Algorithmic Persuasion Through Simulation:
Information Design in the Age of Generative AI

Keegan Harris*[1], Nicole Immorlica[2], Brendan Lucier[2], and Aleksandrs Slivkins[2]

[1]Carnegie Mellon University
[2]Microsoft Research
keeganh@cs.cmu.edu
{nicimm,brlucier,slivkins}@microsoft.com

Abstract

We study a Bayesian persuasion game in which a sender reveals information about a payoff-relevant state to a receiver who may have exogenous sources of information about the state which are unknown to the sender. The receiver then takes an action which affects the payoffs of both players. Motivated in part by recent advances in Generative AI, we consider a setting where the sender has access to an *oracle* (e.g. a large language model) through which they can indirectly learn about the receiver's beliefs by *querying* before sending a signal. Our goal is to explore how access to this additional source of information can help the sender design more persuasive signaling policies. We explore two settings: one in which the sender can *adaptively* query the oracle, and one in which they must make all of their queries *up front*. When the sender is allowed to adaptively query the oracle, we show that the optimal querying policy may be found via backward induction. In contrast, we show that determining the optimal querying policy is NP-Complete when the sender must make all of their queries up front. Motivated by this result, we analyze the effects of querying on the sender's expected utility in the non-adaptive setting. We observe that in a family of persuasion instances with binary states and actions, there is decreasing marginal sender utility for each query made, and the optimal querying policy may be found in polynomial time. In the general setting, we show that decreasing marginal sender utility *does not* hold, which implies that the greedy algorithm, a natural heuristic often used for approximation, provably fails. Finally, we explore the effects of committing to a querying policy and show that this additional commitment power can significantly benefit the sender.

*Some of the results were obtained while the author was an intern at Microsoft Research.

Contents

1 Introduction

Bayesian persuasion [26, 25] is a game of asymmetric information between two players: an informed *sender*, who observes the *state* of the world, and an uninformed *receiver*, who does not see the state but takes an *action*. The payoffs of both players depend on both the world's state and the receiver's action. The game proceeds as follows: The sender commits to a *signaling policy*, i.e. a mechanism for revealing information to the receiver about the state of the world, before the state is realized. Once the state is realized, the sender sends a signal (or *message*) to the receiver according to their signaling policy. Upon receiving the signal, the receiver updates their belief about the state of the world, and takes an action. One motivating example of the Bayesian persuasion framework is the interaction between a *seller* of a product (the sender) and a potential *buyer* (the receiver), where the state of the world is the quality of the product (e.g. high/low quality). The seller would always like to sell the product to the buyer, but the buyer only wants to purchase the product if it is of high quality.[1]

The sender's payoff-maximizing signaling policy often depends on information about the receiver which may not be known to the sender; for example, the receiver's *belief* about the true state of the world. This observation has motivated a line of work on *robust* Bayesian persuasion (e.g. [19, 37, 24]), which broadly aims to determine the sender's optimal signaling policy when almost *no information* is known about the receiver. While robust signaling policies are applicable in a wider range of settings when compared to methods which require *full information* about the receiver, they are not without their drawbacks. Perhaps unsurprisingly, signaling policies which use information about the receiver are more persuasive and are often simpler when compared to their robust counterparts. This raises the question:

How can the sender design simple, persuasive signaling policies when they initially have minimal information about the receiver?

One useful observation is that in many settings of interest, the sender is not at either end of the informational spectrum, but instead may *acquire* additional information about the receiver through external sources. For instance, the sender may be able to learn about the receiver's beliefs through one of the following processes:

1. **Receiver simulation using Generative AI and machine learning.** Given sufficient data about the receiver, the sender may be able to predict the receiver's behavior under different settings using machine learning models. There has been recent interest in using large language models (LLMs) as *simulated economic agents* which can be endowed with preferences or beliefs and used to obtain insights about how humans may behave in real-world strategic scenarios (e.g. [23, 20]). Suppose the sender uses an LLM, fine-tuned with relevant data about the receiver, to try out different methods of information disclosure before attempting to persuade the receiver. As another example, suppose the seller is an online marketplace who wants to bring a new product to market and the buyer is a marketplace user. While the seller may initially be unsure about how the new product will be received, they may be able to use the user's purchase history on the platform to predict something about how the user would respond to a sales pitch for the new product. Finally if the receiver is itself an AI agent, it may be possible to simulate the receiver directly.

2. **Simulation as a metaphor for exploration.** The sender may be able to learn about the behavior of the receiver through their interactions with other receivers in the past. For example, a startup may test out their sales pitch on smaller venture capital firms before trying to persuade a larger firm to fund their business. A company may run a focus group on a small set of customers before bringing a new product to market. More generally, it

[1] The seller's ability to commit may arise from, e.g. legal regulations or the seller's desire to protect their reputation.

may be possible for the sender to conduct market research about the receiver before trying to persuade them.

Motivated by such settings, we consider an idealized model in which the sender in a Bayesian persuasion game has access to an *oracle* with information about the receiver. The sender can *query* the oracle, and each query reveals a piece of information about the receiver, possibly at some *cost* to the sender. Given a set of allowable queries, we are interested in (1) analyzing how querying the oracle can help the sender design more persuasive signaling policies and (2) determining the sender's optimal *querying policy* when given a *budget* on the number of queries allowed.

1.1 Overview of Our Results

We discuss related work in Section 1.2, including connections to online learning for Bayesian persuasion, learning in Stackelberg games, and the simulation of economic agents using LLMs. In Section 2, we overview our setting and provide background on Bayesian persuasion. In Section 3.1, we characterize the form of the sender's optimal signaling policy whenever the sender is uncertain about the receiver's beliefs. We explain how a natural type of oracle access which we call *simulator access* can help the sender achieve higher expected utility in Section 3.2. In particular, borrowing language from the field of convex optimization, we show that making a simulation query is equivalent to specifying a subset of "receiver belief space" to which the sender has *separation oracle* access. If the receiver's true beliefs fall within the specified subset, the separation oracle says so. Otherwise, it returns a separating hyperplane in belief space. Using this characterization, we show how it is possible to use simulation queries to decrease the sender's uncertainty about the receiver's beliefs, which leads to higher sender utility. In Section 4 we shift our focus to computing *adaptive* querying policies, where the sender's i-th query to the oracle can depend on the outcomes of all previous $i-1$ queries. We show that the space of adaptive querying policies can be characterized by an extensive-form game between the sender and nature, and that the optimal adaptive querying policy may be solved for via backward induction on the game tree. We consider *non-adaptive* querying policies (i.e. querying policies which must make all queries *up front*) in Section 5. While we show that the general problem of computing the sender's optimal non-adaptive querying policy is NP-Complete via a reduction from set cover, the problem may be solved efficiently when the sender is restricted to making simulation queries in a canonical setting with binary states and actions. Motivated by these results, we study the effects of querying on the sender's expected utility in the non-adaptive setting. In Section 5.2 we show that the sender's expected utility is a submodular function of the queries made by the sender when there are binary states and actions. This implies that under this binary setting, there are diminishing returns to additional queries. In contrast, we show that this property does *not* hold in the general setting. Using this result, we show that the *greedy* policy, a popular heuristic often used to obtain approximately optimal solutions, is *not* approximately optimal when learning to persuade from oracle queries. All of our positive results for both adaptive and non-adaptive querying policies extend to the setting where the sender incurs a *cost* for making a query. In Section 6, we explore the setting in which the sender can *commit* to a querying policy (in addition to a signaling policy). This additional commitment power never hurts the sender, and we show that it can sometimes help them significantly. Finally, we conclude and highlight directions for future research in Section 7.

1.2 Related Work

Bayesian Persuasion Bayesian persuasion (BP) was first introduced by Kamenica and Gentzkow [26]; see Kamenica [25] for a recent survey of work in the area. Within BP, perhaps the most relevant line of work to ours is that of robust Bayesian persuasion, which aims to relax the assumptions on the information the sender has about the receiver. Work on robust

BP includes Dworczak and Pavan [19] and Hu and Weng [24], who study settings in which the receiver has an exogoneous source of information which they may learn from and that the sender is uncertain about. Parakhonyak and Sobolev [37] study a setting where the sender is uncertain about the true distribution over states, the receiver's beliefs, and the receiver's utility function. Kosterina [30] also study a setting in which the receiver's prior belief is unknown to the sender. Finally, Zu et al. [45] study an online setting in which the sender interacts with a *sequence* of receivers. Under their setting, neither the sender nor the receivers know the distribution of the payoff-relevant state. Our work is distinct from this line of work on robust BP, as their focus is typically on characterizing the "minimax" signaling policy (i.e. the signaling policy which is best in the worst case over the sender's uncertainty), while our focus is on using oracle queries to help the sender overcome their uncertainty.

Another relevant line of work in the Bayesian persuasion literature is that of sequential decision-making [21, 43, 3]. However, the focus here is usually on sequential interactions in which the payoff-relevant state may evolve, while in our setting we focus on repeated interactions between the sender and an oracle. Other relevant work includes topics at the intersection of BP and mechanism design [13, 10, 33], noisy BP with private information [29], BP with costly precision [17], and BP in the digital age [35]. Finally, our setup is similar to that of *online Bayesian persuasion* [11, 12, 4], where the sender interacts with a sequence of receivers. In this line of work, it is usually assumed that the sender faces an *adversarially* chosen sequence of receivers, whereas in our setting the sender has a prior over the type of receiver they are interacting with. Additionally, the goal of the sender in online BP is to minimize some notion of regret, while our goal is to use the oracle queries to learn the optimal signaling policy. As a result our work is also conceptually similar to the line of work on *pure exploration* in various bandit settings, where the first K rounds of a bandit problem are used to explore the arms [8, 14, 44, 9].

Large Language Models as Economic Agents There has been recent interest from the computer science and economics communities on using LLMs as simulated economic agents. Horton [23] show that LLMs can be used to simulate human behavior under different endowments, information, and preferences. Peters and Matz [40] show that LLMs can infer personality traits from users' Facebook status updates, and Brand et al. [6] observe that they answer survey questions in ways that are consistent with economic theory and well-documented patterns of consumer behavior. Fish et al. [20] propose *generative social choice*, a framework which combines social choice theory with LLMs' ability to generate unforeseen alternatives and extrapolate preferences. They model the LLM as an oracle which can precisely answer certain types of queries, and support this modeling assumption by empirically demonstrating that LLMs often provide accurate answers to the queries they ask. Matz et al. [36] show that personalized messages generated by LLMs are significantly more persuasive than non-personalized messages. Duetting et al. [18] study how to design auction mechanisms to combine outputs from different LLMs in an incentive-compatible way. [34, 1, 7, 15, 22, 41] investigate the performance of various LLMs in strategic decision-making settings. Park et al. [38] design "generative agents" capable of simulating realistic human day-to-day behavior. Kim and Lee [27] use LLMs to predict opinions for nationally representative surveys.

Simulation in Games Kovarik et al. [31] study a normal-form game setting in which one player can simulate the behavior of the other. They find that introducing simulation makes computing equilibrium easier and can increase trust between agents. In contrast, we study simulation in Bayesian persuasion games, which are a type of *Stackelberg* game [42, 16]. As a result, while our motivations are similar, our findings and analysis differ significantly from theirs. There is a line of work on learning the optimal strategy to commit to in Stackelberg games from query access [32, 39, 5, 2]. However, the type of Stackelberg game considered in this line of work is different from ours. In this setting, the leader's action (the leader is analogous to the sender in our setting) is to specify a mixed strategy over a finite set of actions. In contrast, in our setting the sender commits to a *signaling policy* which specifies a probability distribution over actions for every possible state realization.

2 Setting and Background

We first overview the traditional BP setting (in which the sender has full information about the receiver), before introducing the relaxed setting we consider in Section 2.2.

2.1 Background on Bayesian Persuasion

We denote the set of possible world states by Ω and the set of receiver actions by $\mathcal{A}$, where $d := |\Omega| < \infty$ and $A := |\mathcal{A}| < \infty$. In the classic BP setup, it is assumed that state ω is drawn from some distribution $\mathbf{p}$ over Ω which is known to both the sender and receiver. Sender and receiver utilities are given by utility functions $u_S : \Omega \times \mathcal{A} \to \mathbb{R}$ and $u_R : \Omega \times \mathcal{A} \to \mathbb{R}$ respectively. Both u_S and u_R are known to the sender.

Definition 2.1 (Signaling Policy)**.** *The sender's* signaling policy $\sigma : \Omega \to \mathcal{M}$ *is a (randomized) mapping from states to* messages *in some message class* $\mathcal{M}$ *such that* $M := |\mathcal{M}| < \infty$*.*

We use $\mathcal{M} \ni m \sim \sigma(\omega)$ to denote a message sampled from σ when the state is $\omega \in \Omega$. When the sender and receiver share a common prior $\mathbf{p}$ it is without loss of generality to signal according to the set of possible actions (i.e. $\mathcal{M} = \mathcal{A}$) via a revelation principle-style argument. (See Kamenica and Gentzkow [26] for more details.) Under this setting, a message has the interpretation of being the action that the sender is recommending the receiver to take. We say that the sender's signaling policy is *Bayesian incentive-compatible* if it is always in the receiver's best interest to follow the sender's action recommendations.

Definition 2.2 (Bayesian Incentive-Compatibility)**.** *A signaling policy* $\sigma : \Omega \to \mathcal{A}$ *is Bayesian incentive-compatible if for every action* $a \in \mathcal{A}$*,*

$$\mathbb{E}_{\omega\sim\mathbf{p}}[u_R(\omega, a)|m = a] \geq \mathbb{E}_{\omega\sim\mathbf{p}}[u_R(\omega, a')|m = a] \quad \textit{for all} \quad a' \in \mathcal{A}.$$

We assume that if the receiver is indifferent between two actions, they break ties in favor of the sender's recommended action.

The sender's optimal signaling policy is given by the solution to the following linear program:

$$\begin{aligned} &\max_{\sigma} \mathbb{E}_{\omega\sim\mathbf{p}, m\sim\sigma(\omega)}[u_S(\omega, m)] \\ &\text{s.t. } \mathbb{E}_{\omega\sim\mathbf{p}}[u_R(\omega, m)|m] \geq \mathbb{E}_{\omega\sim\mathbf{p}}[u_R(\omega, m)|m], \forall a \in \mathcal{A}, m \in \mathcal{M} \end{aligned}$$

The time it takes to compute the optimal signaling policy is $\mathcal{O}(\mathrm{LP}(dA, A^2 + dA))$, where $\mathrm{LP}(x, y)$ is the time it takes to solve a linear program with x variables and y constraints. Some of our results are specialized to the following *binary* setting, which is itself a popular model of persuasion (see, e.g. [37, 30, 24, 28] for various versions of this setting studied in the literature).

Definition 2.3 (Binary Setting)**.** *In the binary setting,* $\omega \in \{0, 1\}$*,* $a \in \{0, 1\}$*, and the prior over states can be described by a single number* $p := \mathbb{P}(\omega = 1)$*. Furthermore, we assume the following utility functions for the sender and receiver:* $u_S(\omega, a) = \mathbb{1}\{\omega = 1\}$*,* $u_R(\omega, a) = \mathbb{1}\{\omega = a\}$*. In other words, the sender always wants to incentivize the receiver to take action* $a = 1$*, but the receiver only wants to take action* $a = 1$ *whenever* $\omega = 1$*.*

Protocol: Bayesian persuasion with Oracle Queries

1. Sender makes K oracle queries, either adaptively (Definition 2.6) or non-adaptively (Definition 2.7).
2. Sender *commits* to signaling policy $\sigma : \Omega \to \mathcal{M}$.
3. Sender observes state $\omega \sim \mathbf{p}_{\tau^*}$, where $\tau^* \sim \mathcal{P}$.
4. Sender sends signal $m \sim \sigma(\omega)$, receiver forms posterior $\mathbf{p}_{\tau^*}|m$.
5. Receiver takes action $\arg\max_{a \in \mathcal{A}} \mathbb{E}_{\omega \sim \mathbf{p}_{\tau^*}}[u_R(\omega, a)|m]$.

Figure 1: Summary of our setting.

2.2 Our Model

We are now ready to introduce our model in full generality. We consider a setting in which there are a finite set of receiver *types* $\mathcal{T}$ ($T := |\mathcal{T}| < \infty$), where each type $\tau \in \mathcal{T}$ has prior $\mathbf{p}_\tau$ over Ω. We denote the *true* receiver type by τ^*, and assume that $\omega \sim \mathbf{p}_{\tau^*}$ (i.e. the receiver is *informed* about the state). Finally, we denote the sender's prior over receiver types by $\mathcal{P}(\mathcal{T})$, and assume that $\tau^* \sim \mathcal{P}$. In other words, the sender has a *prior* over the receiver's type, but does not know it exactly. One may view $\mathcal{P}(\mathcal{T})$ as a "second order prior" which captures the sender's information about the receiver population in aggregate. In Appendix A we describe how to extend our results to the setting in which the receiver is *misinformed* about the state.

We consider a sender with access to an *oracle* which can provide information about the receiver's type. In particular, we suppose that the sender can ask the oracle questions from a set of allowable *queries* $\mathcal{Q}$, where $Q := |\mathcal{Q}| < \infty$. In order to capture various settings in which the sender has different types of oracle access, we abstract away from the specific form of the query and instead define queries through the information they provide about the receiver's type.

Definition 2.4 (Query)**.** *A query $q \in \mathcal{Q}$ partitions the set of possible receiver types into a set of n_q non-overlapping subsets $\{s_{q,i}\}_{i=1}^{n_q}$ such that $s_{q,i} \subseteq \mathcal{T}$ $\forall i \in [n_q]$ and $\cup_{i=1}^{n_q} s_{q,i} = \mathcal{T}$. With a slight abuse of notation, we denote the subset that receiver type τ belongs to as $s_{q,\tau} \subseteq \mathcal{T}$. The oracle's response to query q is the subset s_{q,τ^*}.*

Let $\mathcal{S}$ denote the power set of $\mathcal{T}$, and observe that $s_{q,i} \in \mathcal{S}$ for any query $q \in \mathcal{Q}$ and $i \in [n_q]$. After making a query $q \in \mathcal{Q}$, the sender can leverage the fact that the true receiver type is contained in s_{q,τ^*} to design more persuasive signaling policies. An important special case is when the sender can *simulate* the response of the receiver, as is the case when, e.g., the receiver is a software agent or the sender can use a machine learning model to predict receiver behavior.

Definition 2.5 (Simulation Query)**.** *A simulation query is a tuple $q := (\sigma_q, m_q)$, where $\sigma_q : \Omega \to \mathcal{A}$ is a signaling policy mapping states to actions, and $m_q \in \mathcal{A}$ is an action recommendation. The oracle's response to a simulation query q is the receiver's best-response $a_{q,\tau^*} = \arg\max_{a \in \mathcal{A}}$ $\mathbb{E}_{\omega \sim \mathbf{p}_{\tau^*}}[u_R(\omega, a)|m_q]$.*

We explore the power of simulation queries in Section 3.2. In Section 4 and Section 5, we consider settings in which the sender can make multiple queries to the oracle. We primarily consider the setting in which the sender can make $K > 0$ costless oracle queries, although we also extend our results to the setting in which the sender can make unlimited queries, but each query has an associated cost. As we will see, the order of the interaction matters when the sender can make multiple queries. Two natural ways in which the sender can interact with the oracle are via *adaptive* and *non-adaptive* querying policies.

Definition 2.6 (Adaptive Querying Policy). *An adaptive querying policy $\pi : \mathcal{S} \times \mathbb{N} \rightarrow \mathcal{Q}$ is a mapping from information subsets and number of queries left to queries.*

When the sender is querying according to an adaptive querying policy, they make a query, observe the realized receiver partition, and then repeat until they have no queries remaining. In contrast when using a non-adaptive querying policy, the sender must specify all K queries *up front* before observing the oracle's response to any queries.

