Provably Optimal Learning Algorithms for Assistance Games

Anonymous Authors[1]

Abstract

This paper studies an online variant of the *assistance games* framework, where an informed agent and an uninformed agent repeatedly interact over T timesteps to optimize a common reward function. While the informed agent (the human) observes a latent state of the world, the uninformed agent (the assistant) observes only the human's actions. We provide the first provably efficient learning algorithms for repeated assistance games. We introduce the notion of *assistance regret*: the gap between the cumulative utility of interactions and that of the optimal joint policies in hindsight, which map latent states to action pairs. We present decentralized algorithms for both the human and the assistant that achieve a $(1 - 1/e)$-approximate assistance regret rate of $\mathcal{O}\left(T^{3/4}\right)$, with runtime polynomial in the size of the action and state spaces. We further prove that achieving a regret approximation factor better than $(1-1/e)$ is computationally intractable, and demonstrate how these generic no-regret algorithms can be tailored to a pseudo-decentralized setting—using a shared random string—to achieve the optimal rate of $\mathcal{O}\left(T^{1/2}\right)$.

1. Introduction

Consider a repeated interaction between two cooperative agents who share a common objective but have asymmetric access to information. One agent observes a changing latent state—such as a preference, type, or objective—while the other must act based only on indirect signals produced by the first agent. Such settings arise naturally in abstractions of human–assistant interaction, cooperative multi-agent systems, training of assistive AI, and emergent communication, where the former agent represents a human principal with private preferences and the latter represents an AI assistant seeking to act on the human's behalf.

Despite the absence of strategic misalignment in utilities, coordination in these environments poses a fundamental challenge. On the one hand, the lack of a shared frame of reference or common language means that the agents must learn how to communicate by observing each other's actions over time. On the other hand, actions simultaneously generate utility, causing them to serve a dual role as instruments for achieving reward and signals for conveying the latent state. This tension between informativeness and utility lies at the core of the problem.

One theoretical perspective for capturing these challenges is the *assistance games* framework, also known as *Cooperative Inverse Reinforcement Learning* (Hadfield-Menell et al., 2016). This framework models interactions between a human and an assistive AI system as a cooperative game with partial observability. Prior work has studied equilibria and structural properties of optimal assistive policies—for example, tradeoffs between the informativeness and utility of actions—but the algorithmic problem of computing and learning such policies has been unexplored.

In this paper, we introduce an **online variant of assistance games and give computationally efficient, decentralized algorithms for learning near-optimal policies for both the human and the assistant.**

Our model captures repeated interaction between an informed agent (the human) and an uninformed agent (the assistant). In each round t, a latent state $\theta^{(t)} \in \Theta$—possibly drawn from a nonstationary process—is realized and observed only by the human. This state represents private information about the human's preferences or objectives. The human selects an action $a_H^{(t)}$ from a finite action space $\mathcal{A}_H$ of size M_H. The assistant observes $a_H^{(t)}$, but not $\theta^{(t)}$, and responds with an action $a_A^{(t)}$ from a finite action space $\mathcal{A}_A$ of size M_A. Both agents then receive a common reward $r\left(a_H^{(t)}, a_A^{(t)}; \theta^{(t)}\right)$, which depends on the joint actions and the human's private state.

To measure performance, we consider the space of *joint* human–assistant policies: pairs (π_H, π_A) where a human policy $\pi_H : \Theta \to \mathcal{A}_H$ maps latent states to human actions and an assistant policy $\pi_A : \mathcal{A}_H \to \mathcal{A}_A$ maps hu-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

man actions to assistant actions. We measure success via α-*approximate assistance regret* (Definition 3.1): the gap between the agents' cumulative reward and an α factor of the cumulative reward of the best joint policy in hindsight.

Our main result gives a pair of decentralized learning algorithms for the human and the assistant that achieve $(1-1/e)$-approximate assistance regret at a rate of $\mathcal{O}\left(T^{3/4}\right)$ in time $\text{poly}(N, M_H, M_A)$ (Theorem 4.1); with light initial coordination—a shared encoding from human action sequences to assistant policies—this rate improves to the optimal $\mathcal{O}\left(\sqrt{T}\right)$ (Theorem 4.2). Two technical ingredients drive the result. First is a reduction casting joint policy optimization in assistance games as online submodular maximization with matroid constraints (Lemma D.1 and Proposition 5.4). This reduction enables computational tractability despite exponential size of the joint policy space, by importing tools from online submodular optimization (Salem et al., 2024). Second is a regret-decomposition lemma (Lemma 4.4) that bounds assistance regret by the centralized algorithm's external regret, its number of policy switches, and the assistant's tracking regret against a moving target. This decomposition guides our design toward two structural properties: *stability* (few policy switches) for the human and *adaptivity* (low tracking regret) for the assistant. The approximation factor of $1 - 1/e$ is tight: no efficient algorithm achieves sublinear α-approximate assistance regret for $\alpha > 1 - 1/e$ unless $\mathsf{RP} = \mathsf{NP}$ (Theorem 4.3).

2. Related Work

The idea of modeling the problem of AI assistance mathematically had been first proposed almost ten years ago in work including Fern et al. (2014) and Hadfield-Menell et al. (2016). Shah et al. (2020) show that such an assistant would be able to better satisfy human preferences than traditional reward learning algorithms, and Hadfield-Menell et al. (2016; 2017) show that an optimal assistant would avoid issues of misalignment to human values. Malik et al. (2018) and Laidlaw et al. (2025) propose algorithms for solving assistance games that are empirically performant, but the problem of designing learning algorithms that provably play such games optimally remains open. Similar to us, previous work has also taken the approach of reducing the assistance games problem to other classes of problems — particularly to classes of Partially Observable Markov Decision Processes (POMDPs). We introduce a new reduction with a mathematical structure that allows us to demonstrate provable optimality guarantees. A particular subclass of assistance games are *communication games*. We describe related works on this literature in more detail in Section F.

Our work is related to literature on equilibrium computation. We can frame our goals is decentralized learning of optimal equilibria in assistance games. Computing the optimal equilibrium is computationally intractable (Gilboa & Zemel, 1989) in general, including in games of common interest (Chu & Halpern, 2001; Conitzer & Sandholm, 2006). In common interest games, the optimal equilibrium is also each player's Stackelberg equilibrium. The computational tractability of computing the Stackelberg equilibrium is shown to depend on the geometry of the game and can be intractable in general (Letchford et al., 2009; Peng et al., 2019). For games where the Stackelberg equilibrium can be computed efficiently, convergence to the Stackelberg equilibrium can be achieved through dynamics where one player's learning dynamic is more stable than the other (Brown et al., 2024; Zrnic et al., 2021). The learning dynamics we propose for communication games also satisfy this property.

