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

November 2023

Abstract

How can an informed sender persuade a receiver, having only limited information about the receiver's beliefs? Motivated by research showing generative AI can simulate economic agents, we initiate the study of information design with an oracle. We assume the sender can learn more about the receiver by querying this oracle, e.g., by simulating the receiver's behavior. Aside from AI motivations such as general-purpose Large Language Models (LLMs) and problem-specific machine learning models, alternate motivations include customer surveys and querying a small pool of live users.

Specifically, we study Bayesian Persuasion where the sender has a second-order prior over the receiver's beliefs. After a fixed number of queries to an oracle to refine this prior, the sender commits to an information structure. Upon receiving the message, the receiver takes a payoff-relevant action maximizing her expected utility given her posterior beliefs. We design polynomial-time querying algorithms that optimize the sender's expected utility in this Bayesian Persuasion game. As a technical contribution, we show that queries form partitions of the space of receiver beliefs that can be used to quantify the sender's knowledge.

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

Contents

1 Introduction

Information design[1] is a canonical branch of theoretical economics which analyzes how provision of information influences strategic behavior. A typical framing involves a game with uncertain payoffs and an information designer who can commit to a particular information structure for the participants of this game. The designer chooses among feasible information structures so as to maximize her utility. While more idealized formulations assume that participant beliefs and preferences are fully known to the designer, in reality they often are not.

We initiate the study of information design with an *agent-informed oracle*: an oracle endowed with information about the participant and available to the information designer. The designer can query this oracle at a cost to supplement its incomplete knowledge about the participants. The query structure of the oracle can vary from application to application, but generically it enables the designer to refine her beliefs about the participant's beliefs. A salient case is a simulation oracle: an oracle that can simulate participants' strategic behavior in the game being studied, inducing a certain structure on refinements of the participant belief space. Our high-level goal is to explore how access to this additional source of information can help the principal optimize the information structure.

We focus on Bayesian Persuasion (BP) [KG11], a paradigmatic setting in information design. BP is a game of asymmetric information between two players: an informed *sender* (corresponding to the designer in information design) and an uninformed *receiver* (corresponding to an agent in information design). The sender observes a payoff-relevant *state* of the world and sends a *message* to the receiver, who then chooses an *action* which yields payoffs for both players. The sender commits to a messaging policy before the state is revealed, and the receiver takes an action based on the realized message. We extend this game to allow the sender to query a receiver-informed oracle before she commits to a messaging policy.

Motivation. Agent-informed oracles are motivated by recent advances in Generative AI, particularly Large Language Models (LLMs), e.g., [Ope23, ADF$^+$23, TMS$^+$23]. LLMs exhibit an emergent capability to simulate strategic behavior of economic agents. Notably, LLMs have been shown to make consumer choices that track with those of humans. They exhibit downward-sloping demand curves, diminishing marginal utility of wealth, and state dependence, and further match the stated willingness-to-pay of consumers in a recent market survey [BIN23]. They even exhibit (sometimes non-strategic) behaviors consistent with particular demographics given appropriate framing, such as libertarians in the context of pricing [Hor23] or women in the context of ultimatum games [AAK23]. Recent work also shows they can infer demographics and even user personality traits from online telemetry data such as Facebook status updates [PM23]. Based on these experiments, and the common belief that the capabilities of LLMs will continue to improve, we argue they will be increasingly used as agent-informed oracles in information design settings.

An agent-informed oracle can be implemented with access to more traditional AI as well. One could train a specialized machine learning model using domain-specific data, and use this model to predict strategic behavior in a particular domain. Moreover, strategic parties are increasingly implemented via AI agents, e.g., in online markets, ad auctions, and gaming, in which case it may be possible to simulate them directly through access to (or knowledge of) the AI agent. In each of these cases, the sender must incur some cost to query the oracle.[2]

An agent-informed oracle can also be a metaphor for market research that the sender performs before interacting with the receiver. For example, a startup may test out its funding pitch on smaller

[1] See [BM19] for a survey of this research area.

[2] For example, query costs for LLMs can be substantial, in terms of money and/or delay. In fact, the latest LLMs can be very expensive, as of this writing, if a large volume of queries is needed [J23].

venture capital firms before trying to persuade a larger firm to fund their business. A company may run a customer survey before bringing a new product or service to market, or experiment on a fraction of users in online services [e.g., KLSH09, KL17]. Again, the "queries" in these settings are expensive; a startup may have a limited number of venture capital firms that it can pitch to, and a company may be limited in the number of customers/users it can survey without disrupting overall sales.

As a specific example of BP, consider the interaction between a seller (the sender) of a product with a fixed price and a potential buyer (the receiver). Here, the state of the world is the quality of the product (e.g., high/low quality) and the message corresponds to the sales pitch presented to the buyer. The seller would always like to sell, but the buyer only wants to buy if the product is of sufficiently high quality. A buyer-informed oracle may help the seller optimize her sales pitch. In BP, the seller must commit to this sales pitch before observing the product quality; this ability to commit may arise from, e.g. legal regulations or the seller's desire to protect her reputation. This is a rather general scenario: the "product" may refer to a physical good as well as any service, opportunity, or experience; the "quality" may be absolute and/or buyer-specific; the "sales pitch" encompasses any information presented to the buyer.

A buyer-informed oracle is particularly salient in the context of an online platform that interacts with many potential buyers. Economies of scale may enable a sophisticated automatic workflow with an AI-based oracle, particularly algorithms that choose how to query the oracle and how to use its output to shape buyer-facing communication. This communication can potentially be customized to a particular product and a particular customer. The oracle itself may be either general-purpose or domain-specific, and provided by either a third party or developed within the platform.

Our model. We study Bayesian Persuasion (BP) with a receiver-informed oracle. A problem instance consists of three parts: a BP instance, an oracle, and a cost model.

We introduce a version of BP in which the sender is endowed with partial knowledge of receiver's belief, as expressed by a second-order prior, and full knowledge of the utility structure. The state is drawn from the receiver's belief, which is in turn drawn from the sender's second-order prior. This is a natural midpoint between two extremes studied in prior work. The traditional formulation of BP assumes a known utility structure and a common prior on the state, so the receiver is fully known. In contrast, in *robust* BP [e.g., DP22, PS22, HW21] the receiver is almost fully *not* known.

The sender can query the oracle to gain additional information about the receiver's belief.[3] A generic oracle takes some input (a query) and deterministically maps it to some output, where the output is determined by the input and the true (receiver) belief/prior. The set of all possible inputs and the mapping are known to the sender. Equivalently, the oracle inputs a partition of the *belief space* (the set of feasible beliefs of the receiver) and returns the subset of this partition which contains the receiver's true belief. We refer to this generic class of oracles as *partition oracles*, and characterize it by the set of all feasible partitions of belief space. A paradigmatic special case is a *simulation oracle*, which simulates the receiver's response in the same BP instance. Such an oracle inputs a messaging policy and a particular message realization, and outputs the action that would have been chosen by the receiver with true belief upon seeing this policy and this message. This special case is strongly motivated by the applications discussed previously.

Before the BP game starts, the sender (i) queries the oracle according to some *querying policy* and (ii) computes a messaging policy using the information gained from the oracle. The sender strives to optimize her Bayesian-expected utility, given the cost model for the oracle. The main cost model we consider is a *query budget*: a fixed number of queries that may be posed to the oracle.

[3] Of course, oracle queries may also reveal information about the state, since the state is drawn from the receiver's belief. However, crucially, the oracle cannot reveal anything about the state that the receiver does not already know. The power to commit to a messaging policy before the state is revealed is therefore still valuable to the sender.

Our results also extend to per-query costs that are subtracted from sender's utility; these costs are known to the sender *a priori*, and may be different for different queries.

Our results. From an algorithmic perspective, our goal is to implement steps (i,ii) above so as to optimize the sender's Bayesian-expected utility. Our technical analysis is mainly concerned with the problem of computing an optimal querying policy for step (i), assuming that an optimal messaging policy is chosen in step (ii). For the cases when we have an efficient algorithm for this problem, optimizing the messaging policy in step (ii) can be done efficiently without much difficulty.

We distinguish between *adaptive* querying policies that can choose the next query after seeing the output of the previous one and *non-adaptive* querying policies which must make all queries up front. We show that an optimal adaptive querying policy, for any given partition oracle, can be found via backward induction in time linear in the (worst-case) size of such policy. While the worst-case size of the policy can be exponential in the query budget, naive brute-force search takes doubly-exponential time to compute an optimal adaptive querying policy.

In contrast, computing an optimal non-adaptive querying policy (i.e., an optimal subset of queries) is NP-Complete for partition oracles. In fact, this holds even for a canonical "binary" case of BP when there are only 2 states and 2 actions (*Binary BP*), e.g., high/low quality and buy-or-not in the buyer-seller BP example. Even the greedy algorithm, a natural heuristic often used for approximation, provably fails in the general setting, although, as implied by the diminishing value of queries discussed below, it is approximately optimal in the Binary BP setting. On the other hand, we provide a polynomial-time algorithm for Binary BP with a simulation oracle. We find that oracle queries are totally ordered, and we leverage this ordering to compute an optimal querying policy via dynamic programming.