Definition 2.7 (Non-Adaptive Querying Policy). *A non-adaptive querying policy $\pi : \mathbb{N} \rightarrow \mathcal{Q}^*$ is a mapping from a number of queries to a set of queries.*

See Figure 1 for a summary of the model we consider. At a high level, the sender's goal is to compute a querying policy (either adaptive or non-adaptive, depending on the setting) which enables them to design a maximally persuasive signaling policy for the true receiver type.

3 How Should We Think About Querying?

We begin by overviewing several structural results which will be useful in later results and may be of independent interest. In Section 3.1, we derive the sender's optimal signaling policy whenever they have uncertainty about the receiver's beliefs. In Section 3.2, we derive geometric properties of simulation queries (Definition 2.5) which will be useful for obtaining algorithms in later sections.

3.1 Optimal Persuasion Under Uncertainty

How should the sender signal whenever they are uncertain about the receiver's type? Recall that when the receiver's prior is known, the sender can signal according to the set of receiver actions (i.e. $\mathcal{M} = \mathcal{A}$). The same cannot be said in our setting, so we need to derive an analogous "revelation principle" when the sender has uncertainty about the receiver. It turns out that it suffices to consider A^T messages in the general setting and $T+1$ messages in the binary setting (Definition 2.3).

Theorem 3.1 (Revelation Principle for Unknown Receiver Types). *It always suffices for the sender to signal using a message space $\mathcal{M}$ of size $M = A^T$. In the binary setting, it suffices to signal using $M = T + 1$ messages.*

Proof. Each message sent from a given signaling policy induces a posterior over states for each receiver type. Therefore, one may equivalently tell each receiver type which action they would take under their induced posterior. This takes A^T messages in the general setting.

In the binary setting, a receiver with prior p takes action $a = 1$ after seeing message m if and only if $p \geq \frac{\sigma(m|\alpha=0)}{\sigma(m|\alpha=1)+\sigma(m|\alpha=0)}$. Therefore if a receiver with prior p takes action 1, after seeing a message m, any receiver with prior $p' \geq p$ will also take action $a = 1$. Similarly, a receiver with prior p will take action $a = 0$ after seeing message m if and only if $p < \frac{\sigma(m|\alpha=0)}{\sigma(m|\alpha=1)+\sigma(m|\alpha=0)}$, which implies that a receiver with prior $p' \leq p$ will also take action $a = 0$. □

We can now characterize the sender's optimal signaling policy whenever they have uncertainty about the receiver's type, both in the general setting (Corollary 3.2) and in the binary setting (Corollary 3.3). The optimal signaling policy in the general setting follows readily from the above revelation principle for unknown receiver types.

Corollary 3.2 (Optimal Signaling Policy). *For a given set of receivers $\mathcal{T}'$ and prior over receiver types $\mathcal{P}(\mathcal{T}')$, the sender's optimal signaling policy is given by the solution to the following linear program:*

$$\sigma^*_{\mathcal{P}(\mathcal{T}')} := \arg\max_{\sigma} \ \mathbb{E}_{\tau \sim \mathcal{P}(\mathcal{T}')}[\mathbb{E}_{\omega \sim \mathbf{p}_\tau}[\mathbb{E}_{m \sim \sigma(\omega)}[u_S(\omega, m[\tau])]]]$$
$$s.t. \ \mathbb{E}_{\omega \sim \mathbf{p}_\tau}[u_R(\omega, m[\tau])|m] \geq \mathbb{E}_{\omega \sim \mathbf{p}_\tau}[u_R(\omega, m[\tau])|m], \forall a \in \mathcal{A}, \tau \in \mathcal{T}', m \in \mathcal{M}$$

Here $m[\tau]$ denotes the τ-th component of the message vector $m \in \mathcal{A}^{T'}$, where $T' := |\mathcal{T}'|$. The time it takes to compute the solution to this linear program is $\mathcal{O}(\text{LP}(dA^{T'}, T'A^{T'+1} + dA^{T'}))$.

The form of the sender's optimal signaling policy may be further simplified in the binary setting. In order to simplify the exposition, we assume that $p_\tau \leq \frac{1}{2}$ for every receiver type $\tau \in \mathcal{T}$.[2]

Corollary 3.3 (Optimal Signaling Policy; Binary Setting)**.** *In the binary setting, for a given set of receivers $\mathcal{T}' := \{\tau_L, \tau_{L+1}, \ldots, \tau_{H-1}, \tau_H\}$ where $p_L > p_{L-1} > \ldots > p_{H-1} > p_H$, the sender's optimal signaling policy sets $\sigma(m = m_{i^*}|\omega = 1) = 1$, $\sigma(m = m_{i^*}|\omega = 0) = \frac{p_{i^*}}{1-p_{i^*}}$, and $\sigma(m = 0|\omega = 0) = 1 - \frac{p_{i^*}}{1-p_{i^*}}$ where*

$$i^* := \arg \max_{i' \in [L,H]} \sum_{i=L}^{i'} \mathbb{P}_{\mathcal{P}(\mathcal{T}')}(\tau_i) \cdot \left(p_i + (1-p_i) \cdot \frac{p_{i'}}{1-p_{i'}} \right).$$

The time it takes to compute i^* is $\mathcal{O}((T')^2)$.

Proof. The sender's optimization in Corollary 3.2 may be written as

$$\begin{aligned}
\sigma^*_{\mathcal{P}(\mathcal{T}')} := \arg\max_{\sigma} \sum_{\tau \in \mathcal{T}'} \mathbb{P}_{\mathcal{P}(\mathcal{T}')}(\tau) \cdot \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma(\omega)}[u_S(\omega, m)] \\
\text{s.t. } \forall \tau \in \mathcal{T}', a \in \mathcal{A}, m \in \mathcal{M} \;\; \mathbb{E}_{\omega \sim \mathbf{p}_\tau}[u_R(\omega, m)|m] \geq \mathbb{E}_{\omega \sim \mathbf{p}_\tau}[u_R(\omega, a)|m] \\
\sum_{m \in \mathcal{M}} \sigma(m_| \omega) = 1, \;\; \forall m \in \mathcal{M}.
\end{aligned}$$

Using the fact that there are two states and actions, and that it is without loss of generality to signal using at most $T' + 1$ messages, the sender's optimization may be rewritten as

$$\begin{aligned}
\max_{\sigma} \sum_{i=L}^{H} \mathbb{P}_{\mathcal{P}(\mathcal{T}')}(\tau_i) \cdot \sum_{j=L}^{i} p_i \cdot \sigma(m = m_j|\omega = 1) + (1-p_i) \cdot \sigma(m = m_j|\omega = 0) \\
\text{s.t. } \forall i \in [L, H], \;\; \sigma(m = m_i|\omega = 0) \leq \frac{p_i}{1-p_i} \cdot \sigma(m = m_i|\omega = 1) \\
\sum_{i=L}^{H} \sigma(m = m_i|\omega = 1) \leq 1
\end{aligned}$$

We obtain the desired result by observing that due to the geometry of the linear program constraints, it suffices to pick some $i^* \in [L, H]$ and use message m_{i^*} without loss of generality. □

The optimal signaling policy in the binary setting has the following interpretation: when receiving message $m_{i'}$, all receivers of type τ_i such that $i \leq i'$ should take action $a = 1$.

3.2 How Simulation Helps

In this section we illustrate how the sender can benefit from querying, through the lens of simulation queries (Definition 2.5). By making a simulation query $q = (\sigma_q, m_q)$, the sender is specifying a convex polytope $\mathcal{R}_q \subseteq \Delta^d$ for which we have a *separation oracle*, a concept from optimization which can be used to describe a convex set. In particular, given a point $\mathbf{x} \in \mathbb{R}^d$, a separation oracle for a convex body $\mathcal{K} \subseteq \mathbb{R}^d$ will either (1) assert that $\mathbf{x} \in \mathcal{K}$ or (2) return a hyperplane $\boldsymbol{\theta} \in \mathbb{R}^d$ which separates $\mathbf{x}$ from $\mathcal{K}$, i.e. $\boldsymbol{\theta}$ is such that $\langle \boldsymbol{\theta}, \mathbf{y} \rangle > \langle \boldsymbol{\theta}, \mathbf{x} \rangle$ for all $\mathbf{y} \in \mathcal{K}$. Formally, we have the following equivalent characterization of a simulation query:

[2]This assumption is without loss of generality, since the sender can treat all receiver types with prior $\mathbb{P}(\omega = 1) \geq \frac{1}{2}$ as the same type without any loss in utility.

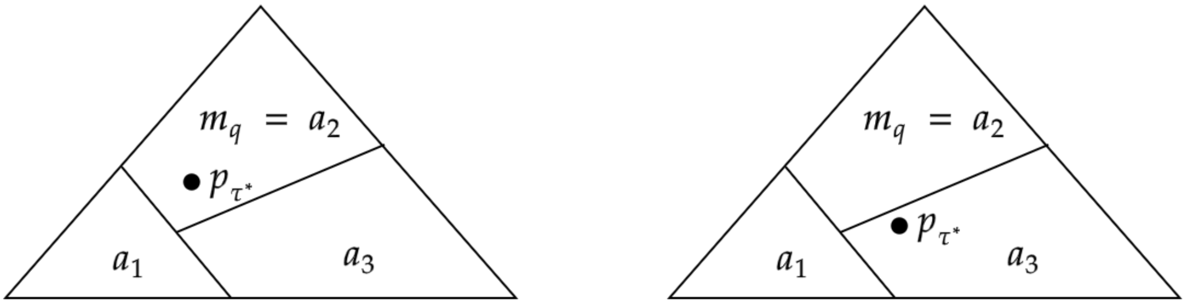

Figure 2: Visualization of receiver best-response regions of Δ^d for simulation query $q = (\sigma_q, m_q)$. Left: A setting in which the receiver follows the sender's recommendation. Under this scenario, we can infer that the receiver's prior $\mathbf{p}_{\tau^*}$ is contained in the best-response region $\mathcal{R}_q$. Right: A setting in which the receiver does not follow the sender's recommendation. Under this scenario, we can infer the region that the receiver's prior belongs to by their best-response (in this case, a_3).

Fact 3.4 (Relationship between simulation queries and separation oracles)**.** *By making a simulation query $q = (\sigma_q, m_q)$, the sender is specifying a polytope*

$$\mathcal{R}_q := \{\mathbf{p} \in \Delta^d \; : \; \sum_{\omega \in \Omega} (u_R(\omega, m_q) - u_R(\omega, a)) \cdot \sigma_q(m_q|\omega) \cdot \mathbf{p}[\omega] \geq 0, \forall a \in \mathcal{A}\}$$

and the oracle returns either

1. *$\mathbf{p}_{\tau^*} \in \mathcal{R}_q$ or*
2. *$\mathbf{p}_{\tau^*} \notin \mathcal{R}_q$ and for some $a' \in \mathcal{A}$ for all $\mathbf{p}' \in \mathcal{R}_q$,*

$$\sum_{\omega \in \Omega} (u_R(\omega, m_q) - u_R(\omega, a')) \cdot \sigma_q(m_q|\omega) \cdot \mathbf{p}'[\omega] > \sum_{\omega \in \Omega} (u_R(\omega, m_q) - u_R(\omega, a')) \cdot \sigma_q(m_q|\omega) \cdot \mathbf{p}_{\tau^*}[\omega].$$

Proof. By Definition 2.2, $\mathcal{R}_q$ is the set of all priors for which the receiver would follow the sender's recommendation m_q when they are signaling according to σ_q. Therefore if the oracle returns action $a_{q,\tau^*} = m_q$, it must be the case that $\mathbf{p}_{\tau^*} \in \mathcal{R}_q$. If $a_{q,\tau^*} \neq m_q$, we know that $\mathbf{p}_{\tau^*} \notin \mathcal{R}_q$ and we can infer that $\sum_{\omega \in \Omega} (u_R(\omega, m_q) - u_R(\omega, a_{q,\tau^*})) \cdot \sigma_q(m_q|\omega) \cdot \mathbf{p}_{\tau^*}[\omega] < 0$, which implies (2). □

An immediate corollary of Fact 3.4 is that in the binary setting, there is a one-to-one correspondence between simulation queries and thresholds in $(0, \frac{1}{2})$. By applying Definition 2.2, we can obtain the following closed-form relationship between the two.

Corollary 3.5. *Given a simulation query $q = (\sigma_q, m_q)$ in the binary setting, $a_{q,\tau^*} = m_q$ if and only if*

$$p_{\tau^*}[m_q] \geq \frac{\sigma_q(m_q|\omega \neq m_q)}{\sigma_q(m_q|\omega = m_q) + \sigma_q(m_q|\omega \neq m_q)}$$

where $p_{\tau^*}[m]$ denotes the prior probability that receiver type τ^* places on state m (i.e. $p_{\tau^*}[m] = p_{\tau^*}$ if $m = 1$ and $p_{\tau^*}[m] = 1 - p_{\tau^*}$ if $m = 0$).

Corollary 3.5 implies that there always exists a simulation query q which can distinguish between any two receiver types τ, τ' such that $p_\tau \neq p_{\tau'}$ in the binary setting. Similar intuition carries over to the general setting, although the form of the receiver's utility function may prevent us from being able to distinguish between types through simulation queries. We capture this intuition through the notion of *separable* types.

Algorithm 1 Computing a query to separate two types

Require: Priors $\mathbf{p}_1$, $\mathbf{p}_2$

for all pairs of actions $(m, a') \in \mathcal{A} \times \mathcal{A}$ **do**

Solve the following linear program:

$$
\begin{aligned}
&\min_{\{\sigma(m|\omega)\}_{\omega\in\Omega}} \eta \\
&\sum_{\omega\in\Omega}(u_R(\omega,m)-u_R(\omega,a))\cdot\sigma(m|\omega)\cdot\mathbf{p}_1[\omega]\geq 0,\ \forall a\in\mathcal{A} \\
&\sum_{\omega\in\Omega}(u_R(\omega,m)-u_R(\omega,a'))\cdot\sigma(m|\omega)\cdot\mathbf{p}_2[\omega]\leq \eta \\
&0\leq\sigma(m|\omega)\leq 1,\ \forall\omega\in\Omega
\end{aligned}
\tag{1}
$$

if $\eta < 0$ **then**

return query (σ', m), where σ' is a signaling policy which maximizes Optimization 1.

end if

end for

return `Fail`

Definition 3.6 (Separability). *Two receiver types τ, τ' are* separable *if there exists a simulation query $q = (\sigma_q, m_q)$ such that*

$$
\begin{aligned}
&\sum_{\omega\in\Omega}(u_R(\omega,m_q)-u_R(\omega,a))\cdot\sigma_q(m_q|\omega)\cdot\mathbf{p}_\tau[\omega]\geq 0\ \ \forall a\in\mathcal{A},\ \textit{and} \\
&\sum_{\omega\in\Omega}(u_R(\omega,m_q)-u_R(\omega,a'))\cdot\sigma_q(m_q|\omega)\cdot\mathbf{p}_{\tau'}[\omega]< 0\ \ \textit{for some } a'\in\mathcal{A}.
\end{aligned}
$$

In other words, a simulation query q separates τ and τ' if type τ follows action recommendation m_q when the sender is signaling according to signaling policy σ_q, but receiver τ' does not. Algorithm 1 finds a simulation query which separates two types when such a query exists by enumerating all pairs of actions $a, a' \in \mathcal{A}$ and checking if there exists a valid signaling policy for which one type takes action a and the other takes action a' when $m = a$.

Theorem 3.7. *For any τ, τ' which are separable (according to Definition 3.6), Algorithm 1 returns a simulation query which separates them in time $\mathcal{O}(A^2 \cdot \mathrm{LP}(d, d + A))$.*

See Appendix B.1 for the proof of Theorem 3.7. While Theorem 3.7 characterizes when it is possible to distinguish between two or more receiver types in the general setting, it may be possible to distinguish between three or more types using a *single* simulation query (unlike in the binary setting). In fact, the following example shows how in some settings it is possible to distinguish between up to d different types using a single simulation query.

Example 3.8. *Suppose that there are d states and d actions, where $u_R(\omega, a) = \mathbb{1}\{\omega = a\}$. Consider d types with priors $\mathbf{p}_1, \ldots, \mathbf{p}_d$ and let $\mathbf{p}_i[\omega_i] = \frac{2}{d+1}$, $\forall i \in [d]$ and $\mathbf{p}_i[\omega_j] = \frac{1}{d+1}$ for $j \neq i$. Under this setting, receiver type i will take action a_i when $m = a_1$ if for all $j \neq 1$, $\sigma(m = a_1|\omega_j) = 2\sigma(m = a_1|\omega_1)$.*

In Appendix B.2, we describe an extension of Algorithm 1 for identifying a simulation query which separates any $n \leq d$ types, if such a query exists. In contrast to "query identification", our goal in Section 4 and Section 5 is to determine the optimal set of queries to make in order to design maximally persuasive signaling policies. As such, we assume that the set of valid queries $\mathcal{Q}$ has already been pre-determined and is known to the sender in the sequel.

Algorithm 2 Computing the Optimal Adaptive Querying Policy: K Queries

Require: Query budget $K \in \mathbb{N}$

- Set $V[\mathcal{T}', 0] := \max_{\sigma^*_{\mathcal{P}(\mathcal{T}')}} \mathbb{E}_{\tau \sim \mathcal{P}(\mathcal{T}')} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma^*_{\mathcal{P}(\mathcal{T}')}(\omega)} u_S(\omega, m)$ for all feasible information partitions $\mathcal{T}' \in \mathcal{F}$, where $\sigma^*_{\mathcal{P}(\mathcal{T}')}$ is the optimal BIC signaling policy as defined in Corollary 3.2.
- For every $1 \leq k \leq K$ and all $\mathcal{T}' \in \mathcal{F}$, compute $V[\mathcal{T}', k] := \max_{q \in \mathcal{Q}} V[\mathcal{T}', k, q]$, where $V[\mathcal{T}', k, q] := \sum_{i=1}^{n_q} V[\mathcal{T}' \bigcap s_{q,i}, k-1] \cdot \mathbb{P}_{\mathcal{P}}(\tau^* \in \mathcal{T}' \bigcap s_{q,i} | \tau^* \in \mathcal{T}')$.
- The optimal adaptive querying policy then makes query $\pi^*(\mathcal{T}'|k) = \arg\max_{q \in \mathcal{Q}} V[\mathcal{T}', k, q]$ when in information partition $\mathcal{T}'$ with k queries left.

4 Adaptive Querying Policies

We now shift our focus to computing the optimal querying policy when the sender can adaptively select K queries from a set of possible queries $\mathcal{Q}$ (as defined in Definition 2.4). Specifically, we are interested in adaptively selecting K queries in such a way that maximizes the sender's utility in expectation when they signal according to the optimal signaling policy for the realized information partition.

Definition 4.1 (Feasible Information Partition)**.** *We say that a subset of receiver types $\mathcal{T}' \subseteq \mathcal{T}$ is a* feasible information partition *if for some query budget K, there exists some $\tau \in \mathcal{T}$ and $\mathcal{Q}' \subseteq \mathcal{Q}$ with $|\mathcal{Q}'| \leq K$ such that $\mathcal{T}' = \bigcap_{q \in \mathcal{Q}'} s_{q,\tau}$. We denote the set of feasible partitions by $\mathcal{F} := \{\mathcal{T}' : \mathcal{T}' = \cap_{q \in \mathcal{Q}'} s_{q,\tau}$ for some $\tau \in \mathcal{T}$ and $\mathcal{Q}' \subseteq \mathcal{Q}\}$ and the number of feasible partitions by $F := |\mathcal{F}| \leq 2^K$.*

An adaptive querying policy needs to specify a query to make for each of the F feasible information partitions. Observe that the sender's optimal query for a particular partition $\mathcal{T}'$ may depend on any future queries which are made, but it does not depend on the queries which led to the sender being in partition $\mathcal{T}'$. Motivated by this observation, we show that the sender's optimal querying policy may be computed via backward induction on an appropriately-constructed extensive-form game tree. In particular, we start by computing the expected utility associated with the optimal signaling policy for each information partition whenever the sender has no queries remaining (recall Corollary 3.2). Using these expected utilities as a building block, we can compute the sender's optimal querying policy when they have only one query. We then use our solution for the one query setting to construct the optimal querying policy whenever the sender has two queries, and we continue iterating until we have the optimal querying policy for K queries. This process is illustrated in Figure 3 and formally described in Algorithm 2.