Other areas studying cooperative interactions between agents include work on team decision theory (e.g. (Radner, 1962; Ho et al., 1972; Nayyar et al., 2013; Mahajan & Mannan, 2016; Malikopoulos, 2022)) which takes a control theoretic approach to studying optimal cooperation, and work on human-AI collaboration studying when human-AI systems achieve complementarity i.e., better performance than the sum of individual components (e.g., (Green & Chen, 2019; Bansal et al., 2021; Wilder et al., 2020; Steyvers et al., 2022; Donahue et al., 2022; Athey et al., 2020; Alur et al., 2024; Greenwood et al., 2025; Collina et al., 2026))

3. Model and Preliminaries

The online assistance game is defined by a preference space Θ of size N, two sets of actions $\mathcal{A}_H$ and $\mathcal{A}_A$ of sizes M_H and M_A respectively, and a bounded reward function $r : \mathcal{A}_H \times \mathcal{A}_A \times \Theta \to [0, 1]$, for some max reward U. Two agents, an assistant and a human, repeatedly play the game over T rounds. Every round $t \in [T]$ of the assistance game involves the following steps:

1. The human observes $\theta^{(t)}$ and plays an action $a_H^{(t)} \in \mathcal{A}_H$.
2. The assistant sees only the action $a_H^{(t)}$ the human takes but not the preference $\theta^{(t)}$. The assistant then takes the action $a_A^{(t)} \in \mathcal{A}_A$ in response.
3. Both the assistant and the human receive the reward $r\left(a_H^{(t)}, a_A^{(t)}; \theta^{(t)}\right)$.

The parameter $\theta^{(t)}$ captures the human's preference at round t. We allow the preference $\theta^{(t)}$ to change over time, allowing robustness to human preferences changing over time. However, we impose the restriction that the sequence of preferences is fixed before the game begins. That is, the preference at each round cannot be selected adaptively based on previous rounds of interaction. We call this the *oblivious* setting. The stronger model of an adversary who can select the preference $\theta^{(t)}$ adversarially based on previous rounds

of interaction is called the *adaptive* setting. We prove in Section B that there are fundamental limitations to learning optimal assistance in the adaptive setting.

In the normal form representation of the assistance game, the human's policy space is the set Π_H of mappings from the human's private preference to human's actions, $\pi_H : \Theta \to \mathcal{A}_H$, and the assistant's policy space is the set Π_A of mappings from an observed human action to an assistant action, $\pi_A : \mathcal{A}_H \to \mathcal{A}_A$. For notational convenience, we write $r(\pi_H, \pi_A; \theta)$ to mean the reward the human and assistant receive when they play policy π_H and π_A when the human's preferences are θ, mathematically $r(\pi_H(\theta),\ \pi_A(\pi_H(\theta));\ \theta)$.

We introduce a notion of (approximate)regret to measure the joint success of the human and assistant in the online assistance game. We call this the *assistance regret* and simply refer to this as regret in the remainder of the paper. Assistance regret is defined in the following definition.

Definition 3.1 (α-Assistance regret)**.** *For a fixed $\alpha \in [0,1]$, the α-assistance regret of a sequence $\chi = \left(\theta^{(t)}, \pi_H^{(t)}, \pi_A^{(t)}\right)_{t=1}^T$, written $R_T^\alpha(\chi)$, is:*

$$\alpha \left(\max_{\substack{\pi_H^* \in \Pi_H \\ \pi_A^* \in \Pi_A}} \sum_{t=1}^T r\left(\pi_H^*, \pi_A^*; \theta^{(t)}\right) \right) - \sum_{t=1}^T r\left(\pi_H^{(t)}, \pi_A^{(t)}; \theta^{(t)}\right).$$

We suppress $\theta^{(t)}$ in the regret notion when it is clear from context and denote $r_t(\pi_H, \pi_A) = r(\pi_H, \pi_A; \theta^{(t)})$. We also refer to $R_T(\text{Alg})$ as the min-max α-assistance regret of algorithm Alg.

3.1. Submodular Maximization With Matroid Constraints

Our results will rely on tools from submodular maximization and matroid theory. We will describe the necessary tools in this section. Given an arbitrary set of elements $\mathcal{U}$, a set function over $\mathcal{U}$ is a mapping from subsets in $\mathcal{U}$ to real numbers, $f : 2^{\mathcal{U}} \to \mathbb{R}$.

A set function is said to be monotone when adding any element can only increase the value of the function. That is f is monotone when for all $S \subseteq \mathcal{U}$ and all $x \notin S$, it holds that $f(S \cup \{x\}) \geq f(S)$.

A set function is said to be submodular when the marginal improvement of adding an element can only decrease as the size of the set increases.

Definition 3.2 (Submodularity)**.** *Let f be a set function on a set $\mathcal{U}$. f is said to be submodular when, for all $S' \subseteq S \subseteq \mathcal{U}$, and all $x \notin \mathcal{U}$, the increase in value after adding x to S is smaller than the increase in value after adding x to S'. That is, $f(S \cup \{x\}) - f(S) \leq f(S' \cup \{x\}) - f(S')$.*

We will consider the constrained maximization of submodular functions under matroid constraints.

Definition 3.3 (Matroids)**.** *A pair $\mathcal{M} = (\mathcal{U}, \mathcal{I})$ is a matroid when $\mathcal{U}$ is a finite set, and $\mathcal{I} \subseteq 2^{\mathcal{U}}$ is a collection of subsets of $\mathcal{U}$ such that 1) For all $I \in \mathcal{I}$, if $J \subseteq I$, then $J \in \mathcal{I}$ and 2) For all $I, J \in \mathcal{I}$, if $|J| < |I|$, then there exists some $e \in I \setminus J$ such that $J \cup \{e\} \subseteq \mathcal{I}$.*

Any set in $I \in \mathcal{I}$ is said to be an independent set. The maximum size of an independent set in $\mathcal{I}$ is said to be the rank of the matroid $\mathcal{M}$.

Our results actually rely on a special subclass of submodular functions i.e., *weighted threshold potentials* (a sum of capped linear functions; Definition A.2) and on a special subclass of matroids i.e., *partition matroids* (Definition A.3). We provide more details on these subclasses in Section A.

3.2. Background on Online Learning

Regret notions and algorithms originating from the online learning literature will play a central role in our paper.

In a setting with action set $[K]$ and T rounds, for sequences of actions and reward functions (a_t, r_t) where $a_t \in [K]$ and $r_t : [K] \to [0,1]$, we can define the following notions of regret that we will make extensive use of in framing our results.

Definition 3.4 (External regret and number of switches)**.** *The α-approximate external regret of a sequence $\chi = (a_t, r_t)_{t=1}^T$ is $R_T^{\alpha,\text{ext}}(\chi) = \alpha \max_{a \in [K]} \sum_{t=1}^T r_t(a) - \sum_{t=1}^T r_t(a_t)$.*

The number of switches in the sequence is $\sum_{t=2}^T \mathbf{1}\{a_t \neq a_{t-1}\}$.

Definition 3.5 (Tracking regret with p segments)**.** *Tracking regret (Herbster & Warmuth, 1998) measures a learner's ability to compete with a sequence of a p-times changing benchmark of actions rather than a single best fixed action. For a sequence $\chi = (a_t, r_t)_{t=1}^T$, the tracking regret is defined as :*

$$R_T^{track}(\chi; p) = \max_{\substack{s_1 < \cdots < s_{p+1} \in [T] \\ s_1 = 1, s_{p+1} = T \\ b_1, \ldots, b_p \in [K]}} \sum_{i=1}^p \sum_{t=s_i}^{s_{i+1}} r_t(b_i) - \sum_{t=1}^T r_t(a_t).$$

This regret formulation is useful in changing environments where the optimal action varies over time.

Feedback models. Online learning algorithms select action a_t in a round t based on the available history consisting of actions and feedback of previous rounds. In a *full-information*

setting the entire reward function r_t is revealed as feedback at the end of round t. In a *bandit* setting, only the reward of the selected action $r_t(a_t)$ is revealed as feedback.

4. Main Results and Techniques

Our main results provide pairs of decentralized learning algorithms for the human and the assistant that are computationally efficient and result in sublinear $(1 - 1/e)$-approximate assistance regret. We first state our main theorems, discuss the optimality of the guarantees stated in these theorems, and finally describe the main ideas behind constructing these algorithms.