We also provide structural insights. First, we are interested in the diminishing returns property in terms of the number of queries the sender makes to the oracle.[4] We establish this property for Binary BP with a simulation oracle. However, this property does not hold in general: we provide a counterexample with a specific (non-binary) BP instance and a specific partition oracle. Second, we consider the effect of allowing the sender to *commit* to a particular querying policy. We find that this additional commitment power cannot hurt the sender, and can sometimes significantly benefit her utility.

1.1 Related Work

Bayesian Persuasion (BP) was introduced by [KG11], and has been extensively studied since then, see [Kam19] for a recent survey. The most relevant direction is *robust BP*, which aims to relax the assumptions on the information the sender has about the receiver [DP22, HW21, PS22, Kos22, ZIX21]. This line of work typically focuses on characterizing the "minimax" messaging policy (i.e., one that is worst-case optimal over the sender's uncertainty), while our focus is on using oracle queries to help the sender overcome her uncertainty. Our work is also related to *online BP* [CCMG20, CMCG21, BCC+23, ZIX21], where the sender interacts with a sequence of receivers. In prior work on this variant, the sequence of receivers is adversarially chosen, and the sender minimizes regret. Our model (with a simulation oracle) can be interpreted as a "pure exploration" variant of online BP.[5] Indeed, all K oracle calls simulate the same "real" receiver, and are provided for free.

Large Language Models (LLMs) in economics. A rapidly growing line of work at the intersection of computer science and economics explores the use of LLMs in various economic contexts. Aside from LLM-simulated economic agents discussed in Section 1, this line of work

[4] In other words, whether the marginal utility of adding one more oracle query decreases with the query budget.

[5] By analogy with "pure exploration" in multi-armed bandits [MT04, EMM06, BMS11, ABM10], where an algorithm explores for K rounds in a stationary environment, and then predicts the best action.

studies LLM-simulated (human-driven) experiments in behavioral economics [Hor23], LLM-generated persuasive messages [MTV+23], LLM-simulated human day-to-day behavior [POC+23], LLM-predicted opinions for nationally representative surveys [KL23], and auction mechanisms to combine LLM outputs [DMPL+23]. A growing body of work uses LLMs to make strategic decisions in various scenarios [LH23, ASCF+23, BD23, CLSZ23, Guo23, Tsu23]. Finally, simultaneous work [FGP+23] adopts a conceptually similar approach with LLM-based oracles, focusing on *social choice*. Their framework combines social choice theory with LLMs' ability to generate unforeseen alternatives and extrapolate preferences.

Simulation in Games. [KOC23] study a normal-form game setting in which one player can simulate the behavior of the other. In contrast, we study simulation in Bayesian Persuasion games, which are a type of *Stackelberg* game [VS10, CS06]. There is a line of work on learning the optimal strategy to commit to in Stackelberg games from query access [LCM09, PSTZ19, BHP14, BBHP15]. 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 *messaging policy* which specifies a probability distribution over actions for every possible state realization.

2 Model and Preliminaries

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 persuasion setting, 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. The sender commits to a messaging policy which maps states to messages.

Definition 2.1 (Messaging Policy)**.** *The sender's* messaging 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 will sometimes write $\sigma(m|\omega)$ to denote* $\Pr[\sigma(\omega) = m]$.

We use $\mathcal{M} \ni m \sim \sigma(\omega)$ to denote a message sampled from σ when the state is $\omega \in \Omega$. The timing of the game is as follows: 1) the sender commits to a messaging policy σ; 2) the state ω is revealed to the sender; 3) the message $m \sim \sigma(\omega)$ is sent to the receiver; 4) the receiver chooses an action $a \in \mathcal{A}$.

When the sender and receiver share a common prior $\mathbf{p}$ it is without loss of generality to message according to the set of possible actions (i.e. $\mathcal{M} = \mathcal{A}$) via a revelation principle-style argument. (See [KG11] 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 messaging policy is *Bayesian incentive-compatible* (BIC) if it is always in the receiver's best interest to follow the sender's action recommendations.

Definition 2.2 (Bayesian Incentive-Compatibility)**.** *A messaging policy $\sigma : \Omega \to \mathcal{A}$ is Bayesian incentive-compatible if for every action $a \in \mathcal{A}$, $\mathbb{E}_{\omega\sim\mathbf{p}}[u_R(\omega, m)|m] \geq \mathbb{E}_{\omega\sim\mathbf{p}}[u_R(\omega, a)|m]$. We assume that if the receiver is indifferent between two actions, she breaks ties in favor of the sender's recommended action.*

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

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

The time it takes to compute the optimal messaging 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. [PS22, Kos22, HW21, KMZL17] for various versions of this setting studied in the literature).

Definition 2.3 (Binary Bayesian Persuasion). *In the Binary Bayesian Persuasion 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 $\omega = 1$.*

Receiver-Informed Oracle. We are now ready to introduce the receiver-informed oracle. 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 corresponding *belief* $\mathbf{p}_\tau$ over Ω. We refer to the set $\{\mathbf{p}_\tau | \tau \in \mathcal{T}\}$ as the receiver *belief space*. 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 in the sense that her beliefs are correct). We denote the sender's prior over receiver types (alternatively, over receiver beliefs) by $\mathcal{P}(\mathcal{T})$, and assume that $\tau^* \sim \mathcal{P}$. 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 belief. An oracle is characterized by the set of allowable *queries* $\mathcal{Q}$ with $Q := |\mathcal{Q}| < \infty$, that the sender can pose. A particular query can be represented via a deterministic mapping from beliefs to the possible answers. To abstract away from details of the query implementation, we adopt an equivalent definition in terms of how the possible answers to the query partition the belief space.

Definition 2.4 (Partition Oracles and Partition Queries). *A partition query $q \in \mathcal{Q}$ is a partition of the receiver belief space into a set of n_q non-overlapping subsets $\{s_{q,i}\}_{i=1}^{n_q}$ such that $\cup_{i=1}^{n} s_{q,i} = \{\mathbf{p}_\tau | \tau \in \mathcal{T}\}$. We use $i(\tau)$ to denote the index of the subset for which receiver type τ belongs, i.e. $\mathbf{p}_\tau \in s_{q,i(\tau)}$. A partition oracle with query space $\mathcal{Q}$ takes a partition query $q \in \mathcal{Q}$ as input, and returns the subset $s_{q,i(\tau^*)}$.*

After making a partition query, the sender can leverage the fact that the true receiver belief is contained in the returned subset to design more persuasive messaging policies. An important special case is when the sender can *simulate* the response of the receiver.

Definition 2.5 (Simulation Oracles and Simulation Queries). *A simulation query is a tuple $q := (\sigma_q, m_q)$, where $\sigma_q : \Omega \to \mathcal{A}$ is a messaging policy mapping states to actions, and $m_q \in \mathcal{A}$ is a message. A simulation oracle takes a simulation query q as input, and returns the (true) receiver type τ^*'s best-response $a_{q,\tau^*} = \arg\max_{a \in \mathcal{A}} \mathbb{E}_{\omega \sim \mathbf{p}_{\tau^*}}[u_R(\omega, a) | m_q]$.*

A simulation oracle is a partition oracle with a restricted set of allowable queries $\mathcal{Q} \subset \mathcal{S}$, where $\mathcal{S}$ is the power set of receiver beliefs $\{\mathbf{p}_\tau | \tau \in \mathcal{T}\}$. The structure of a simulation oracle's query space is discussed in Theorem 3.4. 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 $q \in \mathcal{Q}$ has an associated cost $c_q \in \mathbb{R}$. Under this *costly* setting the sender's goal is to maximize her Bayesian-expected utility, *minus the total cost of querying*. We posit that the sender does not observe the state until *after* the messaging policy is computed. We relax this assumption in Section 6.

Protocol: Bayesian Persuasion with Oracle Queries

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

Figure 1: Summary of our setting.

The sender interacts with the oracle according to a *querying policy*, as defined below. Two natural interaction regimes are *adaptive* and *non-adaptive*. An adaptive querying policy makes a query, observes the realized partition of the belief space, and repeats. A non-adaptive querying policy specifies all K queries *up front*, before observing the oracle's response to any of them.

Definition 2.6 (Querying Policy)**.** *An* adaptive *querying policy* $\pi : \mathcal{S} \times \mathbb{N} \to \mathcal{Q}$ *is a mapping from belief subsets and number of queries left to queries. A* non-adaptive *querying policy* $\pi : \mathbb{N} \to \mathcal{Q}^*$ *is a mapping from a number of queries to a set of queries.*

The timing of the game is summarized in Figure 1. The sender's goal is to compute a querying policy (either adaptive or non-adaptive, depending on the setting) that will maximize the expected sender utility obtained from the optimal messaging policy for the (endogenous) prior over receiver beliefs, i.e., the sender's objective is $\max_\pi \mathbb{E}_{\tau \sim \mathcal{P}_\pi(\mathcal{T})} \left[\max_{\sigma_{s(\tau)}} \mathbb{E}_{\omega \sim p_\tau, m \sim \sigma_{s(\tau)}(\omega)} [u_S(\omega, a(m))] \right]$, where $P_\pi(\mathcal{T})$ is the sender's expected posterior belief over the type given the querying policy π, $s(\tau)$ is the belief partition for which receiver type τ belongs after querying, and $a(m)$ is the action that maximizes the receiver's expected utility over her posterior given messaging policy σ and message m.