Theorem 4.2. *Algorithm 2 computes the sender's optimal adaptive querying policy for K queries in $\mathcal{O}(FKQT)$ time.*

Proof. **Correctness** Observe that via the principle of deferred decisions, we can characterize the set of the sender's possible querying policies as an alternating move, extensive-form game between the sender and nature, in which each sender action note is characterized by an information partition $\mathcal{T}'$. In each round, the sender first selects a query $q \in \mathcal{Q}$, then nature reveals a finer information partition $\mathcal{T} \bigcap s_{q,i}$ with probability $\mathcal{P}(\tau^* \in \mathcal{T} \bigcap s_{q,i} | \tau^* \in \mathcal{T})$ for all $i \in [n_q]$. The optimal querying policy may be solved for by backward induction up the game tree (pictured in Figure 3).

We proceed via induction on the number of queries K. The base case when $K = 0$ is trivially optimal. Assume that Algorithm 2 produces the correct solution for $K = \kappa - 1$ queries. By the inductive argument, we know that $V[\mathcal{T}' \bigcap s_{q,i}, \kappa - 1]$ is the optimal value attainable when in information partition $\mathcal{T}' \bigcap s_{q,i} \subseteq \mathcal{T}'$ with $\kappa - 1$ remaining queries, and so the sender should pick their first query in order to maximize their utility in expectation, subject to querying optimally

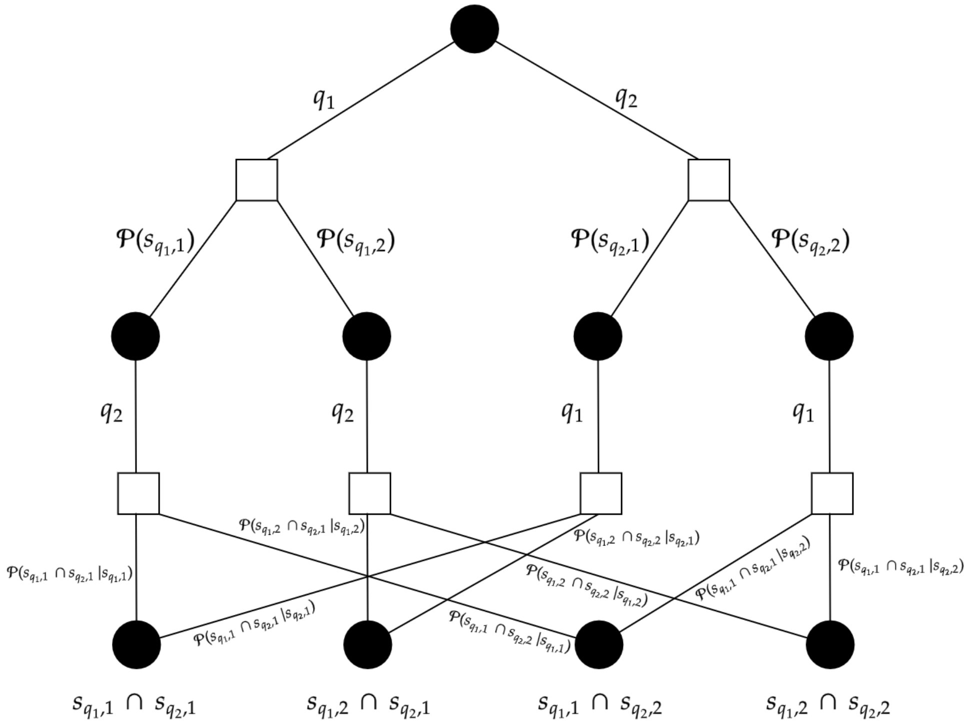

Figure 3: Game tree between sender and nature for a setting with two queries q_1 and q_2. Each query results in two partitions, with query q_1 (resp. q_2) resulting in partitions $s_{q_1,1}$ and $s_{q_1,2}$ (resp. $s_{q_2,1}$ and $s_{q_2,2}$). Sender actions (black circles) are queries to the oracle, and nature actions (white squares) are (refined) partitions of receiver types. No matter what set of types $\mathcal{T}' \in \mathcal{F}$ is reached, the probability of reaching $\mathcal{T}'$ is exactly equal to $\mathcal{P}(\mathcal{T}')$ by construction.

for the remaining $\kappa - 1$ queries. This is precisely the definition of $V[\mathcal{T}', \kappa]$, and so Algorithm 2 produces the correct solution for $K = \kappa$ queries.

Runtime analysis Observe that each probability $\mathbb{P}_{\mathcal{P}}(\tau \in \mathcal{T}' \bigcap s_{q,i} | \tau \in \mathcal{T}')$ for $q \in Q$, $i \in [n_q]$, and $\mathcal{T}' \in \mathcal{F}$ may be pre-computed before the algorithm is run in time $\mathcal{O}(T)$, and that there are at most $\mathcal{O}(FK)$ such probabilities to pre-compute. Similarly each $V[\mathcal{T}', k]$ value may be computed in time $\mathcal{O}(TQ)$, and there are at most $\mathcal{O}(FK)$ such values to compute. □

In order to separate the time it takes to compute the optimal *querying* policy from any time required to compute the sender's *signaling* policy, we assume that the optimal signaling policies for each feasible information partition are known to the sender before Algorithm 2 is run. Alternatively, one could view the computation of the optimal signaling policies as part of Algorithm 2, in which case the runtime would contain an additional additive factor proportional to the time it takes to compute the optimal signaling policy under uncertainty (see Corollary 3.2). Algorithm 2 simplifies significantly in the binary setting with simulation queries. Under this setting, we overload notation and use q to index the receiver type with the smallest prior which takes action $a = 1$ in response to simulation query $q \in \mathcal{Q}$. Note that such a query is well-defined, by the results in Section 3.2.

Corollary 4.3. *In the binary setting with simulation queries, the update step of Algorithm 2 simplifies to computing* $V[(i,j),k] = \max_{q \in \{i,\dots,j\}} V[(i,j),k,q]$ *for all* $1 \le i < j \le T$*, where*

$$V[(i,j),k,q] := V[(i,q),k-1] \cdot \mathbb{P}_{\mathcal{P}}(p_{\tau^*} \le p_q | p_{\tau^*} \in \{p_i,\dots,p_j\}) \\ + V[(q+1,j),k-1] \cdot \mathbb{P}_{\mathcal{P}}(p_{\tau^*} > p_q | p_{\tau^*} \in \{p_i,\dots,p_j\})$$

Algorithm 3 Computing the Optimal Adaptive Querying Policy: Costly Queries

Require: Query costs $\{c_q\}_{q \in \mathcal{Q}}$

- Set $V[\mathcal{T}', 0] := \max_{\sigma^*_{\mathcal{P}(\mathcal{T}')}} \mathbb{E}_{\tau \sim \mathcal{P}(\mathcal{T}')} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma^*_{\mathcal{P}(\mathcal{T}')}(\omega)} u_S(\omega, m)$ for all feasible partitions $\mathcal{T}' \in \mathcal{F}$, where $\sigma^*_{\mathcal{P}(\mathcal{T}')}$ is the optimal BIC signaling policy as defined in Corollary 3.2.
- Let $V[\mathcal{T}', k, q_\varnothing] := V[\mathcal{T}', 0]$ for every $1 \leq k \leq T$ and all $\mathcal{T}' \in \mathcal{F}$.
- For every $1 \leq k \leq T$ and all $\mathcal{T}' \in \mathcal{F}$, compute $V[\mathcal{T}', k] := \max_{q \in \mathcal{Q}} V[\mathcal{T}', k, q]$, where $V[\mathcal{T}', k, q] := \sum_{i=1}^{n_q} V[\mathcal{T}' \bigcap s_{q,i}, k-1] \cdot \mathbb{P}_{\mathcal{P}}(\tau^* \in \mathcal{T}' \bigcap s_{q,i} | \tau^* \in \mathcal{T}') - c_q$ for all $q \neq q_\varnothing$.
- The optimal adaptive querying policy then makes query $\pi^*(\mathcal{T}'|k) = \arg\max_{q \in \mathcal{Q}} V[\mathcal{T}', k, q]$ when in subset $\mathcal{T}'$ with k queries left. If query $q_\varnothing$ is selected, the sender stops querying.

The runtime of Algorithm 2 in this setting is $\mathcal{O}(T^4 K)$.

Proof. Correctness follows from the proof of Theorem 4.2 and the observation that in the binary setting with simulation queries, the set of feasible information partitions simplifies to $\mathcal{F} = \{\{\tau_i, \ldots, \tau_j\} : 1 \leq i \leq j \leq T\}$. Note that in the binary setting with simulation queries, $Q = T - 1$ and each query splits the set of receiver types into at most two partitions, so the time to compute each $V[(i,j), k]$ is $\mathcal{O}(T)$. Finally, observe that under this setting $F = \mathcal{O}(T^2)$. □

4.1 Costly Queries

Consider a setting in which the sender has an unlimited query budget (i.e. $K = \infty$), but pays some fixed cost $c_q > 0$ for making query $q \in \mathcal{Q}$. The sender's goal is now to adaptively select queries in order to maximize their utility in expectation, *minus the costs associated with querying.* Our extension to the costly setting follows by (1) observing that the sender will make at most $\min\{T-1, Q\}$ queries without a query budget and (2) modifying Algorithm 2 to include the ability to opt out of making any more queries. For ease of analysis, it is useful to represent the sender's ability to opt out by a "dummy" query $q_\varnothing \in \mathcal{Q}$ such that $c_{q_\varnothing} = 0$ and $q_\varnothing$ has no effect on the current partition of receiver types. Our procedure for computing the sender's optimal adaptive querying policy in the costly query setting is given in Algorithm 3.

Theorem 4.4. *Algorithm 3 computes the sender's optimal adaptive querying policy (in the costly query setting) in* $\mathcal{O}(FQT \cdot \min\{T, Q\})$ *time.*

Proof. **Correctness**: If the sender *must* make $\min\{T-1, Q\}$ queries, then making query $\arg\max_{q \in \mathcal{Q} \setminus \{q_\varnothing\}} V[\mathcal{T}', k, q]$ is optimal for all $\mathcal{T}' \in \mathcal{F}$ and $1 \leq k \leq \min\{T-1, Q\}$ by an argument identical to the proof of Theorem 4.2. However, it is in the sender's best interest to stop querying if making an additional query leads to an expected decrease in utility once the cost of making an additional query is factored in. Hence, the optimal querying policy only continues querying if the expected gain in utility of making the additional query is non-negative.

Runtime analysis Observe that each $\mathbb{P}_{\mathcal{P}}(\tau \in \mathcal{T}' \bigcap s_{q,i} | \tau \in \mathcal{T}')$ for $q \in Q$, $i \in [n_q]$, and $\mathcal{T}' \subseteq \mathcal{T}$ may be pre-computed before the algorithm is run in time $\mathcal{O}(T)$, and that there are at most $\mathcal{O}(F \cdot \min\{T, Q\})$ such probabilities to pre-compute. Similarly each $V[\mathcal{T}', k]$ value may be computed in time $\mathcal{O}(TQ)$, and there are at most $\mathcal{O}(F \cdot \min\{T, Q\})$ such values to compute. □

In Appendix C.1, we show how Algorithm 3 simplifies in the binary setting with simulation queries in a result analogous to Corollary 4.3.

5 Non-Adaptive Querying Policies

When querying non-adaptively, the sender must make all K queries *up front* (recall Definition 2.7). Our results in this setting are mixed, and differ significantly between the binary setting and the gen-

Algorithm 4 Computing the Optimal Non-Adaptive Querying Policy: K Queries

Require: Query budget $K \in \mathbb{N}$

- Set $V[i,j,0] :=$

$$\max_{\sigma^*_{\mathcal{P}(\tau^* \in \{i,\ldots,j\})}} \mathbb{P}_{\mathcal{P}}(\tau^* \in \{q+1,\ldots,i\}) \cdot \mathbb{E}_{\tau \sim \mathcal{P}(\tau^* \in \{i,\ldots,j\})} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma^*_{\mathcal{P}(\tau^* \in \{i,\ldots,j\})}(\omega)} u_S(\omega, m)$$

  for all $1 \le i \le j \le T$, where $\sigma^*_{\mathcal{P}(\tau^* \in \{i,\ldots,j\})}$ is the optimal BIC signaling policy as defined in Corollary 3.2.
- For every $1 \le k \le K$ and $1 \le j \le T$ compute

$$V[1,j,k] := \max_{q \in \mathcal{Q}} V[q+1,j,0] + V[1,q,k-1]$$

- The optimal non-adaptive querying policy then makes the K queries which obtain value $V[1,T,K]$.

eral setting. The following notion of a *non-adaptive receiver partition* will be useful in the sequel.

Definition 5.1 (Non-Adaptive Receiver Partition)**.** *A set of non-adaptive queries $\mathcal{Q}' \subseteq \mathcal{Q}$ induces a partition over receiver types $H_{\mathcal{Q}'}$ such that $\bigcup_{\eta \in H_{\mathcal{Q}'}} \eta = \mathcal{T}$. Receiver type τ belongs to the partition $\eta_{\mathcal{Q}'}[\tau] \in H_{\mathcal{Q}'}$ such that*

$$\eta_{\mathcal{Q}'}[\tau] := \bigcap_{q \in \mathcal{Q}'} s_{q,\tau}.$$

The key difference between Definition 5.1 and the notion of a feasible information partition (Definition 4.1) from Section 4 is that in the non-adaptive setting, the sender does not get to see the outcome of a query until all queries are made.

5.1 Computing Non-Adaptive Querying Policies

We show that the problem of computing the sender's optimal non-adaptive querying policy may be computed efficiently via dynamic programming in the binary setting with simulation queries. However in the general setting, this problem is shown to be NP-Hard via a reduction to Set Cover.

We find that the optimal non-adaptive querying policy may be computed via dynamic programming when considering simulation queries in the binary setting. However unlike in the adaptive setting, the optimal querying policy is computed by leveraging the fact that there is a "total ordering" over receiver types when there are two states, and iteratively building solutions for larger sets of receivers. We again overload notation and use q to index the receiver type with the smallest prior which takes action $a = 1$ in response to simulation query $q \in \mathcal{Q}$. Given a set of T receiver types $\{1, \ldots, T\}$ where $p_1 > p_2 > \cdots > p_T$, our algorithm (Algorithm 4) keeps track of the optimal sender utility achievable with k queries when in receiver partition $\{1, \ldots, i\}$ for all $1 \le i \le T$. Due to the structure induced by non-adaptivity in this setting, we are able to write the sender's expected utility for $k+1$ queries in receiver partition $\{1, \ldots, i+1\}$ as a function of the optimal solution for k queries in partition $\{1, \ldots, i\}$.

Theorem 5.2. *In the binary setting with simulation queries, Algorithm 4 computes the sender's optimal non-adaptive querying policy in $\mathcal{O}(T^3 K)$ time.*

Costly Queries Algorithm 4 may be modified to compute the optimal querying policy in the binary setting with *costly* simulation queries. See Appendix D.2 for details.

The key idea behind Algorithm 4 is that in the binary setting with simulation queries, there is always a "total ordering" over both receiver types and queries, and thus one can use dynamic

Algorithm 5 Non-Adaptive Greedy Querying Policy

Require: Query budget $K \in \mathbb{N}$

Set $\mathcal{Q}_0 = \emptyset$

for $j = 0, 1, \ldots, K-1$ **do**

Add query

$$x_{j+1} = \arg\max_{q \in \mathcal{Q}} \mathbb{E}_{\tau \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\mathcal{P}(\eta_{\mathcal{Q}_j \cup q})[\tau]}(\omega)} [u_S(\omega, m)]$$

Set $\mathcal{Q}_{j+1} \leftarrow \mathcal{Q}_j \cup \{x_{j+1}\}$

end for

return $\mathcal{Q}_K$

programming in order to iteratively construct an optimal solution. This intuition does not carry over to the general setting, as no such total ordering exists. In fact as we will show, the corresponding decision problem is NP-Complete.

Definition 5.3 (Non-Adaptive Decision Problem)**.** *Given $(\mathcal{T}, \mathcal{P}, \mathcal{Q}, K, u)$ where $\mathcal{T}$ is a set of receiver types, $\mathcal{P}$ is a second-order prior over receiver types, $\mathcal{Q}$ is a set of allowable queries, $K \in \mathbb{N}$, and $u \in \mathbb{R}_+$, does there exist a collection of queries $\mathcal{Q}' \subseteq \mathcal{Q}$ of size $|\mathcal{Q}'| \leq K$ such that the sender's expected utility from signaling optimally after making queries $\mathcal{Q}'$ is at least u?*

Theorem 5.4. *The Non-Adaptive Decision Problem (Definition 5.3) is NP-Complete.*

Proof Sketch. A candidate solution may be checked in polynomial time, which implies that the problem is in NP. To prove NP-Hardness, we proceed via a reduction to Set Cover. Given a universe of elements U, a collection of subsets S, and a number K, the set cover decision problem asks if there exists a set of subsets $S' \subseteq S$ such that $|S'| \leq K$ and $\bigcup_{s \in S'} s = U$. Our reduction proceeds by creating a receiver type for every element in U and a query for every subset in S. We set u and $\{\mathbf{p}_\tau\}_{\tau \in \mathcal{T}}$ in such a way that the sender can only achieve expected utility u if they can distinguish between every receiver type using K non-adaptive queries, i.e. the receiver partition for type τ (Definition 5.1) is a singleton for all $\tau \in \mathcal{T}$. Finally, we show that under this construction the answer to the set cover decision problem is `yes` if and only if the answer to the corresponding Non-Adaptive decision problem is also `yes`. $\square$

5.2 Properties of the Sender's Utility

Motivated by the hardness result of Section 5.1, we examine how the sender's expected utility changes as a function of the queries they make. We begin by showing that the sender gets *decreasing marginal utility* for making additional simulation queries in the binary setting. As a corollary, this implies that the greedy querying policy is approximately optimal under this setting. In contrast, we show that decreasing marginal utility of queries does *not* hold in the general setting.

Theorem 5.5. *Consider two sets of queries $\widehat{\mathcal{Q}}$, $\tilde{\mathcal{Q}}$, where $\tilde{\mathcal{Q}} \subseteq \widehat{\mathcal{Q}} \subseteq \mathcal{Q}$, and any query $q \in \mathcal{Q}$. In the binary setting, we have that*

$$\begin{aligned}&\mathbb{E}_{\tau \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\mathcal{P}(\eta_{\tilde{\mathcal{Q}} \cap q})[\tau]}(\omega)} [u_S(\omega, m)] - \mathbb{E}_{\tau \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\mathcal{P}(\eta_{\tilde{\mathcal{Q}}})[\tau]}(\omega)} [u_S(\omega, m)] \\ \geq &\mathbb{E}_{\tau \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\mathcal{P}(\eta_{\widehat{\mathcal{Q}} \cap q})[\tau]}(\omega)} [u_S(\omega, m)] - \mathbb{E}_{\tau \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\mathcal{P}(\eta_{\widehat{\mathcal{Q}}})[\tau]}(\omega)} [u_S(\omega, m)].\end{aligned}$$

In other words, the sender's expected utility exhibits decreasing marginal returns from additional queries.