Our first theorem constructs decentralized algorithms that use quite general regret minimization algorithms and achieve a $\mathcal{O}\left(T^{3/4}\right)$ rate of $(1 - 1/e)$ -approximate assistance regret.

Theorem 4.1. *There are* $\mathrm{poly}(M_H, M_A, N, T)$ *time learning algorithms for the human and assistant that achieve no-*$(1 - 1/e)$*-approximate regret at a rate of* $\mathcal{O}\left(M_H^{5/4} M_A^{3/4} T^{3/4} \log(M_A T)\right)$.

Moreover, any assistant algorithm with minmax optimal tracking regret rates can be used to obtain this bound.

Our second theorem constructs decentralized algorithms that are more tailored to the assistance game and achieve a tighter $\mathcal{O}\left(\sqrt{T}\right)$ rate of $(1 - 1/e)$-approximate assistance regret. These algorithms require some initial synchronization between the human and assistant in the form of holding a shared mapping from sequences of human actions to the policy space.

Theorem 4.2. *There are* $\mathrm{poly}(M_H, M_A, N, T)$ *time learning algorithms for the human and assistant that make use of a shared map* $\phi : \mathcal{A}_H^* \to \Pi_H$ *from sequences of human actions to assistant policies and achieve no-*$(1 - 1/e)$*-approximate regret at a rate of* $\mathcal{O}\left(M_H M_A^{1/4} \sqrt{T \log M_A}\right)$.

We will describe the main ideas behind constructing these algorithms in this section, deferring the full proofs to Sections D.8 and D.9. But first, let us discuss the optimality of the regret guarantees. Note that guarantees we achieve in both settings of with and without initial synchronization is for assistance regret with a level of $(1-1/e)$-approximation. It turns out that it is not tractable to achieve sublinear assistance regret with a better approximation factor which is stated as Theorem 4.3.

Theorem 4.3. *Unless RP = NP, for any* $\alpha > 1 - 1/e$*, any algorithm that runs in time poly*(N, M_H, M_A) *per iteration has* α*-approximate assistance regret such that either* $R_T^\alpha \in \omega(T^{1-\epsilon})$ *for all* $\epsilon > 0$ *or* $R_T^\alpha \notin$ *poly*(N, M_H, M_A).

We prove this in Section C. Given this approximation factor, the regret bound in Theorem 4.2 has a $\mathcal{O}\left(\sqrt{T}\right)$ dependence on the number of rounds T, which is optimal due to standard minmax lower bounds for regret minimization. This is formalized for assistance games in Proposition E.6.

4.1. Central Challenges and Techniques

There are two central challenges that the human-assistant system needs to overcome to have low assistance regret. The first is *computational*. Namely, the space of human-assistant policy pairs is exponential is size. The second is a *coordination* challenge which is that the human and assistant need to learn jointly optimal strategies despite decentralized learning. We address these challenges in more detail in Section 5 and Section 6 respectively.

Challenge 1: Computational tractability of joint optimization (Section 5). The space of all joint human-assistant policy pairs consists of all pairs of mappings from Θ to $\mathcal{A}_H$ and $\mathcal{A}_H$ to $\mathcal{A}_A$. This space has size $M_H^N \cdot M_A^{M_H}$, which is exponential in the game dimensions N and M_H. To isolate this computational challenge from the coordination challenge, we first consider an idealized *centralized* setting (Definition A.1) where a meta-player jointly chooses both the human's and the assistant's policies. We provide a centralized algorithm with $\mathrm{poly}(M_H, M_A, N, T)$ runtime and prove that it has has sublinear $(1 - 1/e)$-assistance regret Section 5). The key insight is a reduction of joint policy optimization in assistance games to maximization of a special subclass of submodular functions—weighted threshold potentials (Definition A.2)—over a partition matroid (Definition A.3). This lets us apply algorithms from previous work on online convex optimization for submodular maximization (Salem et al., 2024).[1] Even in this idealized centralized setting, the approximation factor of $(1 - 1/e)$ is tight without compromising on the computational efficiency.

Challenge 2: Coordination through decentralized learning (Section 6). In the actual online assistance game, the human and assistant learn in a decentralized manner. That is, their strategies must recover the joint optimal without being jointly optimized. We call this the *coordination* challenge. We show that given any online algorithm for the centralized setting (Alg_C), and any online algorithm for the assistant's problem (Alg_A), we can construct an online algorithm in the decentralized setting for the human that overcomes the coordination challenge. In particular, we decompose the assistance regret of the decentralized pair of policies into the

[1] While any algorithm for online submodular optimization would be effective here, our choice to build upon (Salem et al., 2024) is guided by additional properties we will need for the decentralized setting. In particular, because this algorithm is based on online convex optimization we can make it such that the total number of steps where the selected policies switch is sublinear.

Algorithm 1. Generic algorithms in the centralized setting (Alg_C) and decentralized setting (Alg_H and Alg_A).

Alg_C **(centralized)**	Alg_A **(decentralized assistant)**	Alg_H **(decentralized human)**
1: Play $(\pi_H^{(t)}, \pi_A^{(t)})$ 2: Observe $\theta^{(t)}$ 3: Receive reward $r_t(\pi_H^{(t)}, \pi_A^{(t)})$	1: Observe $\pi_H^{(t)}(\theta^{(t)})$ 2: Play $\pi_A^{(t)}$ 3: Receive and see $r_t(\pi_H^{(t)}, \pi_A^{(t)})$	1: Observe $\theta^{(t)}$ 2: $(\overline{\pi}_H^{(t)}, \overline{\pi}_A^{(t)}) \leftarrow \mathrm{Alg}_C(\theta^{(1)}, \ldots, \theta^{(t-1)})$ 3: Play $\overline{\pi}_H^{(t)}(\theta^{(t)})$

sum of the centralized algorithm's external regret and a coordination cost that depends on tracking regret of Alg_A and the number of switches Alg_C makes in its choices. This is formalized in the following lemma.

Lemma 4.4. *Given any online algorithm Alg_C in the centralized setting, and any online algorithm Alg_A for the the assistant, there is a (decentralized) human algorithm Alg_H, that for every $\delta > 0$, achieve assistance regret of at most*

$$R_T^{\alpha} \leq R_T^{\alpha,ext}(\mathrm{Alg}_C) + R_T^{track}(\mathrm{Alg}_A; S_T(\mathrm{Alg}_C; \delta)) + \delta T,$$

where $S_T(\mathrm{Alg}_C; \delta)$ is an upper bound on the number of times Alg_C switches strategies that holds with probability at least $1 - \delta$ and $R_T^{\alpha,ext}, R_T^{track}$ denote α-approximate-external and (exact)-tracking regrets, respectively. Furthermore, Alg_H, described in Algorithm 1, is a simple wrapper around the centralized algorithm Alg_C.

The decomposition (proved in Section D.5) guides our algorithm design: we want a centralized algorithm Alg_C with low external regret *and* few switches (so $S_T(\mathrm{Alg}_C; \delta)$ is small), and an assistant algorithm Alg_A with low tracking regret. Both *stability* (few switches) and *adaptivity* (low tracking regret) are well-studied notions in the online learning literature. We exploit the fact that our centralized algorithm of Proposition 5.1 is based on online convex optimization to make it stable using standard techniques (Section 6.1, Proposition 6.1), and we construct an efficient adaptive assistant algorithm via standard tracking-regret machinery (Section 6.2).