3 How Should We Think About Querying?

We begin with structural results which showcase intuition and will be useful later. We derive the sender's optimal messaging policy when there is uncertainty about the receiver's belief. Then we discuss geometric properties of simulation queries, illustrating how they benefit the sender.

3.1 Optimal Persuasion Under Uncertainty

How should the sender send messages whenever she is uncertain about the receiver's belief? Recall that when the receiver's prior is known, the sender can send messages according to the set of receiver actions (i.e. $\mathcal{M} = \mathcal{A}$). When the sender has uncertainty about the receiver, A^T messages suffice in the general setting and $T+1$ messages suffice in Binary BP.

Proposition 3.1. *[Revelation Principle for Unknown Receiver Beliefs] It always suffices for the sender to use a message space of size $M = A^T$. In Binary BP, it suffices to use $M = T+1$ messages.*

We can now characterize the sender's optimal messaging policy whenever she has uncertainty about the receiver's belief, both in the general setting (Corollary 3.2) and in Binary BP (Corollary 3.3). The optimal messaging policy in the general setting follows readily from the above revelation principle for unknown receiver beliefs.

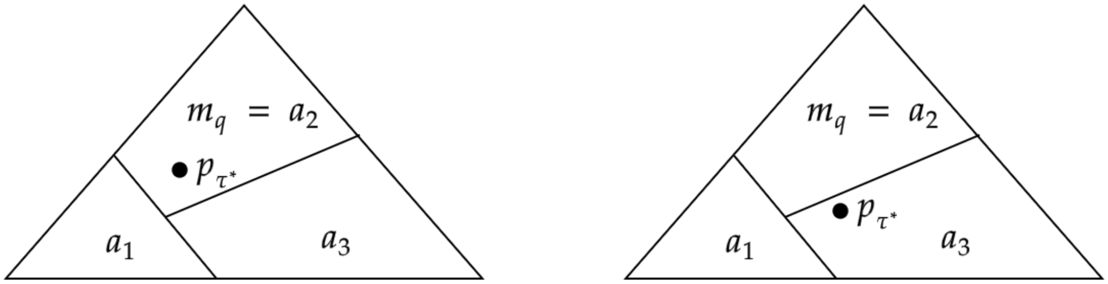

Figure 2: Visualization of receiver best-response regions of Δ^d for simulation query $q = (\sigma_q, m_q)$. On the left, the receiver follows the sender's recommendation, and we can infer that her belief $\mathbf{p}_{\tau^*}$ is contained in the best-response region for a_2. On the right, the receiver does not follow the sender's recommendation, and we can instead infer that her belief is in the best-response region for a_3.

Corollary 3.2 (Optimal Messaging Policy)**.** *For second-order prior $\mathcal{P}$, the sender's optimal messaging policy $\sigma^*_{\mathcal{P}}$ is given by the solution to the following linear program:*

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

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 $\mathcal{O}(\mathrm{LP}(dA^{T'}, T'A^{T'+1} + dA^{T'}))$.

The form of the sender's optimal messaging policy may be further simplified in Binary BP. In particular, the optimal messaging policy in Binary BP has the following interpretation: when receiving message $m_{i'}$, all receivers with belief p such that $p \leq p_{i'}$ should take action $a = 1$. In order to simplify the exposition, we assume that $p_\tau \leq \frac{1}{2}$ for every receiver type $\tau \in \mathcal{T}$.[6]

Corollary 3.3. *[Optimal Messaging Policy; Binary BP] In Binary BP, 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 messaging policy can be computed in time $\mathcal{O}((T')^2)$. The optimal policy is given by $\sigma(m_{i^*}|1) = 1$, $\sigma(m_{i^*}|0) = \frac{p_{i^*}}{1-p_{i^*}}$, and $\sigma(0|0) = 1 - \frac{p_{i^*}}{1-p_{i^*}}$, where $i^* := \arg\max_{i'\in[L,H]} \sum_{i=L}^{i'} \mathcal{P}(\tau_i) \cdot \left(p_i + (1-p_i)\cdot\frac{p_{i'}}{1-p_{i'}}\right)$.*

3.2 How Simulation Helps

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 the oracle's response to a simulation query:

Proposition 3.4. *[Relationship between simulation queries and separation oracles] Let $\alpha(\mathbf{p}) := \sum_{\omega\in\Omega}(u_R(\omega, m_q) - u_R(\omega, a)) \cdot \sigma_q(m_q|\omega) \cdot \mathbf{p}[\omega]$. By making a simulation query $q = (\sigma_q, m_q)$, the sender is specifying a polytope $\mathcal{R}_q := \{\mathbf{p} \in \Delta^d\ :\ \alpha(\mathbf{p}) \geq 0, \forall a \in \mathcal{A}\}$ and the oracle returns either (i) $\mathbf{p}_{\tau^*} \in \mathcal{R}_q$ or (ii) $\mathbf{p}_{\tau^*} \notin \mathcal{R}_q$ and for some $a' \in \mathcal{A}$ for all $\mathbf{p}' \in \mathcal{R}_q$, $\alpha(\mathbf{p}' - \mathbf{p}_{\tau^*}) > 0$.*

[6]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.

Algorithm 1 Computing the Optimal Adaptive Querying Policy: K Queries

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

- Set $V[\mathcal{T}', 0] := \mathbb{E}_{\tau \sim \mathcal{P}(\mathcal{T}')} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma^*_{\mathcal{P}(\mathcal{T}')}(\omega)} u_S(\omega, m)$ for all feasible receiver subsets $\mathcal{T}' \in \mathcal{F}_\mathcal{T}$, where $\sigma^*_{\mathcal{P}(\mathcal{T}')}$ is the optimal messaging policy for receiver subset $\mathcal{T}'$, as defined in Corollary 3.2.
- For every $1 \leq k \leq K$ and all $\mathcal{T}' \in \mathcal{F}_\mathcal{T}$, 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 receiver subset $\mathcal{T}'$ with k queries left.

See Figure 2 for a visualization of how simulation queries can help reduce the sender's uncertainty about the receiver's belief. An immediate corollary of Proposition 3.4 is that in Binary BP, 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 Binary BP, $a_{q,\tau^*} = m_q$ 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 the state ω equaling the recommended action 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 in Binary BP, there always exists a simulation query q which can distinguish between any two receiver beliefs p_τ, $p_{\tau'}$ such that $p_\tau \neq p_{\tau'}$. Similar intuition carries over to the general setting, although it may be possible to distinguish between three or more beliefs using a *single* simulation query (unlike in Binary BP). The following is an example of a setting in which it is possible to distinguish between up to d different beliefs using a single simulation query.

Example 3.6. *Suppose that there are d states and d actions, where $u_R(\omega, a) = \mathbb{1}\{\omega = a\}$. Consider d receiver beliefs $\mathbf{p}_1, \ldots, \mathbf{p}_d$ 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 $j \neq 1$, $\sigma(a_1|\omega_j) = 2\sigma(a_1|\omega_1)$.*

Since there are a finite number of receiver types, it suffices to consider a finite set of possible simulation queries without loss of generality. In the sequel, our goal is to determine the optimal adaptive/non-adaptive querying policy, given a set of valid queries as input.

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 allowable partition 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 she messages according to the optimal messaging policy for the realized belief subset.