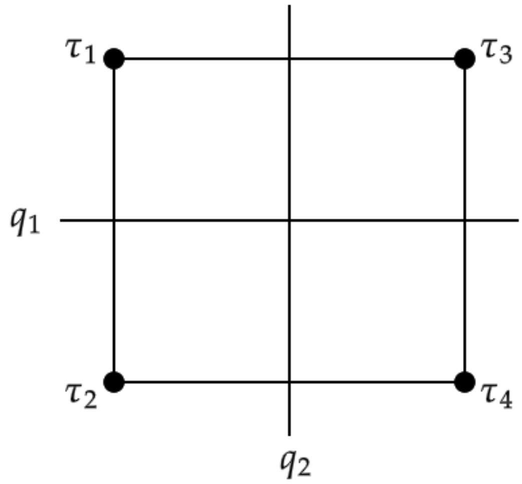

Figure 4: Illustration of Example 5.6 in two dimensions.

Proof Sketch. The proof proceeds via a case-by-case analysis on the difference in expected sender utility as a function of $\tilde{\mathcal{Q}}$ and $\widehat{\mathcal{Q}}$. The key idea is to leverage the form of the sender's optimal signaling policy under uncertainty in the binary setting (Corollary 3.3) to argue that no matter how the extra query partitions $\tilde{\mathcal{Q}}$ and $\widehat{\mathcal{Q}}$, the expected marginal gain is always weakly larger under $\tilde{\mathcal{Q}}$. $\square$

An immediate corollary of Theorem 5.5 is that the greedy non-adaptive querying policy (Algorithm 5) is $(1 - 1/e)$-approximately optimal in the binary setting. See Appendix D.5 for more details.

Costly Queries All of our results for the binary setting carry over to the non-adaptive variant of the costly query setting described in Section 4.1. Under this setting the sender may make unlimited non-adaptive queries, but pays a cost c_i for making query q_i. In particular, in Appendix D.6 we show that the sender's expected utility still exhibits decreasing marginal returns in the costly binary setting, and a natural modification of the non-adaptive greedy policy (Algorithm 7) is $(1 - 1/e)$-approximately optimal.

In contrast to the binary setting, the above properties do not extend to the sender's expected utility in the general setting. Consider the following example.

Example 5.6. *For $L \in \mathbb{N}$, consider a setting with state space $\Omega = \{\omega_0, \omega_1, \ldots, \omega_{2^L-1}, \omega_\varnothing\}$, receiver types $\mathcal{T} = \{\tau_0, \tau_1, \ldots \tau_{2^L-1}\}$ where receiver τ_i has prior $\mathbf{p}_i$ such that $\mathbf{p}_i[\omega_\varnothing] = \epsilon$ for some $\epsilon > 0$, $\mathbf{p}_i[\omega_j] = \frac{\epsilon}{2}$ for all $j \neq i$, which implies that $\mathbf{p}_i[\omega_i] = 1 - (\frac{1}{2} \cdot 2^L + 1) \cdot \epsilon$. Suppose that the set of receiver actions is equal to the set of states, i.e. $\mathcal{A} = \Omega$. Consider the following sender and receiver utility functions: $u_S(\omega, a) = \mathbb{1}\{a \neq \omega_\varnothing\}$ and*

$$u_R(\omega, a) = \begin{cases} 2 \cdot \mathbb{1}\{a = \omega\} - 1 & \text{if } a \in \Omega \backslash \{\omega_\varnothing\} \\ 0 & \text{if } a = \omega_\varnothing. \end{cases}$$

In this example, the receiver's actions can be interpreted as "guesses" as to what the underlying state is. The receiver gets utility 1 if they correctly guess the state and -1 if they make an incorrect state. The receiver always gets utility 0 if they guess that the state is $\omega_\varnothing$. On the other hand, the sender's goal is to incentivize the receiver to guess any state other than $\omega_\varnothing$. Suppose there are L queries $\mathcal{Q} = \{q_1, \ldots, q_L\}$, where query q_j reveals the j-th bit of $\mathbf{p}_{i^*}$ (recall that i^* is the index of the true receiver type τ^*). Finally, let $\mathcal{P}$ be the uniform distribution over $\mathcal{T}$. See Figure 4 for an illustration of this setting in two dimensions.

Protocol: Bayesian Persuasion with Commitment to Query

1. Sender commits to querying policy $\pi : \Omega \to \mathcal{Q}$ and signaling policy $\sigma : \Omega \times \mathcal{S} \to \mathcal{M}$.
2. Sender observes state $\omega \sim \mathbf{p}_{\tau^*}$, where $\tau^* \sim \mathcal{P}$.
3. Sender makes query $q \sim \pi(\omega)$ and observes information partition s.
4. Sender sends signal $m \sim \sigma(\omega, s)$, receiver forms posterior $\mathbf{p}_{\tau^*}|m$.
5. Receiver takes action $a_R := \arg\max_{a \in \mathcal{A}} \mathbb{E}_{\omega \sim \mathbf{p}_{\tau^*}}[u_R(\omega, a)|m]$.

Figure 5: Description of the Bayesian persuasion game between a sender and receiver, when the sender can commit to a querying policy.

Theorem 5.7. *Consider the setting of Example 5.6. Let $\epsilon = \frac{1}{2^{L-1}+2}$ and consider any two subsets of queries $\tilde{\mathcal{Q}} \subset \widehat{\mathcal{Q}} \subset \mathcal{Q}$ such that $|\tilde{\mathcal{Q}}| = L - 2$ and $|\widehat{\mathcal{Q}}| = L - 1$. For the remaining query $q = \mathcal{Q} \backslash \widehat{\mathcal{Q}}$ we have that*

$$\mathbb{E}_{\tau \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\mathcal{P}(\eta_{\tilde{\mathcal{Q}} \cap q})[\tau]}(\omega)}[u_S(\omega, m)] - \mathbb{E}_{\tau \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\mathcal{P}(\eta_{\tilde{\mathcal{Q}}})[\tau]}(\omega)}[u_S(\omega, m)] = 0$$

$$\text{but} \quad \mathbb{E}_{\tau \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\mathcal{P}(\eta_{\widehat{\mathcal{Q}} \cap q})[\tau]}(\omega)}[u_S(\omega, m)] - \mathbb{E}_{\tau \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\mathcal{P}(\eta_{\widehat{\mathcal{Q}}})[\tau]}(\omega)}[u_S(\omega, m)] > 0.$$

Proof Sketch. The proof proceeds by showing that under the setting of Example 5.6, it is possible to completely distinguish between all receivers with L queries, but not with $L' < L$ queries. □

Using Example 5.6 as a building block, it is possible to construct an instance in which the greedy non-adaptive querying policy (Algorithm 5) performs exponentially worse than the optimal querying policy.

Corollary 5.8. *Consider the setting of Example 5.6, but with 2^L additional queries of the form "Does $\mathbf{p}_{i^*} = \mathbf{p}_j$?", for every $j \in \{0, \ldots, 2^L - 1\}$. Under this setting, Algorithm 5 requires 2^L queries to obtain the optimal expected sender utility, whereas the optimal querying policy only requires L queries.*

Proof Sketch. The result follows by showing that Algorithm 5 will always make one of the 2^L new queries, whereas the optimal non-adaptive querying policy will make the original L queries in Example 5.6. □

6 Extension: Committing to Query

In line with the classic BP setup, we have so far assumed that the sender has the ability to commit to a signaling policy, but must make all of their oracle queries before seeing the state. In this section, we explore the effects of *further* commitment on the sender's side; namely, the power to commit to a querying policy. We focus on the simplified setting in which the sender can make one costless query ($K = 1$), for which there is no difference between adaptive and non-adaptive querying policies. Unlike in the previous setting without this extra commitment power, the sender's querying policy can now depend on the realized state, and their signaling policy can depend on both the state *and* the outcome of the query.

Definition 6.1 (State-Informed Querying Policy)**.** *A querying policy $\pi : \Omega \to \mathcal{Q}$ is* state-informed *if the query made depends on the state realization.*

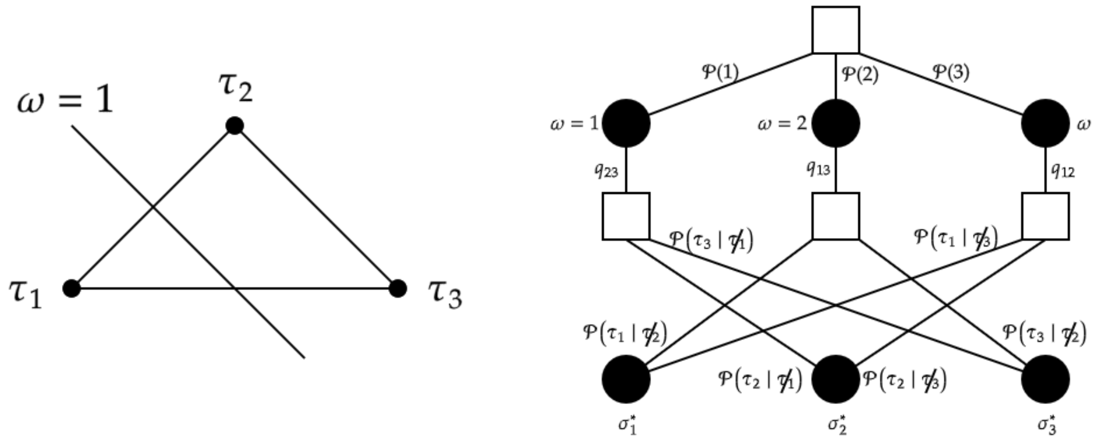

Figure 6: Visualization of Example 6.3. Left: Why seeing the state helps the sender. Right: Why this procedure is incentive-compatible for the receiver.

Similarly, we say that a signaling policy is *partition-informed* if the realized signal depends on both the realized state and the outcome of the query.

Definition 6.2 (Partition-Informed Querying Policy)**.** *A querying policy $\sigma : \Omega \times \mathcal{S} \to \mathcal{M}$ is* partition-informed *if the message sent depends on the realized information partition.*

See Figure 5 for a summary of the modified interaction protocol between the sender, oracle, and receiver. The following example illustrates how the sender can benefit from this additional commitment power.

Example 6.3. *Consider the following natural extension of the binary setting with simulation queries to three states and actions: Let $\Omega = \{-1, 0, 1\}$, $\mathcal{A} = \{-1, 0, 1\}$, $u_S(\omega, a) = a$, and $u_R(\omega, a) = \mathbb{1}\{a = \omega\}$. Let $\mathcal{Q}$ be the set of simulation queries and consider the set of receivers types $\mathcal{T} = \{\tau_1, \tau_2, \tau_3\}$, where $\mathbf{p}_1 = [0.75, 0.25, 0]$, $\mathbf{p}_2 = [0.75 + \delta, 0, 0.25 - \delta]$, $\mathbf{p}_3 = [0, 0.75 + \epsilon, 0.25 - \epsilon]$ for some $\epsilon, \delta > 0$. Observe that after seeing the state realization, the sender will always be able to rule out one of the receiver types (since they all put zero probability mass on a different state), and thus they can use their query to distinguish between the remaining two types. If the sender commits to signaling according to the optimal signaling policy for the realized type, this procedure is Bayesian incentive-compatible since the sender will always send a message to a receiver of type i using signaling policy σ_i^* for each possible state realization. Note that if the sender queried the oracle* before *seeing the state, they would not always be able to distinguish between all three types.*

Equipped with this intuition as to how committing to query can help the sender, we return to the binary setting with simulation queries.

Definition 6.4. *We say that a state-informed querying policy σ* implements *a signaling policy σ_i for receiver type τ_i if $\sigma_i(a|\omega) = \sum_{m \in \mathcal{M}: m[i] = a} \sigma(m|\omega)$ for all $\omega \in \Omega$, $a \in \mathcal{A}$.*

Theorem 6.5. *Given any three signaling policies σ_1, σ_2, σ_3 where σ_1 (resp. σ_2, σ_3) is an arbitrary signaling policy which is Bayesian incentive-compatible for agents of type τ_1 (resp. τ_2, τ_3), there exists a Bayesian incentive-compatible state-informed querying policy which can implement σ_1, σ_2, and σ_3 simultaneously (according to Definition 6.4).*

Proof Sketch. The result follows from writing down the problem of designing a Bayesian incentive-compatible querying and signaling policy in this setting as a linear constraint satisfaction problem (CSP), and showing that this CSP still has a feasible solution whenever additional constraints are added to ensure that σ_1, σ_2, and σ_3 are implemented. □

Note that it is generally only possible to implement *two* signaling policies for different receiver types when querying *before* seeing the state.

7 Conclusion and Future Work

Motivated by recent advances in Large Language Models, as well as more classical settings such as pure exploration in sequential decision-making problems, we study a setting in which the sender in a Bayesian persuasion problem can interact with an oracle for K rounds before trying to persuade the receiver. After showing how the sender can reason about signaling under uncertainty and benefit from this additional query access, we show how to derive the sender's optimal adaptive querying policy via backward induction on an appropriately-defined game tree. Next we show that while in the general setting the sender's optimal non-adaptive querying policy is an NP-Complete problem, it can be solved for efficiently via dynamic programming when there are binary states and actions. Motivated by this observation, we study the sender's expected utility as a function of the queries they make in further detail in the non-adaptive setting. We find that the sender's expected utility exhibits decreasing marginal returns as a function of additional queries in the binary setting. In contrast, this property does not hold in the general setting, which rules out the possibility of achieving good approximation algorithms using the greedy heuristic. Finally, we explore the effects of additional commitment power on the sender's querying policy and find that this additional commitment can significantly benefit the sender. There are several exciting directions for future research.

Imperfect oracle access Throughout this work, we assume that the sender has access to an oracle which is able to *perfectly* simulate the receiver. However in reality the oracle may make mistakes. An LLM may not be able to perfectly replicate the behavior of the person it is being used to simulate. Agents used for pure exploration may not be representative of the larger population. An interesting direction is to analyze the performance of our methods whenever the oracle is allowed to be imprecise, and if necessary, modify our algorithms to be robust to such misspecification.

Properties of the sender's utility in the adaptive setting While our investigation of the sender's utility function in the non-adaptive setting was driven by the NP-Completeness result of Section 5.1, it would be interesting to see which adaptive settings exhibit decreasing marginal returns and which do not.

Unknown receiver utilities While the focus of our work is on settings in which the receiver's utilities are known and the sender is uncertain about the receiver's beliefs, it would be natural to ask how simulation could help the sender achieve higher utility whenever the receiver's utility function is not known to the sender.

Acknowledgements

KH is supported in part by an NDSEG Fellowship. The authors would like to thank James Best, Shuran Zheng, and the participants of the DIMACS Workshop on Foundation Models, Large Language Models, and Game Theory for helpful comments and discussions.

References

[1] Elif Akata, Lion Schulz, Julian Coda-Forno, Seong Joon Oh, Matthias Bethge, and Eric Schulz. Playing repeated games with large language models. *arXiv preprint arXiv:2305.16867*, 2023.

[2] Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of the sixteenth ACM conference on economics and computation*, pages 61–78, 2015.

[3] Martino Bernasconi, Matteo Castiglioni, Alberto Marchesi, Nicola Gatti, and Francesco Trovò. Sequential information design: Learning to persuade in the dark. *arXiv preprint arXiv:2209.03927*, 2022.

[4] Martino Bernasconi, Matteo Castiglioni, Andrea Celli, Alberto Marchesi, Francesco Trovò, and Nicola Gatti. Optimal rates and efficient algorithms for online bayesian persuasion. In *International Conference on Machine Learning*, pages 2164–2183. PMLR, 2023.

[5] Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Learning optimal commitment to overcome insecurity. *Advances in Neural Information Processing Systems*, 27, 2014.

[6] James Brand, Ayelet Israeli, and Donald Ngwe. Using gpt for market research. *Available at SSRN 4395751*, 2023.

[7] Philip Brookins and Jason Matthew DeBacker. Playing games with gpt: What can we learn about a large language model from canonical strategic games? *Available at SSRN 4493398*, 2023.

[8] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *Algorithmic Learning Theory: 20th International Conference, ALT 2009, Porto, Portugal, October 3-5, 2009. Proceedings 20*, pages 23–37. Springer, 2009.

[9] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19):1832–1852, 2011.

[10] Federico Cacciamani, Matteo Castiglioni, and Nicola Gatti. Online mechanism design for information acquisition. *arXiv preprint arXiv:2302.02873*, 2023.

[11] Matteo Castiglioni, Andrea Celli, Alberto Marchesi, and Nicola Gatti. Online bayesian persuasion. *Advances in Neural Information Processing Systems*, 33:16188–16198, 2020.

[12] Matteo Castiglioni, Alberto Marchesi, Andrea Celli, and Nicola Gatti. Multi-receiver online bayesian persuasion. In *International Conference on Machine Learning*, pages 1314–1323. PMLR, 2021.

[13] Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Bayesian persuasion meets mechanism design: Going beyond intractability with type reporting. *arXiv preprint arXiv:2202.00605*, 2022.

[14] Shouyuan Chen, Tian Lin, Irwin King, Michael R Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. *Advances in neural information processing systems*, 27, 2014.

[15] Yiting Chen, Tracy Xiao Liu, You Shan, and Songfa Zhong. The emergence of economic rationality of gpt. *arXiv preprint arXiv:2305.12763*, 2023.

[16] Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 82–90, 2006.

[17] Arianna Degan and Ming Li. Persuasion with costly precision. *Economic Theory*, 72(3): 869–908, 2021.

[18] Paul Duetting, Vahab Mirrokni, Renato Paes Leme, Haifeng Xu, and Song Zuo. Mechanism design for large language models. *arXiv preprint arXiv:2310.10826*, 2023.

[19] Piotr Dworczak and Alessandro Pavan. Preparing for the worst but hoping for the best: Robust (bayesian) persuasion. *Econometrica*, 90(5):2017–2051, 2022.

[20] Sara Fish, Paul Gölz, David C Parkes, Ariel D Procaccia, Gili Rusak, Itai Shapira, and Manuel Wüthrich. Generative social choice. *arXiv preprint arXiv:2309.01291*, 2023.

[21] Jiarui Gan, Rupak Majumdar, Goran Radanovic, and Adish Singla. Bayesian persuasion in sequential decision-making. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 5025–5033, 2022.

[22] Fulin Guo. Gpt agents in game theory experiments. *arXiv preprint arXiv:2305.05516*, 2023.

[23] John J Horton. Large language models as simulated economic agents: What can we learn from homo silicus? *arXiv preprint arXiv:2301.07543*, 2023.

[24] Ju Hu and Xi Weng. Robust persuasion of a privately informed receiver. *Economic Theory*, 72:909–953, 2021.

[25] Emir Kamenica. Bayesian persuasion and information design. *Annual Review of Economics*, 11:249–272, 2019.

[26] Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *American Economic Review*, 101(6):2590–2615, 2011.

[27] Junsol Kim and Byungkyu Lee. Ai-augmented surveys: Leveraging large language models for opinion prediction in nationally representative surveys. *arXiv preprint arXiv:2305.09620*, 2023.

[28] Anton Kolotilin, Tymofiy Mylovanov, Andriy Zapechelnyuk, and Ming Li. Persuasion of a privately informed receiver. *Econometrica*, 85(6):1949–1964, 2017.

[29] Andrew Kosenko. Noisy bayesian persuasion with private information. *mimeo*, 2021.

[30] Svetlana Kosterina. Persuasion with unknown beliefs. *Theoretical Economics*, 17(3):1075–1107, 2022.

[31] Vojtech Kovarik, Caspar Oesterheld, and Vincent Conitzer. Game theory with simulation of other players. *arXiv preprint arXiv:2305.11261*, 2023.

[32] Joshua Letchford, Vincent Conitzer, and Kamesh Munagala. Learning and approximating the optimal strategy to commit to. In *Algorithmic Game Theory: Second International Symposium, SAGT 2009, Paphos, Cyprus, October 18-20, 2009. Proceedings 2*, pages 250–262. Springer, 2009.

