma 12

► **Lemma 12.** *The optimal number of queries is non-monotonic as a function of  $\gamma$  for  $p > 0$ .*

**Proof.** Consider the plot given in Figure 12. We can see that when  $n = 3$  and  $p = 0.1$ , the optimal number of queries to ask decreases from 1 to 0 as  $\gamma$  increases. ◀

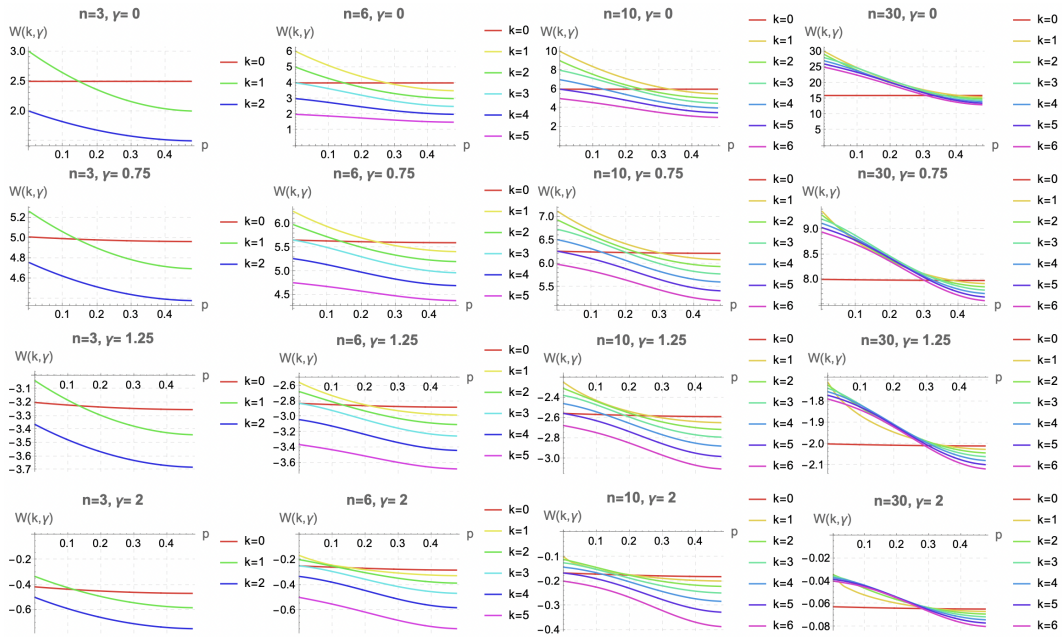
## B.8 Proof of Lemma 13

► **Lemma 13.**  *$\mathbb{E}[H(f_\gamma(a))]$  (the entropy of responses generated by the optimal query and response policies, given a particular  $\gamma$ ) is non-monotonic in  $\gamma$ .*