5. Computationally Efficient Algorithms for Centralized Learning in Assistance Games

This section addresses the computational challenge identified in Section 4. We isolate it from the coordination challenge by studying an idealized *centralized* setting in which a meta-player jointly chooses both the human and assistant policies. We develop a centralized algorithm that achieves $(1 - 1/e)$-approximate assistance regret at a rate of $\mathcal{O}\left(\sqrt{T}\right)$ in $\mathrm{poly}(M_H, M_A, N, T)$ time, despite the exponentially large joint policy space. This algorithm will serve as the building block for the decentralized algorithm in Section 6.

In the centralized online assistance game (Definition A.1), a meta-player chooses a joint policy pair $(\pi_H^{(t)}, \pi_A^{(t)}) \in \Pi_H \times \Pi_A$ at each round before the realized preference $\theta^{(t)}$ is revealed, and receives reward $r(\pi_H^{(t)}, \pi_A^{(t)}; \theta^{(t)})$. The meta-player's external regret in this game—against the best fixed pair (π_H^*, π_A^*) in hindsight—is exactly the assistance regret (Definition 3.1), so any low-external-regret algorithm in the centralized game yields low assistance regret.

Our main result for the centralized setting is a tractable algorithm with sublinear $(1 - 1/e)$-approximate assistance regret.

Proposition 5.1. *There is a $\mathrm{poly}(M_H, M_A, N, T)$ time algorithm for the meta-player in the centralized assistance game (Definition A.1), such that the meta-player's expected $(1 - 1/e)$-assistance regret is at most $\mathcal{O}\left(M_H \sqrt{T \log M_A}\right)$.*

The main proof idea is a reduction of the assistance game to submodular maximization under matroid constraints, which lets us apply algorithms from prior work. We next develop this reduction in Section 5.1.

5.1. Reduction to online submodular maximization

To derive the reduction, we first represent the policy space as a partition matroid. Next, we show that the optimization objective over this representation satisfies structural properties that enable tractable online submodular optimization via the algorithms of Salem et al. (2024).

Policy space representation. We first note that joint optimization over human-asssistant policy pairs(π_H, π_A) reduces to optimization over assistant policies alone. Given any π_A, the meta-player can pair it with the best-response human policy $\pi_H^*(\theta) := \arg\max_{a_H} r(a_H, \pi_A(a_H); \theta)$. We represent each assistant policy $\pi_A : \mathcal{A}_H \to \mathcal{A}_A$ as the set of human-assistant action pairs $\{(a_H, \pi_A(a_H)) : a_H \in \mathcal{A}_H\}$ it induces, and the resulting policy space as a partition matroid.

Definition 5.2 (Assistance Matroid)**.** *Given an assistance game G with human action space $\mathcal{A}_H$ and assistant action space $\mathcal{A}_A$, define the* assistance matroid $\mathcal{M}_G = (\mathcal{U}_G, \mathcal{I}_G)$ *where:*

- $\mathcal{U}_G = \mathcal{A}_H \times \mathcal{A}_A$.
- $\mathcal{I}_G$ *consists of all* $S \subseteq \mathcal{U}_G$ *such that for any* $\left(a_H^{(1)}, a_A^{(1)}\right), \left(a_H^{(2)}, a_A^{(2)}\right) \in S$, *we have* $a_H^{(1)} \neq a_H^{(2)}$.

The assistance matroid is a partition matroid (Definition A.3) of rank M_H, with parts $\mathcal{U}_{a_H} = \{a_H\} \times \mathcal{A}_A$ and capacities $d_{a_H} = 1$ (Lemma D.1, proved in Section D.1).

Optimization objective and its structural properties. For each preference θ, the *value of assistance function* V_θ maps an independent set I (equivalently, an assistant policy) to the reward achieved by pairing it with the best-response human policy under θ. This is the objective function we maximize over the matroid representation of the policy space.

Definition 5.3 (Value of Assistance Function)**.** *Given an assistance game G, the value of assistance function for a human preference $\theta \in \Theta$ is the function $V_\theta : \mathcal{I}_G \to \mathbb{R}$ defined over independent sets $\mathcal{I}_G$ of the assistance matroid $\mathcal{M}_G = (\mathcal{U}_G, \mathcal{I}_G)$, such that $V_\theta(\emptyset) = 0$ and for all non-empty $S \in \mathcal{I}_G$,*

$$V_\theta(S) = \max_{(a_H, a_A) \in S} r(a_H, a_A; \theta).$$

The following proposition formalizes the reduction: the centralized online assistance game reduces to online maximization of value of assistance functions over the assistance matroid. The proof is deferred to Section D.2.

Proposition 5.4. *The centralized online assistance game G reduces to an online optimization over the family of value of assistance functions V_θ defined over independent sets of the assistance matroid $\mathcal{M}_G$. That is, the following conditions are met*

1. *For every independent set $I \in \mathcal{I}_G$ and preference $\theta \in \Theta$, there exists a human-assistant policy pair $\pi_H, \pi_A \in \Pi_H, \Pi_A$ achieving reward at least $V_\theta(I)$.*
2. *For every assistant policy $\pi_A \in \Pi_A$, there exists an independent set $I \in \mathcal{I}_G$ such that for all preferences $\theta \in \Theta$ and all human policies $\pi_H \in \Pi_H$, the reward the policy pair $\pi_H, \pi_A \in \Pi_H, \Pi_A$ achieves is at most $V_\theta(I)$.*

The value of assistance function further satisfies structural properties that enable tractable online optimization via the algorithms of Salem et al. (2024).

Lemma 5.5 (Structural properties of V_θ, informal)**.** *For every preference $\theta \in \Theta$, the value of assistance function V_θ is a weighted threshold potential (Definition A.2), and its concave relaxation (Definition A.4) is 1-Lipschitz.*

The formal statements (Lemmas D.2 and D.3) and proofs are deferred to Section D.3.

Centralized Algorithm. The reduction lets us apply the Randomized-Augmented OCO (RAOCO) algorithm of Salem et al. (2024) directly. RAOCO reduces online submodular maximization of weighted threshold potentials to online convex optimization. It runs a standard OCO algorithm (e.g., Follow the Perturbed Leader or Online Mirror Descent) on concave relaxations of the objectives over the convex hull of the strategy space, then rounds each iterate back to an integral point. The weighted threshold and Lipschitz properties of V_θ ensure that the rounding error is small, yielding the bound in Proposition 5.1.

The full proof, which combines Proposition 5.4 and the structural properties with the regret guarantees of RAOCO (restated in Proposition E.2), is deferred to Section D.4.

6. Decentralized Learning in Assistance Games: Stable and Adaptive Algorithms

In this section, we move beyond the idealized centralized setting and consider the decentralized setting. Lemma 4.4 forms the backbone for building on top of the algorithm design in the centralized setting. It shows how to construct an algorithm for the human Alg_H based on a centralized algorithm Alg_A via the construction in Algorithm 1. Additionally, it identifies stability of the centralized algorithm, in the form of having low switches, and adaptivity of the assistant's algorithm Alg_A, in the form of having low tracking regret, as sufficient properties for yielding low assistance regret. Directed by this, we will construct a stable centralized algorithm that has low external regret while making few switches in Section 6.1 and an adaptive algorithm for the assistant having low tracking regret in Section 6.2. In Section 6.3, we put together these components to prove the regret guarantees provided by our main theorems.