Definition 4.1 (Feasible Belief Subset)**.** *For a fixed query budget K, we say that a subset of receiver beliefs $\{\mathbf{p}_\tau | \tau \in \mathcal{T}'\}$ for $\mathcal{T}' \subseteq \mathcal{T}$ is* feasible *if there exists some $\tau \in \mathcal{T}$ and $\mathcal{Q}' \subseteq \mathcal{Q}$ with $|\mathcal{Q}'| \leq K$ such that $\{\mathbf{p}_\tau | \tau \in \mathcal{T}'\} = \bigcap_{q \in \mathcal{Q}'} s_{q,i(\tau)}$. We call $\mathcal{T}'$ the corresponding feasible receiver subset. We denote the set of feasible belief subsets (resp. receiver subsets) by $\mathcal{F}$ (resp. $\mathcal{F}_\mathcal{T}$) and the number of feasible subsets by $F := |\mathcal{F}| \leq 2^K$.*

An adaptive querying policy needs to specify a query to make for each of the F feasible belief subsets. Observe that the sender's optimal querying policy starting from receiver subset $\mathcal{T}'$ is

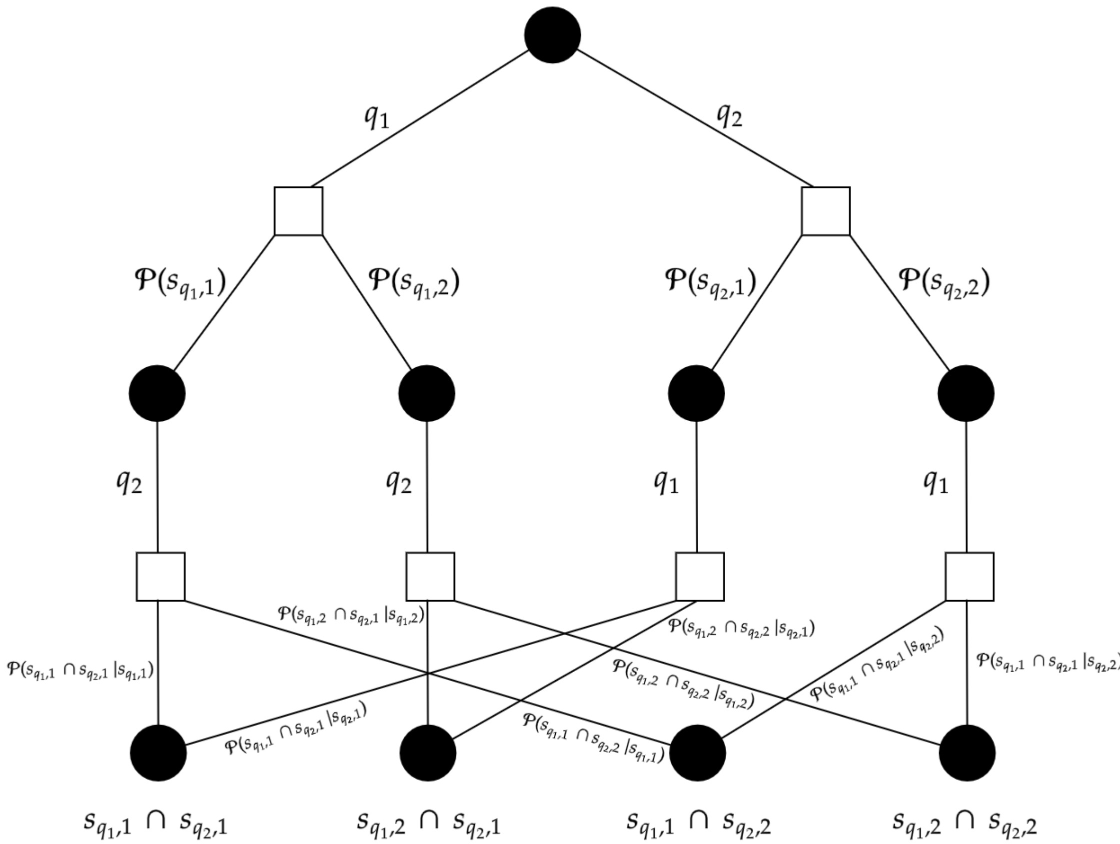

Figure 3: Game tree between sender and nature for a setting with two queries q_1 and q_2. Each query results in two feasible subsets, with query q_1 (resp. q_2) resulting in subsets $s_{q_1,1}$ and $s_{q_1,2}$ (resp. $s_{q_2,1}$ and $s_{q_2,2}$). Sender actions (black circles) are queries to the oracle,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

independent of any queries used previously to refine the subset, but may need to anticipate future queries. For example, consider an instance of Binary BP with four possible receiver types and $K = 2$ simulation queries. In this scenario, it will always be optimal to use the first query to separate the smallest two receiver beliefs from the largest two beliefs, *regardless of the immediate utility gain from doing so or any other parameters of the problem instance*, since this allows the second query to fully separate all receiver beliefs. Any other initial query is always strictly suboptimal.

Motivated by the necessity of anticipating future queries, we show that the sender's optimal querying policy may be computed via backward induction on an appropriately-constructed extensive-form game tree. We start by considering the expected utility associated with the optimal messaging policy for each belief subset whenever the sender has no queries remaining. Using these expected utilities as a building block, we can compute the sender's optimal querying policy when she has only one query. We then use our solution for one query to construct the optimal querying policy whenever the sender has two queries, and we continue iterating until we have the optimal adaptive querying policy for K queries. This process is illustrated in Figure 3 and formally described in Algorithm 1.

Theorem 4.2. *Algorithm 1 computes the sender's optimal adaptive querying policy for K partition*

Algorithm 2 Computing the Optimal Non-Adaptive Querying Policy: Binary BP, K Queries

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

- Set $V[i, j, 0] := \sum_{l=q}^{i} \mathcal{P}(\tau_l) \cdot \mathbb{E}_{\tau \sim \mathcal{P}(i,j)} \mathbb{E}_{\omega \sim \mathbf{p}_\tau} \mathbb{E}_{m \sim \sigma^*_{\mathcal{P}(i,j)}(\omega)} u_S(\omega, m)$ for all $1 \leq i \leq j \leq T$, where $\sigma^*_{\mathcal{P}(i,j)}$ is the optimal messaging policy of Corollary 3.3 under second-order prior $\mathcal{P}(i, j)$, and i, j is shorthand for types $\{i, \ldots, j\}$.
- For every $1 \leq k \leq K$ and $1 \leq j \leq T$, compute $V[1, j, k] := \max_{q \in \mathcal{Q}} V[q+1, j, 0] + V[1, q, k-1]$
- The optimal policy then makes the K queries which obtain value $V[1, T, K]$.

queries in $\mathcal{O}(FKQT)$ time. In Binary BP with simulation queries, the runtime is $\mathcal{O}(T^4K)$.

Note that computing the optimal adaptive policy via brute-force search requires considering $\mathcal{O}(Q^F)$ policies and since $F \leq 2^K$, Algorithm 1 allows us to go from doubly-exponential-in-K to exponential-in-K time in the worst case. In order to separate the time it takes to compute the optimal *querying* policy from any time required to compute the sender's *messaging* policy, we assume that the optimal messaging policies for each feasible belief subset are known to the sender before Algorithm 1 is run. Alternatively, one could view the computation of the optimal messaging policies as part of Algorithm 1, in which case the runtime would contain an additional additive factor proportional to the time it takes to compute the optimal messaging policy under uncertainty (see Corollary 3.2). In Appendix C.2, we show how our results extend to the costly query setting.

5 Non-Adaptive Querying Policies

When querying non-adaptively, the sender must make all K queries *up front* (recall Definition 2.6). Our results in this setting differ significantly between Binary BP with simulation queries and the general setting. The following notion of a *non-adaptive belief partition* will be useful later on.

Definition 5.1 (Non-Adaptive Belief Partition)**.** *A set of non-adaptive queries $\mathcal{Q}' \subseteq \mathcal{Q}$ induces a partition $H_{\mathcal{Q}'}$ over receiver belief space such that $\bigcup_{\eta \in H_{\mathcal{Q}'}} \eta = \{\mathbf{p}_\tau | \tau \in \mathcal{T}\}$. Receiver belief $\mathbf{p}_\tau$ belongs to the subset $\eta_{\mathcal{Q}'}[\tau] \in H_{\mathcal{Q}'}$ such that $\eta_{\mathcal{Q}'}[\tau] := \bigcap_{q \in \mathcal{Q}'} s_{q,i(\tau)}$.*

5.1 Computing Non-Adaptive Querying Policies

We find that the optimal non-adaptive querying policy may be computed via dynamic programming when considering simulation queries in Binary BP. However unlike in the adaptive setting, the optimal querying policy is computed by leveraging the fact that there is a "total ordering" over receiver beliefs when there are two states, and iteratively building solutions for larger sets of receivers. We overload notation and use q to index the receiver belief 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 $p_1 > p_2 > \cdots > p_T$, Algorithm 2 keeps track of the optimal sender utility achievable with k queries when in receiver subset $\{1, \ldots, i\}$ for all $1 \leq i \leq T$. 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 subset $\{1, \ldots, i+1\}$ as a function of the optimal solution for k queries in subset $\{1, \ldots, i\}$. See Figure 4 for a visualization of the sender's expected utility in Binary BP.