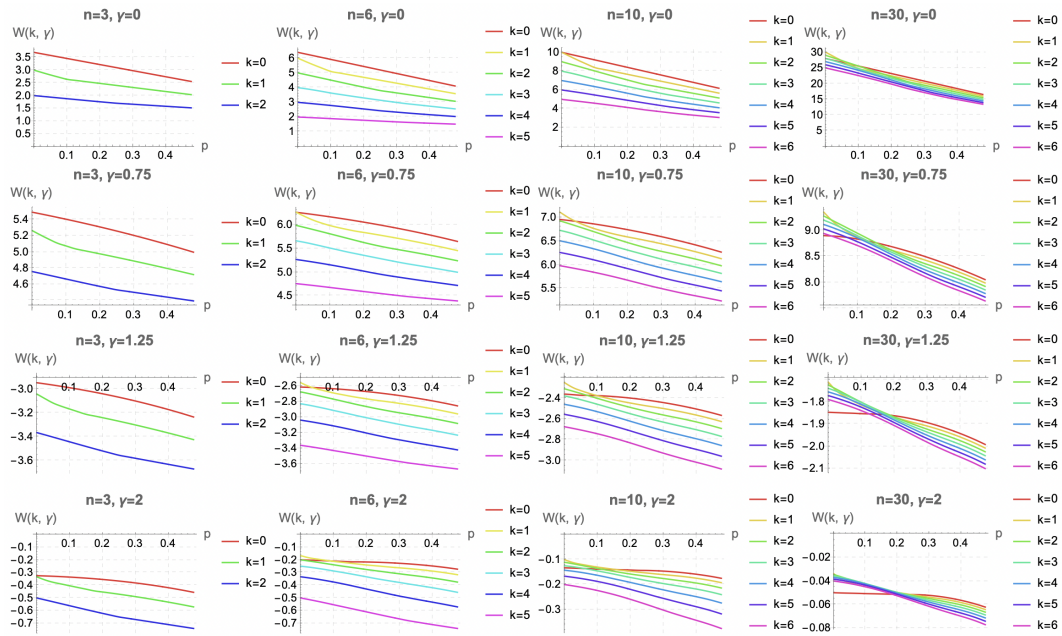
**Proof.** Consider the plot given in Figure 12. We can see that when  $n = 25$  and  $p = 0.1$ , the optimal number of queries to ask increases from 4 to 6 as  $\gamma$  increases from 2 to 5. However, in this same plot when  $k$  increases from 5 to 10, the optimal number of queries to ask decreases back to down to 2. Computing the entropy based on the optimal response policy first when  $k = 4$  and  $\gamma = 2$ , we find that  $H = 1.23$ . When  $k = 6$  and  $\gamma = 5$ , the optimal response policy gives an entropy of  $H = 2.42$ . However, when  $k = 2$  and  $\gamma = 10$ , the limited number of possible responses the model can give results in an entropy of  $H = 1.50$ . From these three values, we can see that the entropy is a non-monotonic function of  $\gamma$ . ◀

## C Additional Plots

Using the full expression for welfare with variable  $\gamma$  given in Equation 5, the following plots were generated to demonstrate how welfare is affected as various parameters are changed.



■ **Figure 9** Welfare  $W(k, \gamma)$  vs flipping probability  $p$  in the case of  $\alpha = 1/2$  for various values of  $\gamma$ . The choice of  $k$  leading to the optimal query length is included in each plot.



■ **Figure 10** Welfare  $W(k, \gamma)$  vs  $p$  in the case of  $\alpha = 0.9$  for various values of  $\gamma$ . The choice of  $k$  leading to the optimal query length is included in each plot.