6.1. Stable centralized algorithm

In this section, we design a stable centralized algorithm that makes a small number of switches. We build on the centralized algorithm designed in Section 5.

Proposition 6.1. *There is a $\mathrm{poly}(M_H, M_A, N, T)$ time algorithm for the centralized assistance game that achieves expected $(1 - 1/e)$-approximate assistance regret $\mathcal{O}\left(\sqrt{M_A M_H T}\right)$ and makes at most $S_T \in O(\sqrt{M_A M_H T \log(1/\delta)})$ switches with probability at least $1 - \delta$.*

Proof sketch of Proposition 6.1. Recall that the centralized algorithm of Section 5 takes an online convex optimization algorithm and rounds the output of the OCO algorithm to the strategy space of the assistance game. To make this algorithm stable, we account for two sources of switches: switches in the OCO algorithm's iterates and switches in the rounding step.

Stable OCO. Online convex optimization with few switches has been studied by previous work (Agarwal et al.,

2024; Anava et al., 2015; Sherman & Koren, 2021). We use the Private Continuous Online Multiplicative Weights (P-COMW) algorithm of Agarwal et al. (2024) as the inner OCO optimizer, P-COMW requires only that the losses be convex and Lipschitz, conditions that hold for the value-of-assistance concave relaxation $\tilde{V}_\theta$ (Lemma D.3).

Coupled rounding. Even if the OCO algorithm's iterates are stable, the rounded outputs may not be, since the rounding algorithm used (like swap rounding or randomized pipage rounding (Chekuri et al., 2010)) is randomized. So even though the OCO algorithm returns the same solution in different rounds, the randomness in the rounding can lead to different rounded outputs. To overcome this, we couple the randomness of the rounding step across rounds. That is, the same source of randomness is used for rounding across all rounds. Doing this does not change the marginal distribution of the rounded output. In particular, it does not change the expected reward of the rounded output. So the property on the expected reward of the rounded output that is used in the regret analysis (the property in Proposition E.1) is preserved. This algorithm is described in Algorithm 3.

The full proof, which combines the regret guarantee of the underlying stable OCO algorithm with the switch-preserving property of coupled rounding, is given in Section D.6. □

6.2. Adaptive assistant algorithm

We now design a computationally efficient assistant algorithm with low tracking regret relative to the policies chosen by the stable centralized algorithm. Tracking regret is a well-studied notion in online learning, with standard algorithms achieving $\mathcal{O}\left(\sqrt{Tp}\right)$ regret against any sequence of reward functions over p segments. We lift these into an assistant algorithm with low tracking regret over the assistant policy space Π_A.

Proposition 6.2. *There is a* $\mathrm{poly}(M_H, M_A, N, T)$ *time assistant algorithm* Alg_A *whose tracking regret over* p *segments satisfies* $R_T^{\mathrm{track}}(\mathrm{Alg}_A; p) \in \mathcal{O}\left(M_H\sqrt{M_A T p \log(M_A T)}\right)$.

Proof sketch of Proposition 6.2. We construct Alg_A from an off-the-shelf tracking-regret minimizer $\mathrm{Alg}_{\mathrm{track}}$ (e.g. Fixed-Share (Herbster & Warmuth, 1998; Cesa-Bianchi et al., 2012)) used as a black box. The assistant maintains M_H independent copies of $\mathrm{Alg}_{\mathrm{track}}$, one per human action $a_H \in \mathcal{A}_H$, each optimizing over the assistant's action space $\mathcal{A}_A$. In round t, after observing $a_H^{(t)}$, the assistant plays the action selected by the copy associated with $a_H^{(t)}$ and updates only that copy with the observed reward. The assistant's tracking regret then decomposes as the sum of the tracking regrets of the M_H copies of $\mathrm{Alg}_{\mathrm{track}}$, which yields the claimed bound after substituting the standard Fixed-Share tracking-regret guarantee (restated as Theorem E.5 in Section E.3). The full proof appears in Section D.7.

□

6.3. Putting it together

We now combine the stable centralized algorithm of Section 6.1 and the adaptive assistant algorithm of Section 6.2 via the stable–adaptive decomposition of Lemma 4.4 to prove our main theorems. Full proofs are in Sections D.8 and D.9.

Proof idea for Theorem 4.1. The decentralized algorithms here are built entirely from general-purpose no-regret components: the human algorithm Alg_H is derived (via Algorithm 1) from the stable centralized algorithm Alg_C of Proposition 6.1, and the assistant runs the tracking-regret algorithm Alg_A of Proposition 6.2. Substituting Alg_C's external regret and high-probability switching bound together with Alg_A's tracking regret into the stable–adaptive decomposition (Lemma 4.4) yields the claimed $\mathcal{O}\left(M_H^{5/4} M_A^{3/4} \log(M_A T)^{3/4} T^{3/4}\right)$ bound; the full calculation appears in Section D.8. Note that this construction, and the assistant's algorithm in particular, requires minimal knowledge of the assistance game structure. The assistant merely needs to know its action space and its observed reward.

Proof idea for Theorem 4.2. We use algorithms more tailored to the assistance game for the optimal $\mathcal{O}\left(\sqrt{T}\right)$ rate. We allow the human and assistant some initial synchronization in the form of a shared mapping $\phi : \mathcal{A}_H^* \to \Pi_A$ from human action sequences to assistant policies, agreed on before the game begins. The mapping ϕ lets the human signal a policy switch and encode the new assistant policy via a string of actions. The assistant decodes this string using ϕ and then plays the new policy directly, bypassing further exploration and decreasing its tracking regret. Plugging this into the stable–adaptive decomposition (Lemma 4.4) yields the claimed $\mathcal{O}\left(\sqrt{T}\right)$ bound. The full proof, including details of how the shared mapping is used, appears in Section D.9.

7. Discussion

Our framework identifies two broad components useful for learning to cooperate under information asymmetry. The first is *submodularity* as a source of tractable approximation. This points to the usefulness of methods that exploit submodular structure—such as greedy algorithms that measure the marginal improvement of adding actions to communicate

or steer. The second component is *stability* and *adaptivity* as properties of the dynamics of interaction. Our decomposition (Lemma 4.4) shows that algorithms satisfying these standard online-learning notions plug in directly as building blocks for learning algorithms in these settings.

Limitations and Extensions. Our regret bounds are tight in T but the dependence on the action-space sizes M_H and M_A may be loose; determining the optimal scaling is an open question. The optimal $\mathcal{O}\left(\sqrt{T}\right)$ rate also relies on an initial shared encoding $\phi : \mathcal{A}_H^* \to \Pi_A$, and it is open whether the same rate is achievable without such pre-game synchronization. Finally, it would be interesting to extend the framework to richer interaction settings such as long horizon interactions under a fixed preference state, or two-sided information asymmetry in which the assistant also holds private information.

References

Agarwal, N., Kale, S., Singh, K., and Thakurta, A. G. Improved differentially private and lazy online convex optimization: Lower regret without smoothness requirements. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 343–361. PMLR, 2024.

Altschuler, J. and Talwar, K. Online learning over a finite action set with limited switching. In *Conference On Learning Theory*, pp. 1569–1573. PMLR, 2018.

Alur, R., Raghavan, M., and Shah, D. Human expertise in algorithmic prediction. *Advances in Neural Information Processing Systems*, 37:138088–138129, 2024.

Anava, O., Hazan, E., and Mannor, S. Online learning for adversaries with memory: price of past mistakes. In *Advances in Neural Information Processing Systems*, pp. 784–792, 2015.