Theorem 5.2. *In Binary BP with simulation queries, Algorithm 2 computes the sender's optimal non-adaptive querying policy in $\mathcal{O}(T^3K)$ time.*

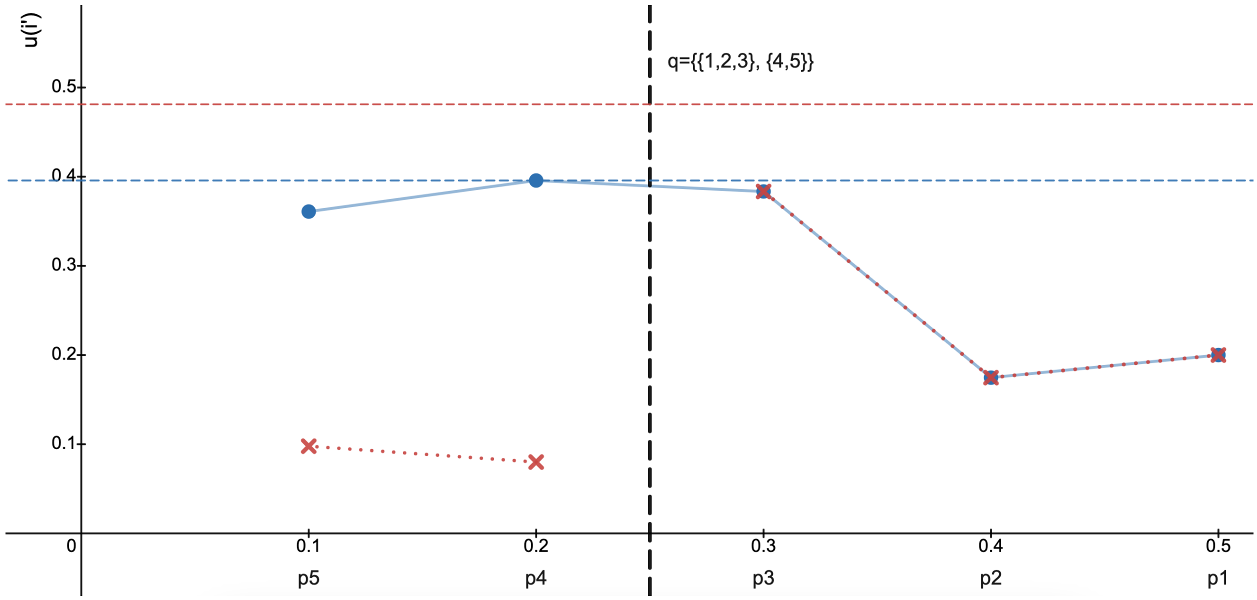

Figure 4: Sender's expected utility $u(i')$ (y-axis) as a function of the cutoff receiver index i' in Binary BP. Under this problem instance there are five types with beliefs $p_1 = 0.5$, $p_2 = 0.4$, $p_3 = 0.3$, $p_4 = 0.2$, and $p_5 = 0.1$ (x-axis), and second-order prior $\mathcal{P}(\tau_1) = 0.2$, $\mathcal{P}(\tau_2) = 0.01$, $\mathcal{P}(\tau_3) = 0.39$, $\mathcal{P}(\tau_4) = 0.2$, and $\mathcal{P}(\tau_5) = 0.2$. (Recall the form of the optimal messaging policy in Corollary 3.3.) Blue: Sender's utility as a function of the cutoff index when they make no queries. The sender's optimal utility is given by the blue dashed line. Red: Sender's utility as a function of cutoff index when they make the simulation query q which separates $\{\tau_1, \tau_2, \tau_3\}$ from $\{\tau_4, \tau_5\}$. The red dashed line denotes the sender's utility from messaging optimally after making query q.

Algorithm 2 may be modified to compute the optimal querying policy in Binary BP with *costly* simulation queries; see Appendix D.2 for details. The key idea behind Algorithm 2 is that in Binary BP with simulation queries, there is always a "total ordering" over both receiver beliefs and queries, and thus one can use dynamic programming in order to iteratively construct an optimal solution. This intuition does not carry over to the setting with general partition queries, as no such total ordering exists. In fact, 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 can achieve expected utility at least u after making queries $\mathcal{Q}'$?*

Theorem 5.4. *The Non-Adaptive Decision Problem is NP-Complete in Binary BP.*

To prove NP-Hardness, we reduce from 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 partition query for every subset in S. We set u and $\{p_\tau\}_{\tau \in \mathcal{T}}$ in such a way that the sender can only achieve expected utility u if she can distinguish between every receiver type using K non-adaptive queries, i.e. the receiver subset for type τ 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`.

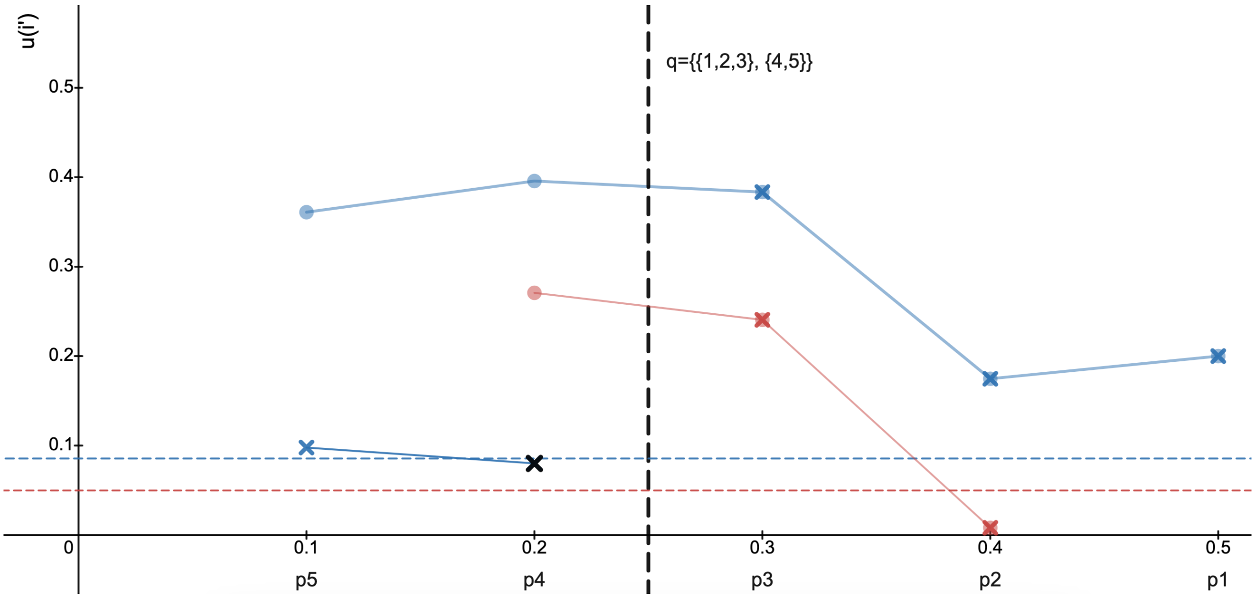

Figure 5: Submodularity in Binary BP with simulation queries. Under this problem instance there are five types with beliefs $p_1 = 0.5$, $p_2 = 0.4$, $p_3 = 0.3$, $p_4 = 0.2$, and $p_5 = 0.1$ (x-axis), and second-order prior $\mathcal{P}(\tau_1) = 0.2$, $\mathcal{P}(\tau_2) = 0.01$, $\mathcal{P}(\tau_3) = 0.39$, $\mathcal{P}(\tau_4) = 0.2$, and $\mathcal{P}(\tau_5) = 0.2$. (Recall the form of the optimal messaging policy in Corollary 3.3.) Blue: Sender's utility as a function of the cutoff index. Red: Sender's utility as a function of cutoff index when they know the true receiver belief is either p_2, p_3, or p_4 from previous queries. Blue and red crosses denote the sender's expected utility after query q is made. The black cross denotes both a blue and red cross on top of one another. The blue and red dashed lines denote the sender's respective marginal gains in utility from messaging optimally after making query q. The marginal gain is always higher when the sender has more uncertainty over the receiver's true type.

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 she makes.

Theorem 5.5. *In Binary BP, the sender's Bayesian-expected utility exhibits decreasing marginal returns from additional simulation queries.*

The proof proceeds via a case-by-case analysis on the difference in expected sender utility as a function of the queries which are made. The key step is to leverage the form of the sender's optimal messaging policy under uncertainty in Binary BP (Corollary 3.3) to argue that no matter how an additional query splits the current belief partition, the expected marginal gain is always weakly larger than whenever the set of queries is a superset. See Figure 5 for a visualization of this idea.

An immediate corollary of Theorem 5.5 is that the greedy non-adaptive querying policy (Algorithm 6) is $(1 - 1/e)$-approximately optimal in Binary BP. See Appendix D.5 for details. All of our results for Binary BP carry over to the non-adaptive variant of the costly (simulation) query setting. In particular, in Appendix D.6 we show that the sender's expected utility still exhibits decreasing marginal returns in the costly Binary BP setting, and a natural modification of the non-adaptive greedy policy (Algorithm 7) is $(1 - 1/e)$-approximately optimal.