[33] Barton L Lipman. Acquisition of/stochastic evidence. 2022.

[34] Nunzio Lorè and Babak Heydari. Strategic behavior of large language models: Game structure vs. contextual framing. *arXiv preprint arXiv:2309.05898*, 2023.

[35] Gabriel Martinez-Roa. Bayesian persuasion in the digital age. 2019.

[36] Sandra Matz, Jake Teeny, Sumer S Vaid, Gabriella M Harari, and Moran Cerf. The potential of generative ai for personalized persuasion at scale. 2023.

[37] Alexei Parakhonyak and Anton Sobolev. Persuasion without priors. *Available at SSRN 4377557*, 2022.

[38] Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023.

[39] Binghui Peng, Weiran Shen, Pingzhong Tang, and Song Zuo. Learning optimal strategies to commit to. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2149–2156, 2019.

[40] Heinrich Peters and Sandra Matz. Large language models can infer psychological dispositions of social media users. *arXiv preprint arXiv:2309.08631*, 2023.

[41] Toshihiro Tsuchihashi. How much do you bid? answers from chatgpt in first-price and second-price auctions. *Journal of Digital Life*, 3, 2023.

[42] Heinrich Von Stackelberg. *Market structure and equilibrium.* Springer Science & Business Media, 2010.

[43] Jibang Wu, Zixuan Zhang, Zhe Feng, Zhaoran Wang, Zhuoran Yang, Michael I Jordan, and Haifeng Xu. Sequential information design: Markov persuasion process and its efficient reinforcement learning. *arXiv preprint arXiv:2202.10678*, 2022.

[44] Liyuan Xu, Junya Honda, and Masashi Sugiyama. A fully adaptive algorithm for pure exploration in linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 843–851. PMLR, 2018.

[45] You Zu, Krishnamurthy Iyer, and Haifeng Xu. Learning to persuade on the fly: Robustness against ignorance. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 927–928, 2021.

Protocol: Bayesian persuasion with Oracle Queries

1. Sender makes K oracle queries, either adaptively (Definition 2.6) or non-adaptively (Definition 2.7).
2. Sender *commits* to signaling policy $\sigma : \Omega \rightarrow \mathcal{M}$.
3. Sender observes state $\omega \sim \mathbf{p}_{\tau^*}$, where $\tau^* \sim \mathcal{P}$.
4. Sender sends signal $m \sim \sigma(\omega)$, receiver forms posterior $\mathbf{p}_{\tau'}|m$ where $\tau' \sim \mathcal{P}$.
5. Receiver takes action $\arg\max_{a \in \mathcal{A}} \mathbb{E}_{\omega \sim \mathbf{p}_{\tau'}}[u_R(\omega, a)|m]$.

Figure 7: Uninformed receiver setting.

A Extension: Uninformed Receivers

In this section, we sketch how our results may be extended to the setting in which the receiver does not know any additional information about the state when compared to the sender, and may hold *incorrect beliefs* about the true state of the world. Under this setting, the world state is drawn according to $\omega \sim \mathbf{p}_{\tau^*}$ where $\tau^* \sim \mathcal{P}$, but the receiver is of type $\tau' \sim \mathcal{P}$, where τ^* and τ' are two independent draws from $\mathcal{P}$. Querying the oracle can provide the sender with information about τ', but not τ^*. The updated setting is described in Figure 7. We now overview how this change to our model affects our results.

1. The sender's optimal signaling policy under uncertainty (Section 3.1) needs to be modified to reflect the fact that the receiver is now uninformed. The sender's optimization now becomes

$$\begin{aligned} \sigma^*_{\mathcal{P}(\mathcal{T}')} &:= \arg\max_{\sigma} \; \mathbb{E}_{\tau \sim \mathcal{P}}[\mathbb{E}_{\omega \sim \mathbf{p}_\tau}[\mathbb{E}_{m \sim \sigma(\omega)}[u_S(\omega, m[\tau])]]] \\ &\quad \text{s.t. } \mathbb{E}_{\omega \sim \mathbf{p}_\tau}[u_R(\omega, m[\tau])|m] \geq \mathbb{E}_{\omega \sim \mathbf{p}_\tau}[u_R(\omega, m[\tau])|m], \forall a \in \mathcal{A}, \tau \in \mathcal{T}', m \in \mathcal{M} \end{aligned}$$

   when the sender is certain the receiver is in subset $\mathcal{T}' \subseteq \mathcal{T}$.

2. The results in Section 3.2 carry over with τ' in place of τ^*.
3. The results of Section 4 carry over with τ' in place of τ^*, with the exception of how $V[\mathcal{T}', 0]$ is computed. In particular, we now set

$$V[\mathcal{T}', 0] := \max_{\sigma^*_{\mathcal{P}(\mathcal{T}')}} \mathbb{E}_{\tau \sim \mathcal{P}(\mathcal{T}'), \tau^* \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_{\tau^*}} \mathbb{E}_{m \sim \sigma^*_{\mathcal{P}(\mathcal{T}')}(\omega)} u_S(\omega, m)$$

4. Algorithm 4 may be modified by replacing τ^* with τ' and setting $V[i, j, 0]$ as

$$V[i, j, 0] := \max_{\sigma^*_{\mathcal{P}(\tau' \in \{i, \ldots j\})}} \mathbb{E}_{\tau \sim \mathcal{P}(\tau' \in \{i, \ldots, j\}), \tau^* \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_{\tau^*}} \mathbb{E}_{m \sim \sigma^*_{\mathcal{P}(\tau' \in \{i, \ldots j\})}(\omega)} u_S(\omega, m)$$

5. Our hardness result of Theorem 5.4 does not depend on whether or not the receiver is informed.
6. The results of Section 6 *do not* carry over to the setting in which the sender is uninformed. Under this setting, being able to commit to querying a certain way does not benefit the sender.

B Appendix for Section 3: How Should We Think About Querying?

B.1 Proof of Theorem 3.7

Theorem 3.7. *For any τ, τ' which are separable (according to Definition 3.6), Algorithm 1 returns a simulation query which separates them in time $\mathcal{O}(A^2 \cdot \mathrm{LP}(d, d+A))$.*

Proof. Observe that under Definition 3.6, there exist states $\omega_1, \omega_2 \in \Omega$ and actions $a_1, a_2 \in \mathcal{A}$ such that

1. $\mathbf{p}_\tau[\omega_1] > \mathbf{p}_{\tau'}[\omega_1]$ and $\mathbf{p}_\tau[\omega_2] < \mathbf{p}_{\tau'}[\omega_2]$
2. $a_1 := \arg\max_{a \in \mathcal{A}} u_R(\omega_1, a)$, $a_2 := \arg\max_{a \in \mathcal{A}} u_R(\omega_2, a)$, and $a_1 \neq a_2$.

Case 1: $\mathbf{p}_\tau[\omega_2] > 0$ The following signaling policy is BIC when receiver type $\mathbf{p}_\tau$ is recommended action $m = a_1$: (1) Set $\sigma(a_1|\omega) = 0$, $\forall \omega \in \Omega \backslash \{\omega_1, \omega_2\}$. (2) Pick $\sigma(a_1|\omega_1)$ and $\sigma(a_1|\omega_2)$ such that

$$(u_R(\omega_1, a_1) - u_R(\omega_1, a_2)) \cdot \sigma(a_1|\omega_1) \cdot \mathbf{p}_\tau[\omega_1] + (u_R(\omega_2, a_1) - u_R(\omega_2, a_2)) \cdot \sigma(a_1|\omega_2) \cdot \mathbf{p}_\tau[\omega_2] = 0.$$

Equivalently, pick $\sigma(a_1|\omega)$ and $\sigma(a_2|\omega)$ such that

$$\frac{\sigma(a_1|\omega_1)}{\sigma(a_1|\omega_2)} = \frac{u_R(\omega_2, a_2) - u_R(\omega_2, a_1)}{u_R(\omega_1, a_1) - u_R(\omega_1, a_2)} \cdot \frac{\mathbf{p}_\tau[\omega_2]}{\mathbf{p}_\tau[\omega_1]}$$

Note that $\frac{u_R(\omega_2, a_2) - u_R(\omega_2, a_1)}{u_R(\omega_1, a_1) - u_R(\omega_1, a_2)} > 0$ and $0 < \frac{\mathbf{p}_\tau[\omega_2]}{\mathbf{p}_\tau[\omega_1]} < \infty$. Therefore $\sigma(a_1|\omega_1)$ and $\sigma(a_1|\omega_2)$ may always be chosen such that they are both strictly greater than zero. Under this setting of $\{\sigma(a_1|\omega)\}_{\omega \in \Omega}$, the BIC expression for receiver $\mathbf{p}_{\tau'}$ is

$$\begin{aligned}
&(u_R(\omega_1, a_1) - u_R(\omega_1, a_2)) \cdot \sigma(a_1|\omega_1) \cdot \mathbf{p}_{\tau'}[\omega_1] + (u_R(\omega_2, a_1) - u_R(\omega_2, a_2)) \cdot \sigma(a_1|\omega_2) \cdot \mathbf{p}_{\tau'}[\omega_2] \\
=&(u_R(\omega_1, a_1) - u_R(\omega_1, a_2)) \cdot \frac{u_R(\omega_2, a_2) - u_R(\omega_2, a_1)}{u_R(\omega_1, a_1) - u_R(\omega_1, a_2)} \cdot \frac{\mathbf{p}_\tau[\omega_2]}{\mathbf{p}_\tau[\omega_1]} \cdot \sigma(a_1|\omega_2) \cdot \mathbf{p}_{\tau'}[\omega_1] \\
&+ (u_R(\omega_2, a_2) - u_R(\omega_2, a_1)) \cdot \sigma(a_1|\omega_2) \cdot \mathbf{p}_{\tau'}[\omega_2] \\
&= (u_R(\omega_2, a_2) - u_R(\omega_2, a_1)) \cdot \sigma(a_1|\omega_2) \cdot (\frac{\mathbf{p}_{\tau'}[\omega_1]}{\mathbf{p}_\tau[\omega_1]} \cdot \mathbf{p}_\tau[\omega_2] - \mathbf{p}_{\tau'}[\omega_2])
\end{aligned}$$

where $(u_R(\omega_2, a_2) - u_R(\omega_2, a_1)) \cdot \sigma(a_1|\omega_2) > 0$ and $\frac{\mathbf{p}_{\tau'}[\omega_1]}{\mathbf{p}_\tau[\omega_1]} \cdot \mathbf{p}_\tau[\omega_2] - \mathbf{p}_{\tau'}[\omega_2] < 0$. This implies that $(u_R(\omega_2, a_2) - u_R(\omega_2, a_1)) \cdot \sigma(a_1|\omega_2) \cdot (\frac{\mathbf{p}_{\tau'}[\omega_1]}{\mathbf{p}_\tau[\omega_1]} \cdot \mathbf{p}_\tau[\omega_2] - \mathbf{p}_{\tau'}[\omega_2]) < 0$, and so receiver $\mathbf{p}_{\tau'}$ will not take action a_1.

Case 2: $\mathbf{p}_\tau[\omega_2] = 0$ Note that if $\sigma(a_2|\omega) = 0$, $\forall \omega \in \Omega \backslash \{\omega_1, \omega_2\}$,

$$(u_R(\omega_1, a_1) - u_R(\omega_1, a_2)) \cdot \sigma(a_1|\omega_1) \cdot \mathbf{p}_\tau[\omega_1] + (u_R(\omega_2, a_1) - u_R(\omega_2, a_2)) \cdot \sigma(a_1|\omega_2) \cdot \mathbf{p}_\tau[\omega_2] > 0$$

as long as $\sigma(a_1|\omega_1) > 0$, since $\mathbf{p}_\tau[\omega_2] = 0$. Therefore, it suffices to pick $\sigma(a_1|\omega_1)$ and $\sigma(a_1|\omega_2)$ such that

$$(u_R(\omega_1, a_1) - u_R(\omega_1, a_2)) \cdot \sigma(a_1|\omega_1) \cdot \mathbf{p}_{\tau'}[\omega_1] + (u_R(\omega_2, a_1) - u_R(\omega_2, a_2)) \cdot \sigma(a_1|\omega_2) \cdot \mathbf{p}_{\tau'}[\omega_2] < 0$$

and $\sigma(a_1|\omega_1) > 0$ simultaneously. Note that this is always feasible since $\mathbf{p}_{\tau'}[\omega_2] > 0$ and $\sigma(a_1|\omega_1)$ can be arbitrarily close to zero. $\square$

Algorithm 6 Computing a query to separate n types

Require: Priors $\mathbf{p}_1$, $\mathbf{p}_2, \ldots, \mathbf{p}_n$

for all n-tuples of actions $(m, a^{(2)}, \ldots, a^{(n)}) \in \mathcal{A} \times \mathcal{A} \times \cdots \times \mathcal{A}$ **do**

Solve the following linear program:

$$\min_{\{\sigma(m|\omega)\}_{\omega\in\Omega}} \quad \eta$$

$$\sum_{\omega\in\Omega}(u_R(\omega, m) - u_R(\omega, a)) \cdot \sigma(m|\omega) \cdot \mathbf{p}_1[\omega] \geq 0, \ \forall a \in \mathcal{A}$$

$$\sum_{\omega\in\Omega}(u_R(\omega, m) - u_R(\omega, a^{(2)})) \cdot \sigma(m|\omega) \cdot \mathbf{p}_2[\omega] \leq \eta$$

$$\vdots$$

$$\sum_{\omega\in\Omega}(u_R(\omega, m) - u_R(\omega, a^{(n)})) \cdot \sigma(m|\omega) \cdot \mathbf{p}_n[\omega] \leq \eta$$

$$0 \leq \sigma(m|\omega) \leq 1, \ \forall \omega \in \Omega$$

if $\eta < 0$ **then**

return query (σ', m), where σ' is a signaling policy which maximizes Optimization 1.

end if

end for

return `Fail`

B.2 A Procedure for Finding All Simulation Queries

Algorithm 6 finds a simulation query to separate n different receiver types, if such a query exists. The formal analysis proceeds analogously to the proof of Theorem 3.7 with the exception of the runtime analysis, which is exponentially worse.

C Appendix for Section 4: Adaptive Querying Policies

C.1 Costly Queries in the Binary Setting

Corollary C.1. *In the binary setting with simulation queries, the update step of Algorithm 3 simplifies to computing* $V[(i, j), k] = \max_{q\in\{i,\ldots,j\}} V[(i, j), k, q]$ *for all* $1 \leq i < j \leq T$*, where*

$$\begin{aligned} V[(i, j), k, q] := & V[(i, q), k-1] \cdot \mathbb{P}_{\mathcal{P}}(p_{\mathcal{T}^*} \leq p_q | p_{\mathcal{T}^*} \in \{p_i, \ldots, p_j\}) \\ & + V[(q+1, j), k-1] \cdot \mathbb{P}_{\mathcal{P}}(p_{\mathcal{T}^*} > p_q | p_{\mathcal{T}^*} \in \{p_i, \ldots, p_j\}) - c_q \end{aligned}$$

The runtime of Algorithm 3 in this setting is $\mathcal{O}(T^4 K)$.

Proof. Correctness follows from the proof of Theorem 4.4 and the observation that in the binary setting with simulation queries, the set of feasible information partitions simplifies to $\mathcal{F} = \{\{\tau_i, \ldots, \tau_j\} : 1 \leq i \leq j \leq T\}$. Note that in the binary setting with simulation queries, $Q = T - 1$ and each query splits the set of receiver types into at most two partitions, so the time to compute each $V[(i, j), k]$ is $\mathcal{O}(T)$. Finally, observe that under this setting $F = \mathcal{O}(T^2)$. □

D Appendix for Section 5: Non-Adaptive Querying Policies

D.1 Proof of Theorem 5.2

Theorem 5.2. *In the binary setting with simulation queries, Algorithm 4 computes the sender's optimal non-adaptive querying policy in* $\mathcal{O}(T^3 K)$ *time.*

Proof. **Correctness** We proceed via induction on K. The base case when $K = 0$ is trivially optimal. Suppose that $U[i, j, K-1]$ is the maximum expected value for making $K-1$ queries in the receiver interval $\{i, \ldots, j\}$. We can write the sender's maximum expected utility for K queries in the range $\{i, \ldots, j\}$ as

$$
\begin{aligned}
U[1, j, K] := & \max_{q_1, q_2, \ldots, q_K \in \mathcal{Q}} \Bigg\{ V[1, q_1, 0] + \sum_{k=1}^{K-1} V[q_k + 1, q_{k+1}, 0] \\
& + V[q_K + 1, j, 0] \Bigg\} \\
& = \max_{q_K \in \mathcal{Q}} \Bigg\{ V[q_K + 1, j, 0] \\
& + \max_{q_1, q_2, \ldots, q_{K-1} \in \mathcal{Q}} \Bigg\{ V[1, q_1, 0] + \sum_{k=1}^{K-2} V[q_k + 1, q_{k+1}, 0] \\
& + V[q_{K-1} + 1, q_K, 0] \Bigg\} \Bigg\} \\
& = \max_{q_K \in \mathcal{Q}} \Bigg\{ V[q_K + 1, j, 0] + V[1, q_K, K-1] \Bigg\} \\
& =: V[1, j, K]
\end{aligned}
$$

where $q_1 < q_2 < \ldots < q_K$. By the inductive hypothesis, we know that $V[1, q_K, K-1]$ is the maximum expected utility achievable for the sender on the interval $\{1, \ldots, q_K\}$ when making $K-1$ queries, for any q_K. Therefore $V[1, j, K]$ must be the maximum expected utility the sender can achieve on the interval $\{i, \ldots, j\}$ when making K queries.

Runtime analysis Observe that each $\mathbb{P}_{\mathcal{P}}(\tau^* \in \{q+1, \ldots, i\})$ for $q \in Q$, $i \in [1, \ldots, T]$ may be pre-computed before the algorithm is run in time $\mathcal{O}(T)$, and that there are at most $\mathcal{O}(T^2)$ such probabilities to pre-compute. Similarly each $V[1, j, k]$ value may be computed in time $\mathcal{O}(Q) = \mathcal{O}(T)$, and there are at most $\mathcal{O}(T^2 K)$ such values to compute. □

D.2 Costly Queries in the Binary Setting

Algorithm 4 may be used to compute the optimal non-adaptive querying policy for the costly setting by (1) setting $K = T - 1$ (2) introducing a new query $q_\varnothing$ such that $c_{q_\varnothing} = 0$, and using the following modified update step:

$$V[1, j, k] := \max_{q \in \mathcal{Q}} V[q + 1, j, 0] + V[1, q, k-1] - c_q.$$

Corollary D.1. *In the binary setting with costly simulation queries, the above algorithm computes the sender's non-adaptive querying policy in $\mathcal{O}(T^3 K)$ time.*

Proof. **Correctness** We proceed via induction on K. The base case when $K = 0$ is trivially optimal. Suppose that $U[i, j, K-1]$ is the maximum expected value for making $K-1$ queries in the receiver interval $\{i, \ldots, j\}$. We can write the sender's maximum expected utility for K

queries in the range $\{i, \dots, j\}$ as

$$\begin{aligned}
U[1,j,K] := \max_{q_1,q_2,\dots,q_K \in \mathcal{Q}} &\Bigg\{ V[1,q_1,0] + \sum_{k=1}^{K-1} V[q_k+1, q_{k+1}, 0] \\
&+ V[q_K+1, j, 0] - \sum_{k=1}^{K} c_{q_k} \Bigg\} \\
&= \max_{q_K \in \mathcal{Q}} \Bigg\{ V[q_K+1, j, 0] - c_{q_K} \\
&+ \max_{q_1,q_2,\dots,q_{K-1} \in \mathcal{Q}} \Bigg\{ V[1,q_1,0] + \sum_{k=1}^{K-2} V[q_k+1, q_{k+1}, 0] \\
&+ V[q_{K-1}+1, q_K, 0] - \sum_{k=1}^{K-1} c_{q_k} \Bigg\}\Bigg\} \\
&= \max_{q_K \in \mathcal{Q}} \Bigg\{ V[q_K+1, j, 0] + V[1, q_K, K-1] - c_{q_K} \Bigg\} \\
&=: V[1,j,K]
\end{aligned}$$

where $q_1 < q_2 < \ldots < q_K$. By the inductive hypothesis, we know that $V[1, q_K, K-1]$ is the maximum expected utility achievable for the sender on the interval $\{1, \dots, q_K\}$ when making $K-1$ queries, for any q_K. Therefore $V[1,j,K]$ must be the maximum expected utility the sender can achieve on the interval $\{i, \dots, j\}$ when making K queries.