Athey, S. C., Bryan, K. A., and Gans, J. S. The allocation of decision authority to human and artificial intelligence. In *AEA Papers and Proceedings*, volume 110, pp. 80–84. American Economic Association 2014 Broadway, Suite 305, Nashville, TN 37203, 2020.

Bansal, G., Wu, T., Zhou, J., Fok, R., Nushi, B., Kamar, E., Ribeiro, M. T., and Weld, D. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pp. 1–16, 2021.

Brown, W., Schneider, J., and Vodrahalli, K. Is learning in games good for the learners? *Advances in Neural Information Processing Systems*, 36, 2024.

Cesa-Bianchi, N., Gaillard, P., Lugosi, G., and Stoltz, G. Mirror descent meets fixed share (and feels no regret). *Advances in Neural Information Processing Systems*, 25, 2012.

Chaabouni, R., Strub, F., Altché, F., Tarassov, E., Tallec, C., Davoodi, E., Mathewson, K. W., Tieleman, O., Lazaridou, A., and Piot, B. Emergent communication at scale. In *International conference on learning representations*, 2022.

Chekuri, C., Vondrák, J., and Zenklusen, R. Dependent randomized rounding via exchange properties of combinatorial structures. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 575–584. IEEE, 2010.

Chu, F. and Halpern, J. On the np-completeness of finding an optimal strategy in games with common payoffs. *International Journal of Game Theory*, 30:99–106, 2001.

Collina, N., Globus-Harris, I., Goel, S., Gupta, V., Roth, A., and Shi, M. Collaborative prediction: Tractable information aggregation via agreement. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 4712–4798. SIAM, 2026.

Conitzer, V. and Sandholm, T. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pp. 82–90, 2006.

Daniely, A., Gonen, A., and Shalev-Shwartz, S. Strongly adaptive online learning. In *International Conference on Machine Learning*, pp. 1405–1411. PMLR, 2015.

Donahue, K., Chouldechova, A., and Kenthapadi, K. Human-algorithm collaboration: Achieving complementarity and avoiding unfairness. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1639–1656, 2022.

Feige, U. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

Fern, A., Natarajan, S., Judah, K., and Tadepalli, P. A decision-theoretic model of assistance. *Journal of Artificial Intelligence Research*, 50:71–104, 2014.

Foerster, J. N., Assael, Y. M., de Freitas, N., and Whiteson, S. Learning to communicate to solve riddles with deep distributed recurrent q-networks. *arXiv preprint arXiv:1602.02672*, 2016.

Franke, M. Interpretation of optimal signals. *New perspectives on games and interaction*, pp. 297–310, 2009a.

Franke, M. *Signal to act: Game theory in pragmatics*. University of Amsterdam, 2009b.

Gilboa, I. and Zemel, E. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1(1):80–93, 1989.

Green, B. and Chen, Y. The principles and limits of algorithm-in-the-loop decision making. *Proceedings of the ACM on human-computer interaction*, 3(CSCW):1–24, 2019.

Greenwood, S., Levy, K., Barocas, S., Heidari, H., and Kleinberg, J. Designing algorithmic delegates: The role of indistinguishability in human-ai handoff. In *Proceedings of the 26th ACM Conference on Economics and Computation*, pp. 306–336, 2025.

Hadfield-Menell, D., Russell, S. J., Abbeel, P., and Dragan, A. Cooperative inverse reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

Hadfield-Menell, D., Dragan, A. D., Abbeel, P., and Russell, S. The off-switch game. In *AAAI Workshops*, 2017.

Havrylov, S. and Titov, I. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *Advances in neural information processing systems*, 30, 2017.

Hazan, E. and Seshadhri, C. Adaptive algorithms for online decision problems. In *Electronic colloquium on computational complexity (ECCC)*, volume 14, 2007.

Herbster, M. and Warmuth, M. K. Tracking the best expert. *Machine learning*, 32(2):151–178, 1998.

Ho, Y.-C. et al. Team decision theory and information structures in optimal control problems–part i. *IEEE Transactions on Automatic control*, 17(1):15–22, 1972.

Jacob, A. P., Farina, G., and Andreas, J. Regularized conventions: Equilibrium computation as a model of pragmatic reasoning. *arXiv preprint arXiv:2311.09712*, 2023.

Jäger, G. Game dynamics connects semantics and pragmatics. In *Game theory and linguistic meaning*, pp. 103–117. Brill, 2007.

Jäger, G. Game theory in semantics and pragmatics. *Semantics: An international handbook of natural language meaning*, 3:2487–2516, 2012.

Kapralov, M., Post, I., and Vondrák, J. Online submodular welfare maximization: Greedy is optimal. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1216–1225. SIAM, 2013.

Kim, J. and Oh, A. Emergent communication under varying sizes and connectivities. *Advances in Neural Information Processing Systems*, 34:17579–17591, 2021.

Kirby, S. Natural language from artificial life. *Artificial life*, 8(2):185–215, 2002.

Kirby, S., Griffiths, T., and Smith, K. Iterated learning and the evolution of language. *Current opinion in neurobiology*, 28:108–114, 2014.

Laidlaw, C., Bronstein, E., Guo, T., Feng, D., Berglund, L., Svegliato, J., Russell, S., and Dragan, A. Assistancezero: Scalably solving assistance games. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=b9hVMJi0t2.

Lazaridou, A. and Baroni, M. Emergent multi-agent communication in the deep learning era. *arXiv preprint arXiv:2006.02419*, 2020.

Lazaridou, A., Peysakhovich, A., and Baroni, M. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182*, 2016.

Letchford, J., Conitzer, V., and Munagala, K. 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*, pp. 250–262. Springer, 2009.

Li, F. and Bowling, M. Ease-of-teaching and language structure from emergent communication. *Advances in neural information processing systems*, 32, 2019.

Lovász, L. and Vempala, S. The geometry of logconcave functions and sampling algorithms. *Random Structures & Algorithms*, 30(3):307–358, 2007.

Mahajan, A. and Mannan, M. Decentralized stochastic control. *Annals of Operations Research*, 241(1):109–126, 2016.

Malik, D., Palaniappan, M., Fisac, J., Hadfield-Menell, D., Russell, S., and Dragan, A. An efficient, generalized bellman update for cooperative inverse reinforcement learning. In *International Conference on Machine Learning*, pp. 3394–3402. PMLR, 2018.

Malikopoulos, A. A. On team decision problems with nonclassical information structures. *IEEE Transactions on Automatic Control*, 68(7):3915–3930, 2022.

Nayyar, A., Mahajan, A., and Teneketzis, D. Decentralized stochastic control with partial history sharing: A common information approach. *IEEE Transactions on Automatic Control*, 58(7):1644–1658, 2013.

Peng, B., Shen, W., Tang, P., and Zuo, S. Learning optimal strategies to commit to. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 2149–2156, 2019.

Radner, R. Team decision problems. *The Annals of Mathematical Statistics*, 33(3):857–881, 1962.

Ren, Y., Guo, S., Havrylov, S., Cohen, S., and Kirby, S. Enhance the compositionality of emergent language by iterated learning. In *3rd NeurIPS Workshop on Emergent Communication (EmeCom@ NeurIPS 2019). URL https://papers. nips. cc/book/advances-in-neural-information-processing-systems-32-2019*, 2019.

Rita, M., Chaabouni, R., and Dupoux, E. " lazimpa": Lazy and impatient neural agents learn to communicate efficiently. *arXiv preprint arXiv:2010.01878*, 2020.