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

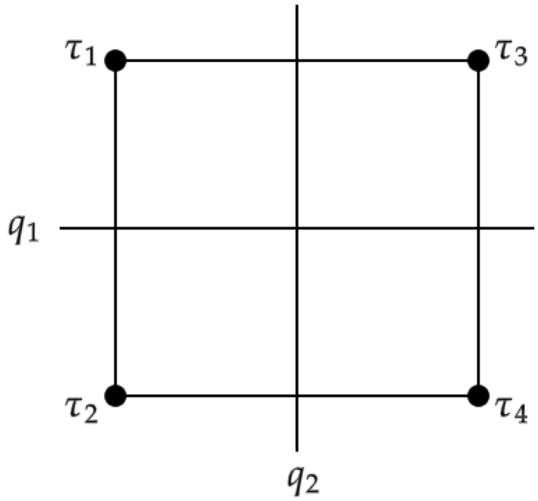

Figure 6: Illustration of Example 5.6 in two dimensions. Under this setting, there are four receiver types τ_1, τ_2, τ_3, τ_4 and two queries. Query q_1 separates τ_1 and τ_3 from τ_2 and τ_4. Query q_2 separates τ_1 and τ_2 from τ_3 and τ_4. The sender and receiver utilities are constructed in such a way that the sender's expected utility only increases after making both q_1 and q_2.

Example 5.6. *For some $L \in \mathbb{N}$, consider a setting with state space $\Omega = \{\omega_0, \omega_1, \ldots, \omega_{2^L-1}, \omega_\varnothing\}$ and receiver types $\mathcal{T} = \{0, 1, \ldots, 2^L - 1\}$ 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 sender utility function $u_S(\omega, a) = \mathbb{1}\{a \neq \omega_\varnothing\}$ and let $u_R(\omega, a) = 2 \cdot \mathbb{1}\{a = \omega\} - 1$ if $a \in \Omega \backslash \{\omega_\varnothing\}$ and $u_R(\omega, \omega_\varnothing) = 0$. Suppose there are L queries $\mathcal{Q} = \{q_1, \ldots, q_L\}$, where query q_j reveals the j-th bit of $\mathbf{p}_{i^*}$, where i^* is the true receiver type. Finally, let $\mathcal{P}$ be the uniform distribution over $\mathcal{T}$.*

See Figure 6 for an illustration of this setting when $L = 2$. The receiver's actions can be interpreted as "guesses" as to what the underlying state is: she gets utility 1 if she correctly guesses the state and utility -1 if she makes an incorrect guess. The receiver always gets utility 0 if she guesses that the state is $\omega_\varnothing$, regardless of whether this guess is correct. The sender's utility is such that she wants to incentivize the receiver to guess any state other than $\omega_\varnothing$. If the sender knows the receiver's belief exactly, she can incentivize her to *never* guess $\omega_\varnothing$. However if the sender has *any* uncertainty over the receiver's belief, the receiver will guess $\omega_\varnothing$ with (the same) constant probability under the sender's optimal messaging policy, regardless of how much uncertainty the sender has.

Theorem 5.7. *Consider the setting of Example 5.6 with $\epsilon = \frac{1}{2^{L-1}+2}$. The sender's Bayesian-expected utility does not exhibit decreasing marginal returns from additional queries.*

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, we can also construct an instance in which the greedy non-adaptive querying policy performs exponentially worse than the optimal policy (Corollary D.8).

Protocol: Bayesian Persuasion with Commitment to Query

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

Figure 7: Description of the Bayesian Persuasion game between a sender and receiver, when the sender can commit to a querying policy.

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 messaging policy, but must make all of her oracle queries before seeing the state. In this section, we explore the effects of *further* commitment power 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 her messaging 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.*

Similarly, a messaging policy is *subset-informed* if the realized message depends on both the realized state and the outcome of the query.

Definition 6.2 (Subset-Informed Messaging Policy)**.** *A messaging policy $\sigma : \Omega \times \mathcal{S} \to \mathcal{M}$ is* subset-informed *if the message sent depends on the realized belief subset.*

Note that a subset-informed messaging policy is different than a messaging policy which operates on a refined subset of receiver beliefs. In particular, a subset-informed messaging policy takes a subset of beliefs as input (in addition to the state) and produces a message. In contrast, our previous definition of a messaging policy (Definition 2.1) does not take any information about receiver belief subsets as input.

See Figure 7 for a summary of the updated interaction protocol under additional commitment. Note that the ability to commit to a state-informed querying policy will never hurt the sender, since the sender has the option of choosing a state-independent querying policy. On the other hand, the sender can sometimes benefit substantially from this additional commitment power, as the following example illustrates.

Example 6.3. *Consider the following natural extension of Binary BP 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 $\mathcal{T} = \{\tau_1, \tau_2, \tau_3\}$ with respective beliefs $\mathbf{p}_1 = [0.75, 0.25, 0]$, $\mathbf{p}_2 = [0.75 + \delta, 0, 0.25 - \delta]$, $\mathbf{p}_3 = [0, 0.75 + \epsilon, 0.25 - \epsilon]$ for some $\epsilon, \delta > 0$.*

Observe that in the setting of Example 6.3, after seeing the state realization the sender will always be able to rule out one of the receiver beliefs (since they all put zero probability mass on

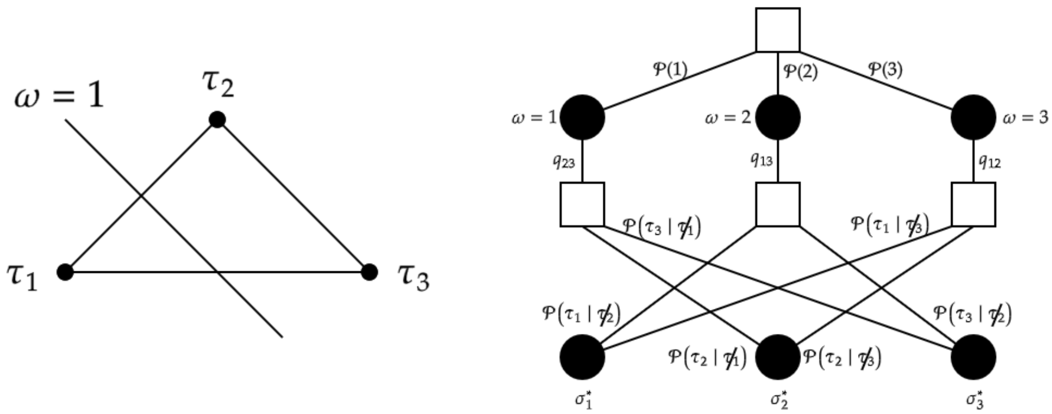

Figure 8: Visualization of Example 6.3. Left: Seeing the state helps the sender because it allows them to eliminate one receiver type before querying the oracle. Right: The procedure described in Example 6.3 is incentive-compatible for the receiver because the messaging policy faced by any individual receiver type will be the same (namely, the sender's optimal messaging policy for that type), regardless of the state realization.

a different state), and thus she can use her query to distinguish between the remaining two beliefs. If the sender commits to messaging according to the optimal messaging policy for the realized belief, this procedure is BIC since the sender will always send a message to a receiver of type i using messaging policy σ_i^* for every possible state realization. Note that if the sender queried the oracle *before* seeing the state, she would not always be able to distinguish between all three beliefs. See Figure 8 for a visualization of this setting.

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

Definition 6.4. *We say that a subset-informed messaging policy σ* implements *a messaging 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 $\omega \in \Omega$, $a \in \mathcal{A}$.*

In other words, a subset-informed messaging policy σ can implement a (subset-uninformed) messaging policy σ_i from receiver type i's point of view if they cannot distinguish between the two after aggregating all of the messages which recommend them to take action a, for every $a \in \mathcal{A}$.

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

The result follows from writing down the problem of designing a BIC querying and messaging 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* messaging policies for different receiver beliefs when querying *before* seeing the state.

7 Conclusion and Future Work

We initiate the study of information design with an oracle, motivated by recent advances in Large Language Models and Generative AI, as well as more traditional settings such as experimentation on a small number of users. 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 messaging under uncertainty and can 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, computing the sender's optimal non-adaptive querying policy is an NP-Complete problem, it can be solved for efficiently via dynamic programming in Binary BP with simulation queries. Motivated by this observation, we study the sender's expected utility as a function of the queries they make. We find that the sender's expected utility exhibits decreasing marginal returns as a function of additional queries in Binary BP. 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. While we focus on Bayesian Persuasion for its simplicity and ubiquity, it would be interesting to study the benefits of oracle access in other information design settings, such as cheap talk [CS82] or verifiable disclosure [GH80]. 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 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. Finally 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 oracle access could help the sender achieve higher utility whenever the receiver's utility function is unknown.

Acknowledgements

The authors would like to thank James Best, the participants of the DIMACS Workshop on Foundation Models, Large Language Models, and Game Theory, and the participants of the SLMath workshop on Algorithms, Approximation, and Learning in Market and Mechanism Design for helpful comments and discussions.

References

[AAK23] Gati Aher, Rosa I. Arriaga, and Adam Tauman Kalai. Using large language models to simulate multiple humans and replicate human subject studies. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[ABM10] Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *23rd Conf. on Learning Theory (COLT)*, pages 41–53, 2010.

[ADF+23] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[ASCF+23] 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.

[BBHP15] 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.

[BCC+23] 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.

[BD23] 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.

[BHP14] Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Learning optimal commitment to overcome insecurity. *Advances in Neural Information Processing Systems*, 27, 2014.