Runtime analysis The runtime analysis is identical to the analysis in the proof of Theorem 5.2. □

D.3 Proof of Theorem 5.4

The following definitions will be useful for the proof of Theorem 5.4.

Definition D.2 (Complete Separation)**.** *We say that a set of queries $\mathcal{Q}' \subseteq \mathcal{Q}$ completely separates the set of receiver types $\mathcal{T}$ if, for every type $\tau \in \mathcal{T}$,*

$$\eta_{\mathcal{Q}'}[\tau] = \{\tau\},$$

where $\eta_{\mathcal{Q}'}[\tau]$ is defined as in Definition 5.1.

In the set cover problem there is a *universe* of elements U and a collection of (sub)sets of elements S. The following decision problem is NP-Hard

Definition D.3 (Set Cover Decision Problem)**.** *Given (U, S, K) where U is a universe of elements, S is a collection of subsets of elements in U, and $K \in \mathbb{N}$, does there exist a collection of subsets $S' \subseteq S$ of size $|S'| \leq K$ such that $\bigcup_{s \in S'} s = U$?*

We use the shorthand $\texttt{set\_cover}(U, S, K)$ and $\texttt{non-adaptive}(\mathcal{T}, \mathcal{P}, \mathcal{Q}, K, u)$ to refer to the Set Cover and Non-Adaptive decision problems respectively.

Theorem 5.4. *The Non-Adaptive Decision Problem (Definition 5.3) is NP-Complete.*

Proof. Observe that given a candidate solution $\mathcal{Q}'$ and the set of corresponding BIC signaling policies for each receiver subset, we can check whether the sender's expected utility is at least u in polynomial time, by computing the expectation. This establishes that the problem is in NP. To prove NP-Hardness, we proceed via a reduction to set cover. Given an arbitrary set cover decision problem $\texttt{set\_cover}(U, S, K)$,

1. Create a set of receiver types $\mathcal{T}(U)$. Specifically, add type $\tau_\varnothing$ to $\mathcal{T}(U)$, and add a receiver type τ_e to $\mathcal{T}(U)$ for every element $e \in U$.

2. For each subset $s \in S$, create a query q_s that separates the receiver types $\{\tau_e\}_{e \in s}$ from both each other and the other types $\mathcal{T}(U) \backslash \{\tau_e\}_{e\in s}$. For example if $s = \{1,2,3\}$, then $q_s = \{\{\tau_1\}, \{\tau_2\}, \{\tau_3\}, \mathcal{T}(U)\backslash\{\tau_1, \tau_2, \tau_3\}\}$. Denote the resulting set of queries by $\mathcal{Q}(S)$.

3. Let $\mathcal{P}$ be the uniform prior over $\mathcal{T}(U)$. Set $\{\mathbf{p}_\tau\}_{\tau\in\mathcal{T}}$ such that each receiver type has a different optimal signaling policy.[3] Set $u = \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma^*_\tau(\omega)}[u_S(\omega, a)]$, where σ^*_τ is the optimal signaling policy when the receiver is known to be of types τ.

Part 1: Suppose `set_cover`$(U, S, K) =$ `yes`. Let S' denote the set of subsets which covers S, and let $\mathcal{Q}' := \{q_s\}_{s\in S'}$ be the corresponding set of queries in the `query_separation` instance. Let us consider each $\tau \in \mathcal{T}$ on a case-by-case basis.

Case 1.1: $\tau \in \mathcal{T}\backslash\{\tau_\varnothing\}$. Since S' covers S, $\{\tau\}$ is a partition induced by at least one $q \in \mathcal{Q}'$ by construction, and so $\eta_{\mathcal{Q}'}[\tau] = \{\tau\}$.

Case 1.2: $\tau = \tau_\varnothing$. Likewise since S' covers S, every $\tau \in \mathcal{T}\backslash\{\tau_\varnothing\}$ is *not* contained in at least one $s_{q,\tau_\varnothing}$ by construction. Therefore $\eta_{\mathcal{Q}'}[\tau_\varnothing] = \{\tau_\varnothing\}$.

Putting the two cases together, we see that $\mathcal{Q}'$ completely separates $\mathcal{T}(U)$ according to Definition D.2, and so the sender will be able to determine the receiver's type and achieve optimal utility. Therefore `non-adaptive`$(\mathcal{T}(U), \mathcal{P}, \mathcal{Q}(S), K, u) =$ `yes`.

Part 2: Suppose `set_cover`$(U, S, K) =$ `no`. Consider any set of queries $\mathcal{Q}' \subseteq \mathcal{Q}(S)$ such that $|\mathcal{Q}'| \leq K$. Note that there is a one-to-one mapping between queries in $\mathcal{Q}(S)$ and subsets in S, and so we can denote the set of subsets corresponding to $\mathcal{Q}'$ by $S' := \{s\}_{q_s\in\mathcal{Q}'}$. Since S' does not cover S, there must be at least one element $e_{S'} \in S\backslash(\bigcup_{z\in S'} z)$. If there are multiple such elements, pick one arbitrarily. By the construction of $\mathcal{Q}$, we know that $\tau_{e_{S'}}$ falls in the same partition as $\tau_\varnothing$ (i.e. $s_{q,\tau_\varnothing}$) for every $q \in \mathcal{Q}'$. Therefore $\eta_{\mathcal{Q}'}[\tau_\varnothing] \neq \{\tau_\varnothing\}$, and so $\mathcal{Q}'$ does not completely separate $\mathcal{T}(U)$ according to Definition D.2. Since the sender cannot perfectly distinguish between all receiver types and our choice of $\mathcal{Q}'$ was arbitrary, this implies that `non-adaptive`$(\mathcal{T}(U), \mathcal{P}, \mathcal{Q}(S), K, u) =$ `no`. $\square$

D.4 Proof of Theorem 5.5

Theorem 5.5. *Consider two sets of queries $\widehat{\mathcal{Q}}$, $\tilde{\mathcal{Q}}$, where $\tilde{\mathcal{Q}} \subseteq \widehat{\mathcal{Q}} \subseteq \mathcal{Q}$, and any query $q \in \mathcal{Q}$. In the binary setting, we have that*

$$\begin{aligned}&\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\mathcal{P}(\eta_{\tilde{\mathcal{Q}}\cap q})[\tau]}(\omega)}[u_S(\omega, m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\mathcal{P}(\eta_{\tilde{\mathcal{Q}}})[\tau]}(\omega)}[u_S(\omega, m)]\\ \geq &\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\mathcal{P}(\eta_{\widehat{\mathcal{Q}}\cap q})[\tau]}(\omega)}[u_S(\omega, m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\mathcal{P}(\eta_{\widehat{\mathcal{Q}}})[\tau]}(\omega)}[u_S(\omega, m)].\end{aligned}$$

In other words, the sender's expected utility exhibits decreasing marginal returns from additional queries.

[3] Note that it is always possible to do this. For example, in the binary setting the optimal signaling policy will be diffrent for two receiver types with priors $p' \neq p$, $p, p' \leq 0.5$.

Proof. For some arbitrary set of queries $\mathcal{Q}'$ and query q,

$$\begin{aligned}
&\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}'\cap q}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}'}[\tau]}(\omega)}[u_S(\omega,m)]\\
&=\sum_{\eta\in H_{\mathcal{Q}'\cap q}}\sum_{\tau\in\eta}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau)\cdot\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_\eta(\omega)}[u_S(\omega,m)] - \sum_{\eta\in H_{\mathcal{Q}'}}\sum_{\tau\in\eta}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau)\cdot\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_\eta(\omega)}[u_S(\omega,m)]\\
&=\sum_{\tau\in\eta_1'}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau)\cdot\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_1'}(\omega)}[u_S(\omega,m)] + \sum_{\tau\in\eta_2'}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau)\cdot\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_2'}(\omega)}[u_S(\omega,m)]\\
&-\sum_{\tau\in\eta'}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau)\cdot\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta'}(\omega)}[u_S(\omega,m)]
\end{aligned}$$

where η' is the partition which is split into η_1' and η_2' by query q. Therefore in order to show submodularity it suffices to prove

$$\begin{aligned}
&\sum_{\tau\in\tilde{\eta}_1}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau)\cdot\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\tilde{\eta}_1}(\omega)}[u_S(\omega,m)] + \sum_{\tau\in\tilde{\eta}_2}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau)\cdot\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\tilde{\eta}_2}(\omega)}[u_S(\omega,m)]\\
&-\sum_{\tau\in\tilde{\eta}}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau)\cdot\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\tilde{\eta}}(\omega)}[u_S(\omega,m)] \geq \sum_{\tau\in\widehat{\eta}_1}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau)\cdot\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\widehat{\eta}_1}(\omega)}[u_S(\omega,m)]\\
&+\sum_{\tau\in\widehat{\eta}_2}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau)\cdot\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\widehat{\eta}_2}(\omega)}[u_S(\omega,m)] - \sum_{\tau\in\eta'}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau)\cdot\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\widehat{\eta}}(\omega)}[u_S(\omega,m)]
\end{aligned}$$

Let $\tilde{\eta}$ contain the interval of types from $\tilde{L}$ to $\tilde{H}$, $\widehat{\eta}$ contain the interval of types from $\widehat{L}$ to $\widehat{H}$, and q separate both intervals at M. Note that $\tilde{L}\leq\widehat{L}\leq M\leq\widehat{H}\leq\tilde{H}$. Plugging in the form of the sender's optimal signaling policy (Corollary 3.3), it suffices to show that

$$\tilde{u}_1(\tilde{i}_1)+\tilde{u}_2(\tilde{i}_2)-\tilde{u}(\tilde{i})\geq\widehat{u}_1(\widehat{i}_1)+\widehat{u}_2(\widehat{i}_2)-\widehat{u}(\widehat{i})$$

where

$$\tilde{u}(j) := \sum_{i=\tilde{L}}^{j\wedge\tilde{H}}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i)\cdot(p_i+(1-p_i)\cdot\frac{p_j}{1-p_j})$$

$$\tilde{u}_1(j) := \sum_{i=\tilde{L}}^{j\wedge M}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i)\cdot(p_i+(1-p_i)\cdot\frac{p_j}{1-p_j})$$

$$\tilde{u}_2(j) := \sum_{i=M+1}^{j\wedge\tilde{H}}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i)\cdot p_i+(1-p_i)\cdot\frac{p_j}{1-p_j}$$

$$\widehat{u}(j) := \sum_{i=\widehat{L}}^{j\wedge\widehat{H}}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i)\cdot(p_i+(1-p_i)\cdot\frac{p_j}{1-p_j})$$

$$\widehat{u}_1(j) := \sum_{i=\widehat{L}}^{j\wedge M}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i)\cdot(p_i+(1-p_i)\cdot\frac{p_j}{1-p_j})$$

$$\widehat{u}_2(j) := \sum_{i=M+1}^{j\wedge\widehat{H}}\mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i)\cdot(p_i+(1-p_i)\cdot\frac{p_j}{1-p_j}),$$

and $\tilde{i}$ is the index which maximizes $\tilde{u}(j)$, and $\tilde{i}_1$, $\tilde{i}_2$, $\widehat{i}$, $\widehat{i}_1$, and $\widehat{i}_2$ are defined analogously. We now prove submodularity by a case-by-case analysis of $\tilde{i}$.

Case 1: $\tilde{i} \leq M$ Since $\tilde{i} \leq M$, we know that $\tilde{u}_1(\tilde{i}_1) = \tilde{u}(\tilde{i})$. Observe that $\tilde{u}_2(\tilde{i}_2) \geq \widehat{u}_2(\widehat{i}_2)$, so it suffices for $\widehat{u}_1(\widehat{i}_1) - \widehat{u}(\widehat{i}) \leq 0$, which is always the case.

Case 2: $M + 1 \leq \tilde{i} \leq \widehat{H}$ Observe that under this setting

$$\begin{aligned}
\widehat{u}(\widehat{i}) - \tilde{u}(\tilde{i}) = & \sum_{i=\widehat{L}}^{\widehat{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}}}{1-p_{\widehat{i}}}) - \sum_{i=\tilde{L}}^{\widehat{L}-1} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) \\
& - \sum_{i=\widehat{L}}^{\tilde{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) \\
\geq & - \sum_{i=\tilde{L}}^{\widehat{L}-1} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}})
\end{aligned}$$

Using this fact, as well as the fact that $\tilde{u}_1(\tilde{i}_1) - \widehat{u}_1(\widehat{i}_1) \geq \sum_{i=\tilde{L}}^{\widehat{L}-1} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}_1}}{1-p_{\widehat{i}_1}})$, we see that it suffices to show that

$$\sum_{i=\tilde{L}}^{\widehat{L}-1} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}_1}}{1-p_{\widehat{i}_1}}) - \sum_{i=\tilde{L}}^{\widehat{L}-1} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) \geq 0,$$

as $\tilde{u}_2(\tilde{i}_2) - \widehat{u}_2(\widehat{i}_2) \geq 0$. However note that this is trivially true, since $\tilde{i} \geq M+1$ implies that $\tilde{i} > i_{a_i}$ (which in turn implies that $p_{\tilde{i}} < p_{\widehat{i}_1}$).

Case 3: $\widehat{H} < \tilde{i}$ We still have that

$$\begin{aligned}
\widehat{u}(\widehat{i}) - \tilde{u}(\tilde{i}) = & \sum_{i=\widehat{L}}^{\widehat{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}}}{1-p_{\widehat{i}}}) - \sum_{i=\tilde{L}}^{\tilde{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) \\
& - \sum_{i=\widehat{L}}^{\tilde{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}),
\end{aligned}$$

So it suffices to show that

$$\begin{aligned}
& \tilde{u}_2(\tilde{i}_2) - \widehat{u}_2(\widehat{i}_2) + \sum_{i=\tilde{L}}^{\widehat{L}-1} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}_1}}{1-p_{\widehat{i}_1}}) - \sum_{i=\tilde{L}}^{\widehat{L}-1} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) \\
& -(\sum_{i=\widehat{L}}^{\tilde{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) - \sum_{i=\widehat{L}}^{\widehat{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}}}{1-p_{\widehat{i}}})) \geq 0.
\end{aligned}$$

Simplifying, it suffices to show

$$\tilde{u}_2(\tilde{i}_2) - \widehat{u}_2(\widehat{i}_2) - (\sum_{i=\widehat{L}}^{\tilde{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) - \sum_{i=\widehat{L}}^{\widehat{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}}}{1-p_{\widehat{i}}})) \geq 0.$$

or

$$\sum_{i=\widehat{L}}^{\tilde{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) - \sum_{i=\widehat{L}}^{\widehat{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}}}{1-p_{\widehat{i}}}) \leq \tilde{u}_2(\tilde{i}_2) - \widehat{u}_2(\widehat{i}_2).$$

We will now show this. Since $\widehat{i}$ is the index which maximizes $\widehat{u}(j)$,

$$
\begin{aligned}
&\sum_{i=\widehat{L}}^{\tilde{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) - \sum_{i=\widehat{L}}^{\widehat{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}}}{1-p_{\widehat{i}}}) \\
\leq &\sum_{i=\widehat{L}}^{\tilde{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) - \sum_{i=\widehat{L}}^{\widehat{i}_2} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}_2}}{1-p_{\widehat{i}_2}}) \\
= &\sum_{i=\widehat{L}}^{M} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) - \sum_{i=\widehat{L}}^{M} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}_2}}{1-p_{\widehat{i}_2}}) \\
+ &\sum_{i=M+1}^{\tilde{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) - \sum_{i=M+1}^{\widehat{i}_2} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}_2}}{1-p_{\widehat{i}_2}})
\end{aligned}
$$

Next, observe that $\sum_{i=\widehat{L}}^{M} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) - \sum_{i=\widehat{L}}^{M} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}_2}}{1-p_{\widehat{i}_2}}) < 0$, since we are in the case where $\tilde{i} > \widehat{H}$, and so $\tilde{i} > \widehat{i}_2$ (which implies that $p_{\tilde{i}} < p_{\widehat{i}_2}$). Using this fact, we have that

$$
\begin{aligned}
&\sum_{i=\widehat{L}}^{\tilde{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) - \sum_{i=\widehat{L}}^{\widehat{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}}}{1-p_{\widehat{i}}}) \\
\leq &\sum_{i=M+1}^{\tilde{i}} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}}}{1-p_{\tilde{i}}}) - \sum_{i=M+1}^{\widehat{i}_2} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}_2}}{1-p_{\widehat{i}_2}}) \\
\leq &\sum_{i=M+1}^{\tilde{i}_2} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\tilde{i}_2}}{1-p_{\tilde{i}_2}}) - \sum_{i=M+1}^{\widehat{i}_2} \mathbb{P}_{\mathcal{P}(\mathcal{T})}(\tau_i) \cdot (p_i + (1-p_i) \cdot \frac{p_{\widehat{i}_2}}{1-p_{\widehat{i}_2}})
\end{aligned}
$$

where the last line follows from the fact that $\tilde{i}_2$ maximizes $\tilde{u}_2(j)$. □

D.5 Greedy Approximation in the Binary Setting

Corollary D.4. *In the binary setting, the non-adaptive querying policy $\mathcal{Q}_K$ produced by Algorithm 5 is a $(1-1/e)$-approximation of $\mathcal{Q}_K^*$ and can be computed in time $\mathcal{O}(T^4)$.*

Proof. The proof of Corollary D.4 proceeds similarly to the standard proof for the (approximate) optimality of the greedy algorithm when optimizing a submodular function. We include it here for completeness. Recall that $\eta_{\mathcal{Q}_i}[\tau]$ is the partition for which type τ belongs to after making queries $\mathcal{Q}_i$. Define the *non-adaptive* benchmark $\mathcal{Q}_K^* = \{q_1, \ldots, q_K\}$ to be

$$\mathcal{Q}_K^* := \arg \max_{\mathcal{Q}_K \subseteq \mathcal{Q}; |\mathcal{Q}_K|=K} \mathbb{E}_{\tau \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\eta_{\mathcal{Q}_K}[\tau]}(\omega)}[u_S(\omega, m)].$$

Observe that $\eta_{\mathcal{Q}_i \cup \mathcal{Q}_K^*}[\tau]$ is the partition which τ belongs to after making queries $\mathcal{Q}_i \cup \mathcal{Q}^*$ and $\eta_{\mathcal{Q}_i \cup \{q_1,\ldots,q_j\}}[\tau]$ is the partition which type τ belongs to after making queries $\mathcal{Q}_i \cup \{q_1, \ldots, q_j\}$. Note that $\eta_{\mathcal{Q}_i \cup \{q_1,\ldots,q_K\}}[\tau] = \eta_{\mathcal{Q}_i \cup \mathcal{Q}_K^*}[\tau]$, $\forall i, \tau$.