Salem, T. S., Özcan, G., Nikolaou, I., Terzi, E., and Ioannidis, S. Online submodular maximization via online convex optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 15038–15046, 2024.

Shah, R., Freire, P., Alex, N., Freedman, R., Krasheninnikov, D., Chan, L., Dennis, M. D., Abbeel, P., Dragan, A., and Russell, S. Benefits of assistance over reward learning, 2020.

Shalev-Shwartz, S. et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Sherman, U. and Koren, T. Lazy oco: Online convex optimization on a switching budget. In *Conference on Learning Theory*, pp. 3972–3988. PMLR, 2021.

Steyvers, M., Tejeda, H., Kerrigan, G., and Smyth, P. Bayesian modeling of human–ai complementarity. *Proceedings of the National Academy of Sciences*, 119(11): e2111547119, 2022.

Trapa, P. E. and Nowak, M. A. Nash equilibria for an evolutionary language game. *Journal of mathematical biology*, 41(2):172–188, 2000.

Wilder, B., Horvitz, E., and Kamar, E. Learning to complement humans. *arXiv preprint arXiv:2005.00582*, 2020.

Zrnic, T., Mazumdar, E., Sastry, S., and Jordan, M. Who leads and who follows in strategic classification? *Advances in Neural Information Processing Systems*, 34: 15257–15269, 2021.

A. Extended Model and Preliminaries

This appendix presents the two special subclasses of submodular functions and matroids that arise in our reduction of assistance games, together with the concave relaxation we use to optimize over them. We also state the formal definition of the centralized online assistance game referenced in Section 5.

Centralized online assistance game.

Definition A.1 (Centralized online assistance game)**.** *The centralized online assistance game is a repeated game between nature and a meta-player who controls both the assistant and human. At every round t, both players take actions simultaneously.*[2] *That is, at the start of the interaction, nature chooses a sequence of preferences $\theta^{(1)}, \cdots, \theta^{(T)} \in \Theta$. At every iteration, the meta-player chooses a pair of policies $\left(\pi_H^{(t)}, \pi_A^{(t)}\right)$, then the realized preference $\theta^{(t)}$ is revealed to the meta-player, and the meta-player receives a reward $r\left(\pi_H^{(t)}, \pi_A^{(t)}; \theta^{(t)}\right)$.*

Weighted threshold potentials. A useful subclass of submodular functions consists of those that can be written as a sum of capped linear functions.

Definition A.2 (Weighted Threshold Potential)**.** *Fix a set of elements $\mathcal{U}$. A set function g over $\mathcal{U}$ is modular when each element $i \in \mathcal{U}$ is associated with a weight w_i such that $g(S) = \sum_{j \in S} w_j$. A set function f over $\mathcal{U}$ is a threshold potential function, also known as a budget-additive set function, when*

$$f(S) = \min\{b, g(S)\}$$

for some $b \in \mathbb{R}$ and some monotone modular function g. A set function f over $\mathcal{U}$ is a weighted threshold potential when it is a non-negative linear combination of threshold potential functions.

All weighted threshold potentials are submodular (Salem et al., 2024).

Partition matroids. A useful subclass of matroids in our setting is the partition matroid: the set of policies in an assistance game forms a matroid of this form.

Definition A.3 (Partition Matroids)**.** *A matroid $(\mathcal{U}, \mathcal{I})$ is a partition matroid when there exists some $k \in \mathbb{N}$, a partition of $\mathcal{U}$ into subsets $\mathcal{U}_1, \ldots, \mathcal{U}_k$, and integers $d_1, \ldots, d_k$ such that a set I is independent if and only if*

$$|I \cap \mathcal{U}_i| \leq d_i.$$

Concave relaxation. To optimize over the space of weighted threshold potential functions, we follow Salem et al. (2024), which runs an online convex optimization algorithm on a concave relaxation of these functions.

Definition A.4 (Concave relaxation of weighted threshold potentials)**.** *Given a weighted threshold potential $f(S) = \sum_{\ell=1}^{L} c_\ell \min\{b_\ell, \sum_{j \in S} w_{\ell,j}\}$ defined over $\mathcal{I} \subseteq 2^{\mathcal{U}}$, its* concave relaxation *$\tilde{f} : [0,1]^{|\mathcal{U}|} \to \mathbb{R}$ is*

$$\tilde{f}(\mathbf{y}) = \sum_{\ell=1}^{L} c_\ell \min\left\{ b_\ell, \sum_{j \in \mathcal{U}} w_{\ell,j}\, y_j \right\}.$$

This function is concave, since each $\min\{b_\ell, \cdot\}$ applied to an affine function is concave, and a positive linear combination of concave functions is concave. Moreover, $\tilde{f}$ agrees with f on integral points: $\tilde{f}(\mathbb{1}_S) = f(S)$ for all $S \subseteq \mathcal{U}$.

B. An Adaptive Nature Player Is Too Powerful

A key assumption in this work is that nature cannot adaptively pick preferences based on the choices of the human and assistant. Nature becomes too powerful otherwise, removing any hope the learners have of achieving sub-linear regret. This

[2] This means the meta-player chooses a pair of actions for the human and assistant without having access to the realized state. We do this to make the idealized setting serve as a building block for the decentralized setting where the assistant must choose a policy without having access to the realized state.

is analogous to results in the prediction from experts with switching costs literature, where the learner must suffer linear regret with an adaptive adversary (Altschuler & Talwar, 2018).

Nature's strategy will be to exploit the fact that the human and assistant play moves that are uncorrelated given the history of the game, which nature also has access to. By being adaptive, nature can exploit uncoordinated agents by taking advantage of the fact that it is impossible to randomize over best-response pairs in a way that beats an adaptive nature player when actions are independently chosen.

We will prove an adaptive nature player is too powerful, even for a very simple and restricted class of assistance games. Consider the class of games where the assistant's action set equals the set of preferences, and the agents only get a reward when the assistant can reproduce the human's preference. The human's action has no effect on the reward, and serves only to signal the preference to the assistant. Even on such simple games, when nature can adversarially choose rewards, the agents must suffer linear regret.

First, we prove a lower bound when the human has one less action than the number of preferences, so there does not exist a perfect signaling scheme the human can use to exactly reveal their preferences.

Lemma B.1. *When nature is adaptive, there exist assistance games where $N \geq 3$ and $M_H = N - 1$, such that any learning algorithm must achieve a regret of at least $\frac{N-2}{2N} \cdot T$.*

Proof. Consider the following assistance game. The set of preferences is an arbitrary set Θ of size N, the set of human actions is an arbitrary set $\mathcal{A}_H$ of size $N-1$, and the set of assistant actions is the set of preferences $\mathcal{A}_A = \Theta$. The reward function is defined as $r(a_H, \theta'; \theta) = \mathbb{1}(\theta = \theta')$.

First we will lower bound the utility of the optimal human and assistant strategy in hindsight. Notice that regardless of the preferences chosen by nature, there must always exist a human-assistant policy pair that achieves $\frac{N-1}{N}T$ utility in hindsight. Indeed, let $\theta^{(1)}, \ldots, \theta^{(T)}$ be the sequence of preferences played by the adversary. There must exist a preference θ that appears at most $\frac{T}{N}$ times. The rest of the states must appear at least $\frac{N-1}{N} \cdot T$ times. Let π_H be the human policy that assigns to each of these preferences a unique one of the $N-1$ messages, and π_A be the robot policy that reverses this. This correctly recovers the state when θ does not appear, so achieves a utility of at least $\frac{N-1}{N} \cdot T$.