[BIN23] James Brand, Ayelet Israeli, and Donald Ngwe. Using gpt for market research. *Available at SSRN 4395751*, 2023.

[BM19] Dirk Bergemann and Stephen Morris. Information design: A unified perspective. *Journal of Economic Literature*, 57(1):44–95, March 2019.

[BMS11] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure Exploration in Multi-Armed Bandit Problems. *Theoretical Computer Science*, 412(19):1832–1852, 2011.

[CCMG20] Matteo Castiglioni, Andrea Celli, Alberto Marchesi, and Nicola Gatti. Online bayesian persuasion. *Advances in Neural Information Processing Systems*, 33:16188–16198, 2020.

[CLSZ23] Yiting Chen, Tracy Xiao Liu, You Shan, and Songfa Zhong. The emergence of economic rationality of gpt. *arXiv preprint arXiv:2305.12763*, 2023.

[CMCG21] 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.

[CS82] Vincent P Crawford and Joel Sobel. Strategic information transmission. *Econometrica: Journal of the Econometric Society*, pages 1431–1451, 1982.

[CS06] 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.

[DMPL+23] Paul Duetting, Vahab Mirrokni, Renato Paes Leme, Haifeng Xu, and Song Zuo. Mechanism design for large language models. *arXiv preprint arXiv:2310.10826*, 2023.

[DP22] Piotr Dworczak and Alessandro Pavan. Preparing for the worst but hoping for the best: Robust (bayesian) persuasion. *Econometrica*, 90(5):2017–2051, 2022.

[EMM06] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *J. of Machine Learning Research (JMLR)*, 7:1079–1105, 2006.

[FGP+23] 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.

[GH80] Sanford J Grossman and Oliver D Hart. Disclosure laws and takeover bids. *The Journal of Finance*, 35(2):323–334, 1980.

[Guo23] Fulin Guo. Gpt agents in game theory experiments. *arXiv preprint arXiv:2305.05516*, 2023.

[Hor23] John J Horton. Large language models as simulated economic agents: What can we learn from homo silicus? *arXiv preprint arXiv:2301.07543*, 2023.

[HW21] Ju Hu and Xi Weng. Robust persuasion of a privately informed receiver. *Economic Theory*, 72:909–953, 2021.

[J23] Joshua J. How much does gpt-4 cost?, 2023.

[Kam19] Emir Kamenica. Bayesian persuasion and information design. *Annual Review of Economics*, 11:249–272, 2019.

[KG11] Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *American Economic Review*, 101(6):2590–2615, 2011.

[KL17] Ron Kohavi and Roger Longbotham. Online controlled experiments and a/b testing. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning and Data Mining*, pages 922–929. Springer, Boston, MA, 2017.

[KL23] 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.

[KLSH09] Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M. Henne. Controlled experiments on the web: survey and practical guide. *Data Min. Knowl. Discov.*, 18(1):140–181, 2009.

[KMZL17] Anton Kolotilin, Tymofiy Mylovanov, Andriy Zapechelnyuk, and Ming Li. Persuasion of a privately informed receiver. *Econometrica*, 85(6):1949–1964, 2017.

[KOC23] Vojtech Kovarik, Caspar Oesterheld, and Vincent Conitzer. Game theory with simulation of other players. *arXiv preprint arXiv:2305.11261*, 2023.

[Kos22] Svetlana Kosterina. Persuasion with unknown beliefs. *Theoretical Economics*, 17(3):1075–1107, 2022.

[LCM09] 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.

[LH23] Nunzio Lorè and Babak Heydari. Strategic behavior of large language models: Game structure vs. contextual framing. *arXiv preprint arXiv:2309.05898*, 2023.

[MT04] Shie Mannor and John N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *J. of Machine Learning Research (JMLR)*, 5:623–648, 2004.

[MTV+23] Sandra Matz, Jake Teeny, Sumer S Vaid, Gabriella M Harari, and Moran Cerf. The potential of generative ai for personalized persuasion at scale. *PsyArXiv*, 2023.

[Ope23] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv 2303.08774*, 2023.

[PM23] Heinrich Peters and Sandra Matz. Large language models can infer psychological dispositions of social media users. *arXiv preprint arXiv:2309.08631*, 2023.

[POC+23] 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.

[PS22] Alexei Parakhonyak and Anton Sobolev. Persuasion without priors. *Available at SSRN 4377557*, 2022.

[PSTZ19] 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.

[TMS+23] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[Tsu23] Toshihiro Tsuchihashi. How much do you bid? answers from chatgpt in first-price and second-price auctions. *Journal of Digital Life*, 3, 2023.

[VS10] Heinrich Von Stackelberg. *Market structure and equilibrium.* Springer Science & Business Media, 2010.

[ZIX21] 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.

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 9.

Protocol: Bayesian Persuasion with Oracle Queries

1. Sender makes K oracle queries, either adaptively or non-adaptively.
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 $\mathbf{p}_{\tau'}|m$ 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 9: Uninformed receiver setting.

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 2 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 To Think About Querying?

B.1 Proof of Proposition 3.1

Proposition 3.1. *[Revelation Principle for Unknown Receiver Beliefs] It always suffices for the sender to use a message space of size $M = A^T$. In Binary BP, it suffices to use $M = T+1$ 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 BP setting, a receiver with prior p takes action $a = 1$ after seeing message m if and only if $p \geq \frac{\sigma(m|0)}{\sigma(m|1)+\sigma(m|0)}$. 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 $p < \frac{\sigma(m|0)}{\sigma(m|1)+\sigma(m|0)}$, which implies that a receiver with prior $p' \leq p$ will also take action $a = 0$. □

B.2 Proof of Corollary 3.3

Corollary 3.3. *[Optimal Messaging Policy; Binary BP] In Binary BP, 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 messaging policy can be computed in time $\mathcal{O}((T')^2)$. The optimal policy is given by $\sigma(m_{i^*}|1) = 1$, $\sigma(m_{i^*}|0) = \frac{p_{i^*}}{1-p_{i^*}}$, and $\sigma(0|0) = 1 - \frac{p_{i^*}}{1-p_{i^*}}$, where $i^* := \arg\max_{i' \in [L,H]} \sum_{i=L}^{i'} \mathcal{P}(\tau_i) \cdot \left(p_i + (1-p_i) \cdot \frac{p_{i'}}{1-p_{i'}}\right)$.*

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

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

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

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

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. □

B.3 Proof of Proposition 3.4

Proposition 3.4. *[Relationship between simulation queries and separation oracles] Let $\alpha(\mathbf{p}) := \sum_{\omega \in \Omega}(u_R(\omega, m_q) - u_R(\omega, a)) \cdot \sigma_q(m_q|\omega) \cdot \mathbf{p}[\omega]$. By making a simulation query $q = (\sigma_q, m_q)$, the sender is specifying a polytope $\mathcal{R}_q := \{\mathbf{p} \in \Delta^d \; : \; \alpha(\mathbf{p}) \geq 0, \forall a \in \mathcal{A}\}$ and the oracle returns either (i) $\mathbf{p}_{\tau^*} \in \mathcal{R}_q$ or (ii) $\mathbf{p}_{\tau^*} \notin \mathcal{R}_q$ and for some $a' \in \mathcal{A}$ for all $\mathbf{p}' \in \mathcal{R}_q$, $\alpha(\mathbf{p}' - \mathbf{p}_{\tau^*}) > 0$.*

Algorithm 3 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`

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 (ii). □

B.4 Example 3.6 Expanded

Example B.1. *Suppose that there are d states and d actions, where $u_R(\omega, a) = \mathbb{1}\{\omega = a\}$. Consider d receiver beliefs $\mathbf{p}_1, \ldots, \mathbf{p}_d$ 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 $j \neq 1$,*

$$\sigma(a_1|\omega_1) \cdot \frac{2}{d+1} \geq \sigma(a_1|\omega_j) \cdot \frac{1}{d+1}.$$

Therefore, $\mathbf{p}_1$ will take action a_1 when $m = a_i$ if for all $j \neq 1$, $\sigma(a_1|\omega_j) = 2\sigma(a_1|\omega_1)$. Now let us consider another receiver type $i \neq 1$. Type i will default to taking action i under this messaging policy whenever $m = a_1$, since

$$\begin{aligned} \sigma(m|\omega_1) \cdot \frac{1}{d+1} < 2\sigma(m|\omega_1) \cdot \frac{2}{d+1} \\ \sigma(m|\omega_1) \cdot \frac{1}{d+1} < 2\sigma(m|\omega_1) \cdot \frac{1}{d+1} \end{aligned}$$

where the first line is proportional to how much the receiver loses in expectation by not taking action $i \neq 1$ when recommended action 1, and the second line is proportional to how much she loses in expectation by not taking action $j \neq i \neq 1$ when recommended action 1.

B.5 A Procedure for Finding All Simulation Queries

In general, 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.