$$\mathbb{E}_{\tau \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\eta_{\mathcal{Q}_K^*}[\tau]}(\omega)}[u_S(\omega, m)] \leq \mathbb{E}_{\tau \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\eta_{\mathcal{Q}_i}[\tau]}(\omega)}[u_S(\omega, m)] + \sum_{j=1}^{K} u_{\mathcal{Q}_i \cup \{q_1,\ldots,q_j\}}(q_j)$$

where we define the shorthand

$$
\begin{aligned}
u_{\mathcal{Q}_i \cup \{q_1,\ldots,q_j\}}(q_j) := &\mathbb{E}_{\tau \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\eta_{\mathcal{Q}_i \cup \{q_1,\ldots,q_j\}}[\tau]}(\omega)}[u_S(\omega, m)] \\
&- \mathbb{E}_{\tau \sim \mathcal{P}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\eta_{\mathcal{Q}_i \cup \{q_1,\ldots,q_{j-1}\}}[\tau]}(\omega)}[u_S(\omega, m)].
\end{aligned}
$$

Observe that by submodularity (Theorem 5.5),

$$u_{\mathcal{Q}_i\cup\{q_1,\ldots,q_{j-1}\}}(q_j) \le u_{\mathcal{Q}_i}(q_j)$$

Combining the above two statements, we get that

$$\begin{aligned}\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)] &\le \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_i}[\tau]}(\omega)}[u_S(\omega,m)] + \sum_{j=1}^{K} u_{\mathcal{Q}_i}(q_j)\\ &\le \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_i}[\tau]}(\omega)}[u_S(\omega,m)] + K\cdot u_{\mathcal{Q}_i}(x_{i+1})\end{aligned}$$

Rearranging, we get that

$$\begin{aligned}u_{\mathcal{Q}_i}(x_{i+1}) &= \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_i\cup x_{i+1}}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_i}[\tau]}(\omega)}[u_S(\omega,m)]\\ &\ge \frac{1}{K}(\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_i}[\tau]}(\omega)}[u_S(\omega,m)])\end{aligned}$$

Adding and subtracting $\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)]$ from the left hand side, we get that

$$\begin{aligned}&\left(\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_i}[\tau]}(\omega)}[u_S(\omega,m)]\right)\\ &-\left(\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_i\cup x_{i+1}}[\tau]}(\omega)}[u_S(\omega,m)]\right)\\ \ge&\frac{1}{K}(\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_i}[\tau]}(\omega)}[u_S(\omega,m)])\end{aligned}$$

or equivalently,

$$\begin{aligned}&\left(\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_i}[\tau]}(\omega)}[u_S(\omega,m)]\right)\\ &-\left(\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_{i+1}}[\tau]}(\omega)}[u_S(\omega,m)]\right)\\ \ge&\frac{1}{K}(\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_i}[\tau]}(\omega)}[u_S(\omega,m)]).\end{aligned}$$

Rearranging,

$$\begin{aligned}&\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_{i+1}}[\tau]}(\omega)}[u_S(\omega,m)]\\ \le&(1-\frac{1}{K})\left(\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_i}[\tau]}(\omega)}[u_S(\omega,m)]\right)\end{aligned}$$

By induction,

$$\begin{aligned}&\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_K}[\tau]}(\omega)}[u_S(\omega,m)]\\ \le&(1-\frac{1}{K})^K\left(\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\mathcal{T}}(\omega)}[u_S(\omega,m)]\right)\\ \le&\frac{1}{e}\cdot\left(\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\mathcal{T}}(\omega)}[u_S(\omega,m)]\right)\end{aligned}$$

where $\eta_{\mathcal{Q}_0}[\tau]=\mathcal{T}$ is the set of all possible receiver types. Finally, adding $\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\mathcal{T}}(\omega)}[u_S(\omega,m)]$ to each side and rearranging gets us the desired result:

$$\begin{aligned}&\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{\mathcal{Q}_K}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\mathcal{T}}(\omega)}[u_S(\omega,m)]\\ \ge&(1-1/e)\cdot(\mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\eta_{Q_K^*}[\tau]}(\omega)}[u_S(\omega,m)] - \mathbb{E}_{\tau\sim\mathcal{P}}\mathbb{E}_{\omega\sim\mathbf{p}_\tau}\mathbb{E}_{m\sim\sigma_{\mathcal{T}}(\omega)}[u_S(\omega,m)])\end{aligned}$$

□

Algorithm 7 Non-Adaptive Greedy Querying Policy with Costs

Require: Query cost $c > 0$

Set $\mathcal{Q}_0^{\text{cost}} = \emptyset$

while $j \leq \min\{T, Q\} - 1$ **do**

Let

$$x_{j+1} = \arg\max_{q \in \mathcal{Q}} \mathbb{E}_{\tau \sim \mathcal{T}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\eta_{\mathcal{Q}_j^{\text{cost}}}[\tau](\omega)}} [u_S(\omega, m)]$$

if $\mathbb{E}_{\tau \sim \mathcal{T}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\eta_{Q_j^{\text{cost}} \cup x_{j+1}}[\tau](\omega)}} [u_S(\omega, m)] - c \geq \mathbb{E}_{\tau \sim \mathcal{T}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\eta_{Q_j^{\text{cost}}}[\tau](\omega)}} [u_S(\omega, m)]$

then

Set $\mathcal{Q}_{j+1}^{\text{cost}} \leftarrow \mathcal{Q}_j^{\text{cost}} \cup \{x_{j+1}\}$

else

`return` $\mathcal{Q}_j^{\text{cost}}$

end if

end while

D.6 Greedy Approximation in the Costly Binary Setting

Define the non-adaptive *costly* benchmark $\mathcal{Q}^{*\text{cost}} = \{q_1^{\text{cost}}, \ldots\}$ to be

$$\mathcal{Q}^{*\text{cost}} := \arg \max_{\mathcal{Q}^{\text{cost}} \subseteq \mathcal{Q}} \mathbb{E}_{\tau \sim \mathcal{T}} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma_{\eta_{\mathcal{Q}^{\text{cost}}}[\tau](\omega)}} [u_S(\omega, m)] - \sum_{q \in \mathcal{Q}^{\text{cost}}} c_q.$$

Corollary D.5. *In the binary setting with costly queries, the non-adaptive querying policy $\mathcal{Q}^{cost}$ returned by Algorithm 7 is a $(1 - 1\backslash e)$-approximation of $\mathcal{Q}^{*cost}$ and can be computed in time $\mathcal{O}(T^3 K)$.*

Proof. Observe that the greedy algorithm (and the optimal querying policy) will query at most T times. Consider a dummy query $q_\varnothing$ which is costless, but has no effect on the current partition. Observe that if this query is ever selected in some round j', then it is selected for all rounds $j \geq j'$. Under this modified set of queries, the sender's utility is still submodular (according to Theorem 5.5). To see this, observe that if query $q_\varnothing$ is selected, then the marginal gain is always exactly zero. On the other hand if any query in $\mathcal{Q}$ is selected, then the proof of submodularity follows exactly as in the proof of Theorem 5.5 (since the cost cancels out on both sides). The analysis of the approximate optimality of greedy then proceeds exactly as in the proof of Corollary D.4. □

D.7 Proof of Theorem 5.7

Observe that under no signal (or an uninformative signal), each receiver will guess $\omega_\varnothing$ as long as

$$1 \cdot (1 - (M \cdot 2^L + 1) \cdot \epsilon) - N \cdot (M \cdot 2^L + 1) \cdot \epsilon < 0$$

(i.e. the receiver prefers not guessing to guessing according to their type) and

$$1 \cdot (1 - (M \cdot 2^L + 1) \cdot \epsilon) \geq M\epsilon$$

(i.e. if forced to guess, receiver prefers guessing according to their type versus guessing something else). Rewriting the above two equations, we get that the receiver will guess $g_\varnothing$ as long as $\epsilon > \frac{1}{(N+1)(M \cdot 2^L + 1)}$ and $\epsilon \leq \frac{1}{M(2^L+1)+1}$. Note that it is possible to completely identify the receiver's type after L queries. Consider the following (deterministic) signaling policy:

- Send message g_i when the state is ω_i, for $i \in [2^L]$

- Send message g_{i^*} when the state is $\omega_\varnothing$

Corollary D.6. *After making L queries, the above signaling policy is optimal if*

$$1 \cdot \mathbf{p}_{i^*}[\omega_{i^*}] - 1 \cdot \mathbf{p}_{i^*}[\omega_\varnothing] \geq 0 \text{ and } 1 \cdot (\frac{1}{2} \cdot 2^L + 1) \cdot \epsilon - 1 \cdot \epsilon \geq 0$$

or equivalently $\epsilon \leq \frac{1}{\frac{1}{2} \cdot 2^L + 1 + 1}$. *If* $\frac{1}{2} = \frac{1}{2}$ *and* $1 = 1$, *this condition simplifies to* $\epsilon \leq \frac{1}{2^{L-1}+2}$. *(We will see why this is a reasonable choice of* $\frac{1}{2}$ *and* 1 *later.)*

Corollary D.7. *After making* $L' < L$ *queries, the optimal signaling policy sets* $\sigma(m = \text{ all types guess } \omega_j|\omega_j) = 1$ *for* $j \in [2^L]$ *and* $\sigma(m = \text{ all types guess } \omega_j|\omega_\varnothing) = \frac{\frac{1}{2}}{1} = \frac{1}{2}$.

Proof. Under this setting, the BIC constraint becomes: $\forall i \in [2^{L'}]$, $m \in \mathcal{M}$,

$$\epsilon \cdot \sigma(m|\omega_\varnothing) \cdot u_R(\omega_\varnothing, m[i]) + (1 - (\frac{1}{2} \cdot 2^L + 1) \cdot \epsilon) \cdot \sigma(m|\omega_i) \cdot u_R(\omega_i, m[i]) + \sum_{j=1, j \neq i}^{2^L} \frac{1}{2} \cdot \epsilon \cdot \sigma(m|\omega_j) \cdot u_R(\omega_j, m[i]) \geq 0$$

We proceed on a case-by-case basis.

Case 1: $m[i] = g_i$ Under this setting, the BIC constraint simplifies to: $\forall i \in [2^{L'}]$, $m \in \{\mathcal{M} : m[i] = g_i\}$,

$$-\epsilon \cdot \sigma(m|\omega_\varnothing) \cdot 1 + (1 - (\frac{1}{2} \cdot 2^L + 1) \cdot \epsilon) \cdot \sigma(m|\omega_i) \cdot 1 + 0 \geq 0,$$

or equivalently $\sigma(m|\omega_\varnothing) \leq \frac{1/\epsilon - (\frac{1}{2} \cdot 2^L + 1)}{1} \cdot \sigma(m|\omega_i)$.

Case 2: $m[i] = g_j$ Under this setting, the BIC constraint simplifies to: $\forall i \in [2^{L'}]$, $m \in \{\mathcal{M} : m[i] = g_j\}$,

$$-\epsilon \cdot \sigma(m|\omega_\varnothing) \cdot 1 + \frac{1}{2} \cdot \epsilon \cdot \sigma(m|\omega_j) \cdot 1 \geq 0,$$

or equivalently $\sigma(m|\omega_\varnothing) \leq \frac{\frac{1}{2}}{1} \cdot \sigma(m|\omega_j)$. Note that $\frac{\frac{1}{2}}{1} \leq \frac{1/\epsilon - (\frac{1}{2} \cdot 2^L + 1)}{1}$ if $\epsilon \leq \frac{1}{\frac{1}{2} \cdot (2^L + 1) + 1}$. Observe that if $\frac{1}{2} = \frac{1}{2}$ and $1 = 1$, the conditions on epsilon simplify to

$$\frac{1}{2^L + 2} < \epsilon \leq \frac{1}{2^{L-1} + 2}$$

which is always satisfiable for some ϵ (e.g. $\epsilon = \frac{1}{2^{L-1}+2}$). Suppose for now that it is optimal for the sender to set $\sigma(m = \text{all types guess } \omega_j|\omega_j) = 1$, for $j \neq \varnothing$. We will show that this is without loss of generality later. This implies that all variables of the form $\sigma(m \neq \text{all types guess } \omega_j|\omega_j) = 0$. The only variables which remain to be set are those of the form $\sigma(m|\omega_\varnothing)$. By the constraints in the optimization (and the fact that we have already set $\sigma(m|\omega_j)$, $j \neq \varnothing$), we have that variables of the form $\sigma(m \neq \text{all types guess } \omega_j|\omega_\varnothing) = 0$. Finally, it remains to set the 2^L remaining variables $\sigma(m = \text{all types guess } \omega_j|\omega_\varnothing)$. Out of the two types of constraints, the one in Case 2 is the one which binds, and so it is optimal to set $\sigma(m = \text{all types guess } \omega_j|\omega_\varnothing) = \frac{\frac{1}{2}}{1}$.

To see why it is optimal for the sender to set $\sigma(m = \text{all types guess } \omega_j|\omega_j) = 1$ for $j \neq \varnothing$, note that the fact that $\sum_{m \in \mathcal{M}} \sigma(m|\omega_j) = 1$ implies that $\sum_{m \in \mathcal{M}} \sigma(m|\omega_\varnothing) \leq \frac{1}{2}$, which the signaling policy described above obtains with equality. $\square$

D.8 Proof of Corollary 5.8

Corollary 5.8. *Consider the setting of Example 5.6, but with 2^L additional queries of the form "Does $\mathbf{p}_{i^*} = \mathbf{p}_j$?", for every $j \in \{0, \dots, 2^L - 1\}$. Under this setting, Algorithm 5 requires 2^L queries to obtain the optimal expected sender utility, whereas the optimal querying policy only requires L queries.*

Proof. Now consider the same setting of Example 5.6, but now instead consider a set of $L + 2^L$ queries $\mathcal{Q} = \mathcal{Q}' \bigcup \mathcal{Q}''$, where the first L queries $\mathcal{Q}'$ reveal a specific bit of $\mathbf{p}_{i^*}$ and the remaining 2^L queries $\mathcal{Q}''$ are of the form "does $\mathbf{p}_{i^*} = \mathbf{p}_j$?", for each $j \in [2^L]$. Observe that at every time-step, the greedy algorithm will pick a query from Q'' (until there are only two receiver types remaining, when it is indifferent between a query in Q' and one in Q''). To see this, note that picking a query in Q'' will *always* increase the expected utility of the sender, but picking a query in Q' will *never* increase the expected utility of the sender until there are only two possible receiver types remaining. $\square$

Corollary D.8. *[Marginal gain of greedy] There exists a setting in which after L queries the optimal querying policy achieves expected utility 1, but the greedy algorithm achieves utility $\frac{(2^L - L) \cdot u + L}{2^L}$, where $u < 1$ is the expected utility of signaling according to the signaling policy of Corollary D.7.*

Proof. Since the greedy algorithm asks a query in $\mathcal{G}''$ at each time-step, they will be able to identify the correct receiver type with probability $\frac{L}{2^L}$ (in which case they get utility 1) and they will not be able to identify the correct receiver type with probability $\frac{2^L - L}{2^L}$ (in which case they will get expected utility u). $\square$

E Appendix for Section 6: Committing to Query

E.1 Proof of Theorem 6.5

Theorem 6.5. *Given any three signaling policies σ_1, σ_2, σ_3 where σ_1 (resp. σ_2, σ_3) is an arbitrary signaling policy which is Bayesian incentive-compatible for agents of type τ_1 (resp. τ_2, τ_3), there exists a Bayesian incentive-compatible state-informed querying policy which can implement σ_1, σ_2, and σ_3 simultaneously (according to Definition 6.4).*

Proof. Consider the binary setting and suppose there are three receiver types τ_1, τ_2, τ_3 such that $p_1 < p_2 < p_3$. (This is without loss of generality.) We consider the querying policy which makes the threshold query θ_{12} such that $p_1 < \theta_{12} < p_2$ if $\omega = 0$ and threshold query θ_{23} such that $p_2 < \theta_{23} < p_3$ if $\omega = 1$. Under this setting, the sender needs to commit to a distribution over signals for each possible "information partition" $I \in \{(\omega = 0, \{\tau_1\}), (\omega = 0, \{\tau_2, \tau_3\}), (\omega = 1, \{\tau_1, \tau_2\}), (\omega = 1, \{\tau_3\})\}$. When a particular receiver type τ_i is faced with a message m, they will take action $a = 1$ if and only if

$$\mathbb{E}_{\omega \sim \mathbf{p}_{\tau_i}}[u_S(\omega, 1)|m] \geq \mathbb{E}_{\omega \sim \mathbf{p}_{\tau_i}}[u_S(\omega, 0)|m]$$

or equivalently $q_i(m|\omega = 1) \cdot p_i \geq q_i(m|\omega = 0) \cdot (1 - p_i)$, where $q_i(m|\omega = \alpha)$ is the probability that receiver type τ_i is sent message m when the state is α.

Using the above formulation and partitioning scheme, we see that receiver type τ_1 takes action $a = 1$ when faced with message m if and only if

$$\sigma(m|\omega = 1, \{\tau_1, \tau_2\}) \cdot p_1 \geq \sigma(m|\omega = 0, \{\tau_1\}) \cdot (1 - p_1),$$

receiver type τ_2 takes action $a = 1$ when faced with message m if and only if

$$\sigma(m|\omega = 1, \{\tau_1, \tau_2\}) \cdot p_2 \geq \sigma(m|\omega = 0, \{\tau_2, \tau_3\}) \cdot (1 - p_2),$$

and receiver type τ_3 takes action $a = 1$ when faced with message m if and only if

$$\sigma(m|\omega = 1, \{\tau_3\}) \cdot p_3 \geq \sigma(m|\omega = 0, \{\tau_2, \tau_3\}) \cdot (1 - p_3).$$

Note that via Theorem 3.1, it is without loss of generality to signal using (at most) 2^3 different messages. When written in binary, a message $m = jk\ell$ sent by a BIC signaling policy has the interpretation that receivers of type τ_1 (resp. τ_2, τ_3) should take action $a = 1$ if and only if $j = 1$ (resp. $k = 1$, $\ell = 1$). The following constraints ensure that a signaling policy σ of this form is Bayesian incentive-compatible:

$$\begin{aligned}
&\sigma(m = 111|\omega = 1, \{\tau_1, \tau_2\}) \cdot p_1 \geq \sigma(m = 111|\omega = 0, \{\tau_1\}) \cdot (1 - p_1),\\
&\sigma(m = 110|\omega = 1, \{\tau_1, \tau_2\}) \cdot p_1 \geq \sigma(m = 110|\omega = 0, \{\tau_1\}) \cdot (1 - p_1),\\
&\sigma(m = 101|\omega = 1, \{\tau_1, \tau_2\}) \cdot p_1 \geq \sigma(m = 101|\omega = 0, \{\tau_1\}) \cdot (1 - p_1),\\
&\sigma(m = 100|\omega = 1, \{\tau_1, \tau_2\}) \cdot p_1 \geq \sigma(m = 100|\omega = 0, \{\tau_1\}) \cdot (1 - p_1),\\
&\sigma(m = 111|\omega = 1, \{\tau_1, \tau_2\}) \cdot p_2 \geq \sigma(m = 111|\omega = 0, \{\tau_2, \tau_3\}) \cdot (1 - p_2),\\
&\sigma(m = 110|\omega = 1, \{\tau_1, \tau_2\}) \cdot p_2 \geq \sigma(m = 110|\omega = 0, \{\tau_2, \tau_3\}) \cdot (1 - p_2),\\
&\sigma(m = 011|\omega = 1, \{\tau_1, \tau_2\}) \cdot p_2 \geq \sigma(m = 011|\omega = 0, \{\tau_2, \tau_3\}) \cdot (1 - p_2),\\
&\sigma(m = 010|\omega = 1, \{\tau_1, \tau_2\}) \cdot p_2 \geq \sigma(m = 010|\omega = 0, \{\tau_2, \tau_3\}) \cdot (1 - p_2),\\
&\sigma(m = 111|\omega = 1, \{\tau_3\}) \cdot p_3 \geq \sigma(m = 111|\omega = 0, \{\tau_2, \tau_3\}) \cdot (1 - p_3),\\
&\sigma(m = 101|\omega = 1, \{\tau_3\}) \cdot p_3 \geq \sigma(m = 101|\omega = 0, \{\tau_2, \tau_3\}) \cdot (1 - p_3),\\
&\sigma(m = 011|\omega = 1, \{\tau_3\}) \cdot p_3 \geq \sigma(m = 011|\omega = 0, \{\tau_2, \tau_3\}) \cdot (1 - p_3),\\
&\sigma(m = 001|\omega = 1, \{\tau_3\}) \cdot p_2 \geq \sigma(m = 001|\omega = 0, \{\tau_2, \tau_3\}) \cdot (1 - p_2),\\
&\sum_{m \in \mathcal{M}} \sigma(m|I) = 1, \quad \forall I
\end{aligned}$$