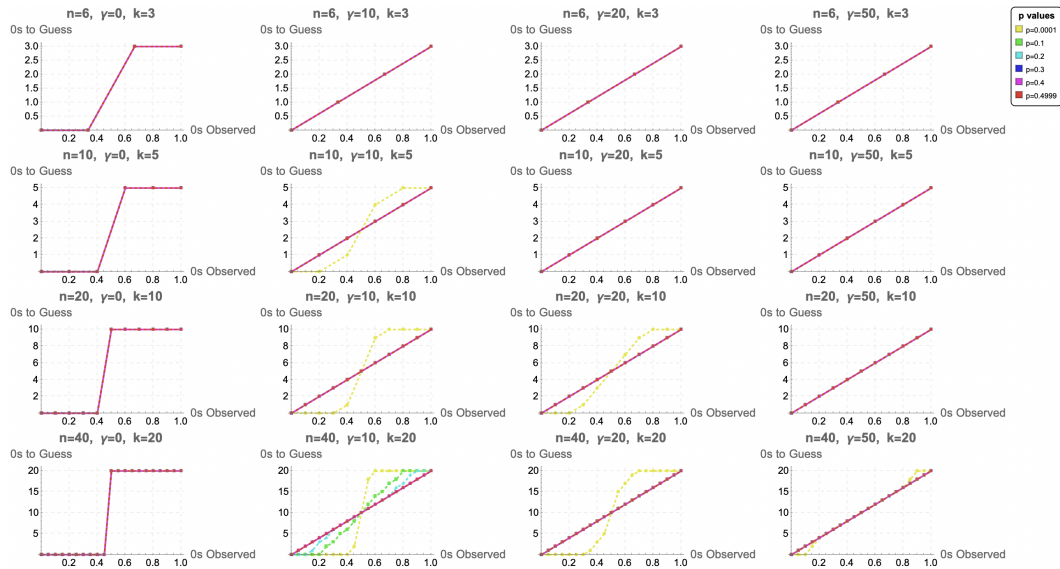


Figure 11 The number of 0s to output based on the optimal response policy  $n - k - f_\gamma(a)$  as a function of  $a$  for various  $p$  values when  $\alpha = 1/2$  and  $k = 3$ .

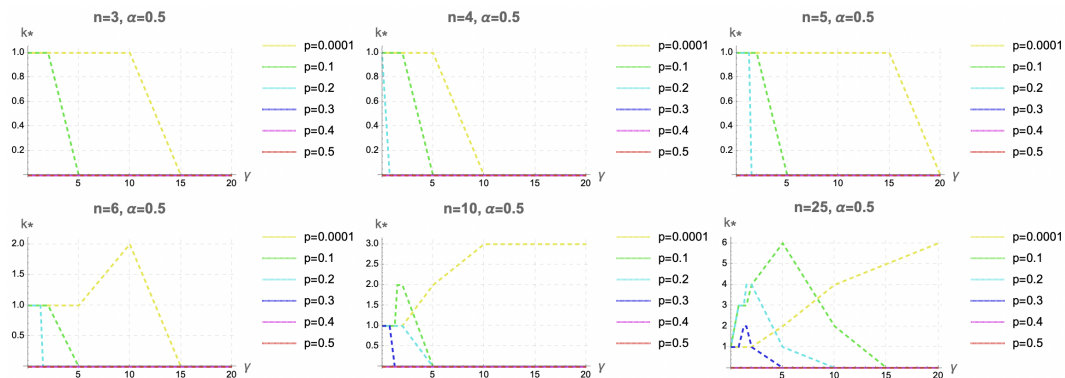


Figure 12 Optimal choice of  $k$  vs  $\gamma$  for various values of  $p$  and  $\alpha = 1/2$ .

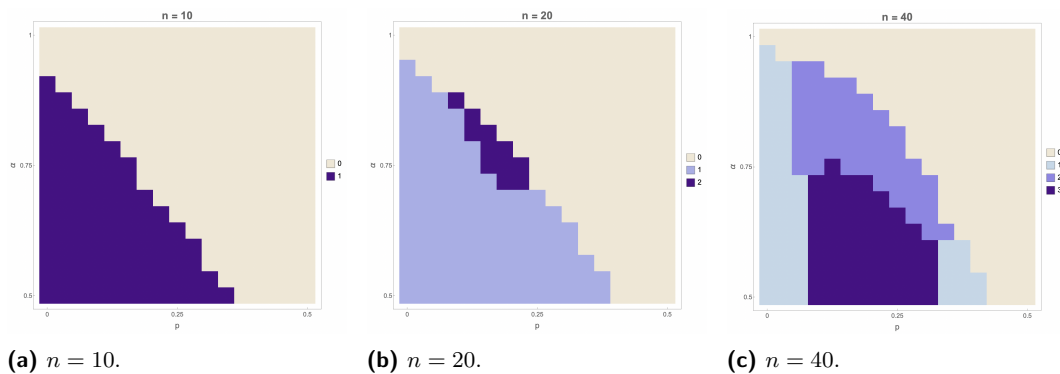


Figure 13 Optimal number of queries  $k$  for various values of  $p$  and  $\alpha$  across different values of  $n$ .