Definition B.2 (Separability)**.** *Two receiver types τ, τ' are* separable *if there exists a simulation query $q = (\sigma_q, m_q)$ such that*

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

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 3 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 B.3. *For any τ, τ' which are separable (according to Definition B.2), Algorithm 3 returns a simulation query which separates them in time $\mathcal{O}(A^2 \cdot \mathrm{LP}(d, d + A))$.*

Proof. Observe that under Definition B.2, 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. □

Algorithm 4 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:

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

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

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

$$\vdots$$

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

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

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

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

end if

end for

return `Fail`

Algorithm 4 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 B.3 with the exception of the runtime analysis, which is exponentially worse.[7]

[7]Our focus is on the computational complexity of computing the optimal querying policy, as opposed to the computational complexity of discovering all unique simulation queries because the set of all unique simulation queries may be pre-computed ahead of time for all problem instances as long as the set of possible receiver types is known.

C Appendix for Section 4: Adaptive Querying Policies

C.1 Proof of Theorem 4.2

Theorem 4.2. *Algorithm 1 computes the sender's optimal adaptive querying policy for K partition queries in $\mathcal{O}(FKQT)$ time. In Binary BP with simulation queries, the runtime is $\mathcal{O}(T^4K)$.*

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 partition 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 1 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 for the remaining $\kappa - 1$ queries. This is precisely the definition of $V[\mathcal{T}', \kappa]$, and so Algorithm 1 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.

Algorithm 1 simplifies significantly in the Binary BP 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. In the Binary BP setting with simulation queries, the set of feasible information partitions simplifies to $\mathcal{F} = \{\{\tau_i, \dots, \tau_j\} : 1 \leq i \leq j \leq T\}$. This implies that the update step of Algorithm 1 simplifies to computing $V[(i,j), k] = \max_{q \in \{i,\dots,j\}} V[(i,j), k, q]$ for all $1 \leq i < j \leq T$, where

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

Note that in the Binary BP 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 $F = \mathcal{O}(T^2)$. □

C.2 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 1 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 5.

Algorithm 5 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 messaging policy as defined in Corollary 3.2.
- Let $V[\mathcal{T}',k,q_\varnothing] := V[\mathcal{T}',0]$ for every $1\le k\le T$ and all $\mathcal{T}'\in\mathcal{F}$.
- For every $1\le k\le T$ 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 $q\ne q_\varnothing$.
- 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.

Theorem C.1. *Algorithm 5 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 $1\le k\le\min\{T-1,Q\}$ 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. □

Corollary C.2. *In the Binary BP setting with simulation queries, the update step of Algorithm 5 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*

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

The runtime of Algorithm 5 in this setting is $\mathcal{O}(T^4K)$.

Proof. Correctness follows from the proof of Theorem C.1 and the observation that in the Binary BP setting with simulation queries, the set of feasible information partitions simplifies to $\mathcal{F} = \{\{\tau_i,\dots,\tau_j\} : 1\le i\le j\le T\}$. Note that in the Binary BP 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 Binary BP with simulation queries, Algorithm 2 computes the sender's optimal non-adaptive querying policy in $\mathcal{O}(T^3K)$ 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 $\{i, \dots, j\}$. We can write the sender's maximum expected utility for K queries in the range $\{i, \dots, j\}$ as

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

where $q_1 < q_2 < \dots < q_K$. 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 Observe that each $\mathbb{P}_{\mathcal{P}}(\tau^* \in \{q+1, \dots, i\})$ for $q \in Q$, $i \in [1, \dots, T]$ 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 $\mathcal{O}(T^2K)$ such values to compute. □

D.2 Costly Queries in the Binary BP Setting

Algorithm 2 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 BP setting with costly simulation queries, the above algorithm computes the sender's non-adaptive querying policy in $\mathcal{O}(T^3K)$ 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

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

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 $\{1, \ldots, q_K\}$ 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 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 `set_cover`(U, S, K) and `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 is NP-Complete in Binary BP.*

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.[8] 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 $\texttt{set\_cover}(U, S, K) = \texttt{yes}$. 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 $\texttt{query\_separation}$ 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 $s_{q, i(\tau_\varnothing)}$ 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 $\texttt{non-adaptive}(\mathcal{T}(U), \mathcal{P}, \mathcal{Q}(S), K, u) = \texttt{yes}$.

Part 2: Suppose $\texttt{set\_cover}(U, S, K) = \texttt{no}$. 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. $s_{q, i(\tau_\varnothing)}$) 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 $\texttt{non-adaptive}(\mathcal{T}(U), \mathcal{P}, \mathcal{Q}(S), K, u) = \texttt{no}$. □

D.4 Proof of Theorem 5.5

Theorem 5.5. *In Binary BP, the sender's Bayesian-expected utility exhibits decreasing marginal returns from additional simulation queries.*

[8]Note that it is always possible to do this, as in the Binary BP setting the optimal signaling policy will be different for two receiver types with priors $p' \neq p$, $p, p' \leq 0.5$.

Proof. It suffices to show that for any two sets of simulation queries $\widehat{\mathcal{Q}}$, $\tilde{\mathcal{Q}}$ such that $\tilde{\mathcal{Q}} \subseteq \widehat{\mathcal{Q}} \subseteq \mathcal{Q}$, and any query $q \in \mathcal{Q}$. we have that

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

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 BP Setting

Corollary D.4. *In the Binary BP setting, the non-adaptive querying policy $\mathcal{Q}_K$ produced by Algorithm 6 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

Algorithm 6 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 $\mathcal{Q}_K$

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 $\eta_{\mathcal{Q}_i \cup \{q_1, \ldots, q_K\}}[\tau] = \eta_{\mathcal{Q}_i \cup \mathcal{Q}_K^*}[\tau]$, $\forall i, \tau$.

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

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),

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

Combining the above two statements, we get that

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

Rearranging, we get that

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

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

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

or equivalently,

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

Rearranging,

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

By induction,

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

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:

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

□

D.6 Greedy Approximation in the Costly Binary BP Setting

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

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

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

Corollary D.5. *In the Binary BP 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

Theorem 5.7. *Consider the setting of Example 5.6 with $\epsilon = \frac{1}{2^{L-1}+2}$. The sender's Bayesian-expected utility does not exhibit decreasing marginal returns from additional queries.*

It suffices to show that for 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

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

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

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 D.8

Corollary D.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 6 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 by Theorem 5.7, 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. □

Corollary D.9. *[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 $\frac{L}{2^L} + u(1 - \frac{L}{2^L})$, 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). □

E Appendix for Section 6: Committing to Query

E.1 Proof of Theorem 6.5

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

Proof. Consider the Binary BP 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 $\sigma_i(m|1) \cdot p_i \geq \sigma_i(m|0) \cdot (1 - p_i)$, where $\sigma_i(m|\alpha)$ 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 Proposition 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

$$\begin{aligned}
\sigma(m=100|\omega=1,\{\tau_1,\tau_2\}) &= \sigma_1(m=1|\omega=1) - \sigma(m=111|\omega=1,\{\tau_1,\tau_2\}) \\
&\quad - \sigma(m=110|\omega=1,\{\tau_1,\tau_2\}) - \sigma(m=101|\omega=1,\{\tau_1,\tau_2\}) \\
\sigma(m=100|\omega=0,\{\tau_1\}) &= \sigma_1(m=1|\omega=0) - \sigma(m=111|\omega=0,\{\tau_1\}) \\
&\quad - \sigma(m=110|\omega=0,\{\tau_1\}) - \sigma(m=101|\omega=0,\{\tau_1\}) \\
\sigma(m=010|\omega=1,\{\tau_1,\tau_2\}) &= \sigma_2(m=1|\omega=1) - \sigma(m=111|\omega=1,\{\tau_1,\tau_2\}) \\
&\quad - \sigma(m=110|\omega=1,\{\tau_1,\tau_2\}) - \sigma(m=011|\omega=1,\{\tau_1,\tau_2\}) \\
\sigma(m=010|\omega=0,\{\tau_2,\tau_3\}) &= \sigma_2(m=1|\omega=0) - \sigma(m=111|\omega=0,\{\tau_2,\tau_3\}) \\
&\quad - \sigma(m=110|\omega=0,\{\tau_2,\tau_3\}) - \sigma(m=011|\omega=0,\{\tau_2,\tau_3\}) \\
\sigma(m=001|\omega=1,\{\tau_3\}) &= \sigma_3(m=1|\omega=1) - \sigma(m=111|\omega=1,\{\tau_3\}) \\
&\quad - \sigma(m=101|\omega=1,\{\tau_3\}) - \sigma(m=011|\omega=1,\{\tau_3\}) \\
\sigma(m=001|\omega=0,\{\tau_2,\tau_3\}) &= \sigma_3(m=1|\omega=0) - \sigma(m=111|\omega=0,\{\tau_2,\tau_3\}) \\
&\quad - \sigma(m=101|\omega=0,\{\tau_2,\tau_3\}) - \sigma(m=011|\omega=0,\{\tau_2,\tau_3\})
\end{aligned}$$

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

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

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,

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

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:

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

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:

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

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. $\square$