Implementing σ_1, σ_2, σ_3 (according to Definition 6.4) is equivalent to imposing the following additional constraints:

$$\begin{aligned}
\sigma_1(m = 1|\omega = 1) = \sigma(m = 111|\omega = 1, \{\tau_1, \tau_2\}) + \sigma(m = 110|\omega = 1, \{\tau_1, \tau_2\})\\
+ \sigma(m = 101|\omega = 1, \{\tau_1, \tau_2\}) + \sigma(m = 100|\omega = 1, \{\tau_1, \tau_2\})
\end{aligned}$$

$$\begin{aligned}
\sigma_1(m = 1|\omega = 0) = \sigma(m = 111|\omega = 0, \{\tau_1\}) + \sigma(m = 110|\omega = 0, \{\tau_1\})\\
+ \sigma(m = 101|\omega = 0, \{\tau_1\}) + \sigma(m = 100|\omega = 0, \{\tau_1\})
\end{aligned}$$

$$\begin{aligned}
\sigma_2(m = 1|\omega = 1) = \sigma(m = 111|\omega = 1, \{\tau_1, \tau_2\}) + \sigma(m = 110|\omega = 1, \{\tau_1, \tau_2\})\\
+ \sigma(m = 011|\omega = 1, \{\tau_1, \tau_2\}) + \sigma(m = 010|\omega = 1, \{\tau_1, \tau_2\})
\end{aligned}$$

$$\begin{aligned}
\sigma_2(m = 1|\omega = 0) = \sigma(m = 111|\omega = 0, \{\tau_2, \tau_3\}) + \sigma(m = 110|\omega = 0, \{\tau_2, \tau_3\})\\
+ \sigma(m = 011|\omega = 0, \{\tau_2, \tau_3\}) + \sigma(m = 010|\omega = 0, \{\tau_2, \tau_3\})
\end{aligned}$$

$$\begin{aligned}
\sigma_3(m = 1|\omega = 1) = \sigma(m = 111|\omega = 1, \{\tau_3\}) + \sigma(m = 101|\omega = 1, \{\tau_3\})\\
+ \sigma(m = 011|\omega = 1, \{\tau_3\}) + \sigma(m = 001|\omega = 1, \{\tau_3\})
\end{aligned}$$

$$\begin{aligned}
\sigma_3(m = 1|\omega = 0) = \sigma(m = 111|\omega = 0, \{\tau_2, \tau_3\}) + \sigma(m = 101|\omega = 0, \{\tau_2, \tau_3\})\\
+ \sigma(m = 011|\omega = 0, \{\tau_2, \tau_3\}) + \sigma(m = 001|\omega = 0, \{\tau_2, \tau_3\})
\end{aligned}$$

Rearranging terms, we get that

$$\begin{aligned}
\sigma(m = 100|\omega = 1, \{\tau_1, \tau_2\}) = \sigma_1(m = 1|\omega = 1) - \sigma(m = 111|\omega = 1, \{\tau_1, \tau_2\})\\
- \sigma(m = 110|\omega = 1, \{\tau_1, \tau_2\}) - \sigma(m = 101|\omega = 1, \{\tau_1, \tau_2\})
\end{aligned}$$

$$\begin{aligned}
\sigma(m = 100|\omega = 0, \{\tau_1\}) = \sigma_1(m = 1|\omega = 0) - \sigma(m = 111|\omega = 0, \{\tau_1\})\\
- \sigma(m = 110|\omega = 0, \{\tau_1\}) - \sigma(m = 101|\omega = 0, \{\tau_1\})
\end{aligned}$$

$$
\begin{aligned}
\sigma(m=010|\omega=1,\{\tau_1,\tau_2\}) &= \sigma_2(m=1|\omega=1)-\sigma(m=111|\omega=1,\{\tau_1,\tau_2\}) \\
&\quad -\sigma(m=110|\omega=1,\{\tau_1,\tau_2\})-\sigma(m=011|\omega=1,\{\tau_1,\tau_2\}) \\
\sigma(m=010|\omega=0,\{\tau_2,\tau_3\}) &= \sigma_2(m=1|\omega=0)-\sigma(m=111|\omega=0,\{\tau_2,\tau_3\}) \\
&\quad -\sigma(m=110|\omega=0,\{\tau_2,\tau_3\})-\sigma(m=011|\omega=0,\{\tau_2,\tau_3\}) \\
\sigma(m=001|\omega=1,\{\tau_3\}) &= \sigma_3(m=1|\omega=1)-\sigma(m=111|\omega=1,\{\tau_3\}) \\
&\quad -\sigma(m=101|\omega=1,\{\tau_3\})-\sigma(m=011|\omega=1,\{\tau_3\}) \\
\sigma(m=001|\omega=0,\{\tau_2,\tau_3\}) &= \sigma_3(m=1|\omega=0)-\sigma(m=111|\omega=0,\{\tau_2,\tau_3\}) \\
&\quad -\sigma(m=101|\omega=0,\{\tau_2,\tau_3\})-\sigma(m=011|\omega=0,\{\tau_2,\tau_3\})
\end{aligned}
$$

Plugging these equalities into the relevant BIC constraints, we get

$$
\begin{aligned}
&(\sigma_1(m=1|\omega=1)-\sigma(m=111|\omega=1,\{\tau_1,\tau_2\})-\sigma(m=110|\omega=1,\{\tau_1,\tau_2\}) \\
&-\sigma(m=101|\omega=1,\{\tau_1,\tau_2\}))\cdot p_1 \geq (\sigma_1(m=1|\omega=0)-\sigma(m=111|\omega=0,\{\tau_1\}) \\
&-\sigma(m=110|\omega=0,\{\tau_1\})-\sigma(m=101|\omega=0,\{\tau_1\}))\cdot(1-p_1)
\end{aligned}
$$

$$
\begin{aligned}
&(\sigma_2(m=1|\omega=1)-\sigma(m=111|\omega=1,\{\tau_1,\tau_2\})-\sigma(m=110|\omega=1,\{\tau_1,\tau_2\}) \\
&-\sigma(m=011|\omega=1,\{\tau_1,\tau_2\}))\cdot p_2 \geq (\sigma_2(m=1|\omega=0)-\sigma(m=111|\omega=0,\{\tau_2,\tau_3\}) \\
&-\sigma(m=110|\omega=0,\{\tau_2,\tau_3\})-\sigma(m=011|\omega=0,\{\tau_2,\tau_3\}))\cdot(1-p_2)
\end{aligned}
$$

$$
\begin{aligned}
&(\sigma_3(m=1|\omega=1)-\sigma(m=111|\omega=1,\{\tau_3\})-\sigma(m=101|\omega=1,\{\tau_3\}) \\
&-\sigma(m=011|\omega=1,\{\tau_3\}))\cdot p_3 \geq (\sigma_3(m=1|\omega=0)-\sigma(m=111|\omega=0,\{\tau_2,\tau_3\}) \\
&-\sigma(m=101|\omega=0,\{\tau_2,\tau_3\})-\sigma(m=011|\omega=0,\{\tau_2,\tau_3\}))\cdot(1-p_3)
\end{aligned}
$$

Observe that the above BIC constraints will be satisfied if we set

$$
\begin{aligned}
\sigma(m=111|\omega=0,\{\tau_1\}) &= \frac{p_1}{1-p_1}\cdot\sigma(m=111|\omega=1,\{\tau_1,\tau_2\}) \\
\sigma(m=110|\omega=0,\{\tau_1\}) &= \frac{p_1}{1-p_1}\cdot\sigma(m=110|\omega=1,\{\tau_1,\tau_2\}) \\
\sigma(m=101|\omega=0,\{\tau_1\}) &= \frac{p_1}{1-p_1}\cdot\sigma(m=101|\omega=1,\{\tau_1,\tau_2\}) \\
\sigma(m=111|\omega=0,\{\tau_2,\tau_3\}) &= \frac{p_2}{1-p_2}\cdot\sigma(m=111|\omega=1,\{\tau_1,\tau_2\}) \\
\sigma(m=110|\omega=0,\{\tau_2,\tau_3\}) &= \frac{p_2}{1-p_2}\cdot\sigma(m=110|\omega=1,\{\tau_1,\tau_2\}) \\
\sigma(m=011|\omega=0,\{\tau_2,\tau_3\}) &= \frac{p_2}{1-p_2}\cdot\sigma(m=011|\omega=1,\{\tau_1,\tau_2\}) \\
\sigma(m=111|\omega=1,\{\tau_3\}) &= \frac{1-p_3}{p_3}\cdot\sigma(m=111|\omega=0,\{\tau_2,\tau_3\}) \\
\sigma(m=101|\omega=1,\{\tau_3\}) &= \frac{1-p_3}{p_3}\cdot\sigma(m=101|\omega=0,\{\tau_2,\tau_3\}) \\
\sigma(m=011|\omega=1,\{\tau_3\}) &= \frac{1-p_3}{p_3}\cdot\sigma(m=011|\omega=0,\{\tau_2,\tau_3\}),
\end{aligned}
$$

which then implies that

$$
\begin{aligned}
\sigma(m=111|\omega=1,\{\tau_3\}) &= \frac{1-p_3}{p_3}\cdot\frac{p_2}{1-p_2}\cdot\sigma(m=111|\omega=1,\{\tau_1,\tau_2\}) \\
\sigma(m=011|\omega=1,\{\tau_3\}) &= \frac{1-p_3}{p_3}\cdot\frac{p_2}{1-p_2}\cdot\sigma(m=011|\omega=1,\{\tau_1,\tau_2\})
\end{aligned}
$$

That just leaves us with the remaining BIC constraints and the constraints of the form $\sum_{m\in\mathcal{M}}\sigma(m|I)=1$ for the four possible information partitions $I\in\{(\omega=1,\{\tau_1,\tau_2\}),(\omega=0,\{\tau_1\}),(\omega=0,\{\tau_2,\tau_3\}),(\omega=1,\{\tau_3\})\}$ and the following unaccounted for variables:

- All variables of the form $\sigma(m|\omega = 1, \{\tau_1, \tau_2\})$
- $\sigma(m = 011|\omega = 0, \{\tau_1\})$, $\sigma(m = 010|\omega = 0, \{\tau_1\})$, $\sigma(m = 001|\omega = 0, \{\tau_1\})$, $\sigma(m = 000|\omega = 0, \{\tau_1\})$
- $\sigma(m = 101|\omega = 0, \{\tau_2, \tau_3\})$, $\sigma(m = 100|\omega = 0, \{\tau_2, \tau_3\})$, $\sigma(m = 001|\omega = 0, \{\tau_2, \tau_3\})$, $\sigma(m = 000|\omega = 0, \{\tau_2, \tau_3\})$
- $\sigma(m = 110|\omega = 1, \{\tau_3\})$, $\sigma(m = 100|\omega = 1, \{\tau_3\})$, $\sigma(m = 010|\omega = 1, \{\tau_3\})$, $\sigma(m = 000|\omega = 1, \{\tau_3\})$

Plugging in terms to the four equations of the form $\sum_{m\in\mathcal{M}} \sigma(m|I) = 1$ and simplifying, we get that

$$\begin{aligned}&\sigma_1(m = 1|\omega = 1) + \sigma_2(m = 1|\omega = 1) - \sigma(m = 111|\omega = 1, \{\tau_1, \tau_2\})\\&- \sigma(m = 110|\omega = 1, \{\tau_1, \tau_2\}) + \sigma(m = 001|\omega = 1, \{\tau_1, \tau_2\}) + \sigma(m = 000|\omega = 1, \{\tau_1, \tau_2\}) = 1\end{aligned} \tag{2}$$

$$\begin{aligned}&\sigma_1(m = 1|\omega = 0) + \sigma(m = 011|\omega = 0, \{\tau_1\}) + \sigma(m = 010|\omega = 0, \{\tau_1\})\\&+ \sigma(m = 001|\omega = 0, \{\tau_1\}) + \sigma(m = 000|\omega = 0, \{\tau_1\}) = 1\end{aligned} \tag{3}$$

$$\begin{aligned}&\sigma_2(m = 1|\omega = 0) + \sigma_3(m = 1|\omega = 0) - \sigma(m = 111|\omega = 0, \{\tau_2, \tau_3\})\\&- \sigma(m = 011|\omega = 0, \{\tau_2, \tau_3\}) + \sigma(m = 100|\omega = 0, \{\tau_2, \tau_3\}) + \sigma(m = 000|\omega = 0, \{\tau_2, \tau_3\}) = 1\end{aligned} \tag{4}$$

$$\begin{aligned}&\sigma_3(m = 1|\omega = 1) + \sigma(m = 110|\omega = 1, \{\tau_3\}) + \sigma(m = 100|\omega = 1, \{\tau_3\})\\&+ \sigma(m = 010|\omega = 1, \{\tau_3\}) + \sigma(m = 000|\omega = 1, \{\tau_3\}) = 1\end{aligned} \tag{5}$$

Note that Equations 3 and 5 may be trivially satisfied, and contain no variables overlapping with Equations 2 and 4. Now writing everything in terms of the variables which are not yet accounted for, Equation 4 becomes

$$\begin{aligned}&\sigma_2(m = 1|\omega = 0) + \sigma_3(m = 1|\omega = 0) - \frac{p_2}{1-p_2} \cdot \sigma(m = 111|\omega = 1, \{\tau_1, \tau_2\})\\&- \frac{p_2}{1-p_2} \cdot \sigma(m = 011|\omega = 1, \{\tau_1, \tau_2\}) + \sigma(m = 100|\omega = 0, \{\tau_2, \tau_3\}) + \sigma(m = 000|\omega = 0, \{\tau_2, \tau_3\}) = 1\end{aligned}$$

Solving for $-\sigma(m = 111|\omega = 1, \{\tau_1, \tau_2\})$, we get

$$\begin{aligned}&- \sigma(m = 111|\omega = 1, \{\tau_1, \tau_2\}) = \frac{1-p_2}{p_2} - \frac{1-p_2}{p_2} \cdot \sigma_2(m = 1|\omega = 0) - \frac{1-p_2}{p_2} \cdot \sigma_3(m = 1|\omega = 0)\\&+ \sigma(m = 011|\omega = 1, \{\tau_1, \tau_2\}) - \frac{1-p_2}{p_2} \cdot \sigma(m = 100|\omega = 0, \{\tau_2, \tau_3\}) - \frac{1-p_2}{p_2} \cdot \sigma(m = 000|\omega = 0, \{\tau_2, \tau_3\})\end{aligned}$$

Next we combine Equations 2 and 4 by plugging our expression for $-\sigma(m = 111|\omega = 1, \{\tau_1, \tau_2\})$ into Equation 2:

$$\begin{aligned}&\sigma_1(m = 1|\omega = 1) + \sigma_2(m = 1|\omega = 1) + \frac{1-p_2}{p_2} - \frac{1-p_2}{p_2} \cdot \sigma_2(m = 1|\omega = 0)\\&- \frac{1-p_2}{p_2} \cdot \sigma_3(m = 1|\omega = 0) + \sigma(m = 011|\omega = 1, \{\tau_1, \tau_2\})\\&- \frac{1-p_2}{p_2} \cdot \sigma(m = 100|\omega = 0, \{\tau_2, \tau_3\}) - \frac{1-p_2}{p_2} \cdot \sigma(m = 000|\omega = 0, \{\tau_2, \tau_3\})\\&- \sigma(m = 110|\omega = 1, \{\tau_1, \tau_2\}) + \sigma(m = 001|\omega = 1, \{\tau_1, \tau_2\}) + \sigma(m = 000|\omega = 1, \{\tau_1, \tau_2\}) = 1\end{aligned}$$

Moving all constants to the same side, we obtain

$$
\begin{aligned}
&\sigma(m=011|\omega=1,\{\tau_1,\tau_2\}) - \frac{1-p_2}{p_2}\cdot\sigma(m=100|\omega=0,\{\tau_2,\tau_3\}) - \frac{1-p_2}{p_2}\cdot\sigma(m=000|\omega=0,\{\tau_2,\tau_3\}) \\
&-\sigma(m=110|\omega=1,\{\tau_1,\tau_2\}) + \sigma(m=001|\omega=1,\{\tau_1,\tau_2\}) + \sigma(m=000|\omega=1,\{\tau_1,\tau_2\}) \\
&= 1 - \frac{1-p_2}{p_2} - \sigma_1(m=1|\omega=1) - \sigma_2(m=1|\omega=1) + \frac{1-p_2}{p_2}\cdot\sigma_2(m=1|\omega=0) \\
&+ \frac{1-p_2}{p_2}\cdot\sigma_3(m=1|\omega=0)
\end{aligned}
\tag{6}
$$

Note that by the BIC conditions,

$$-1 \le -c_2 := -\sigma_2(m=1|\omega=1) + \frac{1-p_2}{p_2}\cdot\sigma_2(m=1|\omega=0) \le 0.$$

Plugging this into Equation 6, our expression now becomes

$$
\begin{aligned}
&\sigma(m=011|\omega=1,\{\tau_1,\tau_2\}) - \frac{1-p_2}{p_2}\cdot\sigma(m=100|\omega=0,\{\tau_2,\tau_3\}) - \frac{1-p_2}{p_2}\cdot\sigma(m=000|\omega=0,\{\tau_2,\tau_3\}) \\
&-\sigma(m=110|\omega=1,\{\tau_1,\tau_2\}) + \sigma(m=001|\omega=1,\{\tau_1,\tau_2\}) + \sigma(m=000|\omega=1,\{\tau_1,\tau_2\}) \\
&= 1 - \frac{1-p_2}{p_2} - \sigma_1(m=1|\omega=1) - c_2 + \frac{1-p_2}{p_2}\cdot\sigma_3(m=1|\omega=0)
\end{aligned}
$$

We conclude by showing feasible solutions for both upper- and lower-bounds on the right hand side. It suffices to show this because if we can show that feasible solutions exist for both extremes, then we can take a convex combination of the two solutions to get a solution for any setting in between.

Upper bound The right-hand side may be upper-bounded as follows:

$$
\begin{aligned}
1 - \frac{1-p_2}{p_2} - \sigma_1(m=1|\omega=1) - c_2 + \frac{1-p_2}{p_2}\cdot\sigma_3(m=1|\omega=0) &\le 1 - \frac{1-p_2}{p_2} - 0 - 0 + \frac{1-p_2}{p_2} \\
&= 1
\end{aligned}
$$

Under this setting, a feasible solution clearly exists, e.g. by setting $\sigma(m=011|\omega=1,\{\tau_1,\tau_2\})=1$ and all other variables equal to zero.

Lower bound The right-hand side may be lower-bounded as follows:

$$
\begin{aligned}
1 - \frac{1-p_2}{p_2} - \sigma_1(m=1|\omega=1) - c_2 + \frac{1-p_2}{p_2}\cdot\sigma_3(m=1|\omega=0) &\ge 1 - \frac{1-p_2}{p_2} - 1 - 1 + 0 \\
&= -1 - \frac{1-p_2}{p_2}
\end{aligned}
$$

One feasible solution in this setting is to set $\sigma(m=110|\omega=1,\{\tau_1,\tau_2\})=1$, $\sigma(m=100|\omega=0,\{p_2,p_2\})=1$, and all other variables equal to zero. $\square$