Now we will lower bound the utility the agents can achieve during the learning process. On step t over the T steps, let $\mathcal{H}_t = \left(\theta^{(1)}, \pi_H^{(1)}, \pi_A^{(1)}, \ldots, \theta^{(t-1)}, \pi_H^{(t-1)}, \pi_A^{(t-1)}\right)$ be the history of preferences, human policies, and assistant policies played. The key property that we will exploit is that the policies the human and assistant play are independent given the history, so that $\pi_H^{(t)} \perp \pi_A^{(t)} \mid \mathcal{H}_t$.

Given a $\pi_H^{(t-1)}$ and $\pi_R^{(t-1)}$, what preference should the adversary play? The expected utility that the human and assistant achieve, when the adversary plays state $\theta \in \Theta$ is the probability that the agents correctly recover θ:

$$\begin{aligned}
\mathbb{P}_{\pi_H^{(t)}, \pi_A^{(t)}}\left(\theta = \pi_A^{(t)}\left(\pi_H^{(t)}(\theta)\right) \mid \mathcal{H}_t\right) &= \sum_{a_H \in \mathcal{A}_H} \mathbb{P}_{\pi_H^{(t)}, \pi_A^{(t)}}\left(\theta = \pi_A^{(t)}(a_H) \text{ and } a_H = \pi_H^{(t)}(\theta) \mid \mathcal{H}_t\right) \\
&= \sum_{a_H \in \mathcal{A}_H} \mathbb{P}_{\pi_A^{(t)}}\left(\theta = \pi_A^{(t)}(a_H) \mid \mathcal{H}_t\right) \mathbb{P}_{\pi_H^{(t)}}\left(a_H = \pi_H^{(t)}(\theta) \mid \mathcal{H}_t\right) \\
&= \mathop{\mathbb{E}}_{a_H \sim \pi_H^{(t)}(\theta) \mid \mathcal{H}_t}\left[\mathbb{P}_{\pi_A^{(t)}}\left(\theta = \pi_A^{(t)}(a_H) \mid \mathcal{H}_t\right)\right].
\end{aligned}$$

We will show that there must always exist a preference $\theta \in \Theta$ such that

$$\mathop{\mathbb{E}}_{a_H \sim \pi_H^{(t)}(\theta) \mid \mathcal{H}_t}\left[\mathbb{P}_{\pi_A^{(t)}}\left(\theta = \pi_A^{(t)}(a_H) \mid \mathcal{H}_t\right)\right] \leq \frac{1}{2}.$$

Order the preferences $\theta_1, \ldots, \theta_N$. If this is true for any of the first $N-1$ preferences, we are done. Otherwise, it is the case that for all i from 1 to $N-1$,

$$\mathop{\mathbb{E}}_{a_H \sim \pi_H^{(t)}(\theta) \mid \mathcal{H}_t}\left[\mathbb{P}_{\pi_A^{(t)}}\left(\theta_i = \pi_A^{(t)}(a_H) \mid \mathcal{H}_t\right)\right] > \frac{1}{2}.$$

Therefore, for each i from 1 to $N-1$, there must exist some $a_{H,i} \in \mathcal{A}_H$ such that

$$\mathbb{P}_{\pi_A^{(t)}}\left(\theta_i = \pi_A^{(t)}(a_{H,i}) \mid \mathcal{H}_t\right) > \frac{1}{2}. \tag{1}$$

For any i from 1 to n, for any preference $\theta \in \Theta$ with $\theta \neq \theta_i$:

$$\begin{aligned}\mathbb{P}_{\pi_A^{(t)}}\left(\theta = \pi_A^{(t)}(a_{H,i}) \mid \mathcal{H}_t\right) &= 1 - \mathbb{P}_{\pi_A^{(t)}}\left(\theta \neq \pi_A^{(t)}(a_{H,i}) \mid \mathcal{H}_t\right) \\ &\leq 1 - \mathbb{P}_{\pi_A^{(t)}}\left(\theta_i = \pi_A^{(t)}(a_{H,i}) \mid \mathcal{H}_t\right) \\ &< 1 - \frac{1}{2} = \frac{1}{2}.\end{aligned}$$

So, for any preference θ that is not θ_i,

$$\mathbb{P}_{\pi_A^{(t)}}\left(\theta = \pi_A^{(t)}(a_{H,i}) \mid \mathcal{H}_t\right) < \frac{1}{2}. \tag{2}$$

By Equations (1) and (2), no two $a_{H,i}$ can be equal. Since this is true for $N-1 = M_H$ preferences, and every preference is assigned a unique human action by the above, every human action appears in the set of $a_{H,i}, \ldots, a_{H,N-1}$. That is, there is a perfect matching between the set of all human actions and the first $N-1$ preferences. When an action $a_{H,i}$ is played, it must correspond to some preference θ_i with high probability. Therefore, for the final preference θ_N, every action is unlikely to correspond to it, i.e. for every action $a_H \in \mathcal{A}_H$,

$$\mathbb{P}_{\pi_A^{(t)}}\left(\theta_N = \pi_A^{(t)}(a_H) \mid \mathcal{H}_t\right) \leq \frac{1}{2},$$

and so

$$\mathop{\mathbb{E}}_{a_H \sim \pi_H^{(t)}(\theta) \mid \mathcal{H}_t}\left[\mathbb{P}_{\pi_A^{(t)}}\left(\theta_N = \pi_A^{(t)}(a_H) \mid \mathcal{H}_t\right)\right] \leq \frac{1}{2}.$$

So, the adversary can always play a preference θ that forces the expected utility the agents achieve to be at most $\frac{1}{2}$. Whereas in hindsight, the agents could have achieved an average of $\frac{N-1}{N}$ per step, resulting in an expected regret of at least:

$$\mathbb{E}\left[\max_{\pi_H, \pi_A} \sum_{t=1}^{T} \mathbb{1}(\theta^{(t)} = \pi_A(\pi_H(\theta^{(t)})) - \sum_{t=1}^{T} \mathbb{1}\left(\theta^{(t)} = \pi_A^{(t)}\left(\pi_H^{(t)}(\theta^{(t)})\right)\right)\right] \geq \frac{N-1}{N} \cdot T - \frac{1}{2} \cdot T = \frac{N-2}{2N} \cdot T. \quad \square$$

To extend this result to the case where the number of messages is arbitrary, the adversary can commit beforehand to only using a subset of the states. It is interesting that even if the learners know which subset of the states the adversary has committed to using, they still cannot achieve sublinear regret.

Theorem B.2. *When nature is adaptive, there exist assistance games with $N \geq 3$ and $1 < M_H < N$, such that any learning algorithm must suffer a regret of at least $\frac{M_H - 1}{2M_H + 2} T$.*

Proof. When M_H may be arbitrary, the adversary may just commit upfront to only sending $M_H + 1 \leq N$ states, fixed at the start arbitrarily. Since $M_H \geq 2$, this means that there are $M_H + 1 \geq 3$ states, so the lower bound in Lemma B.1 applies directly. $\square$

For the learning algorithms we derive for assistance games to be nontrivial, we then must assume that the adversary is oblivious to the actions of the human and assistant.

C. A Lower Bound on Assistance Regret

In this section we prove Theorem 4.3, that efficiently achieving an approximation ratio better than $1 - 1/e$ is impossible unless P = NP.

First, it will be helpful to consider the offline case, where preferences are drawn from a distribution $\mathcal{D}$ over preferences, and a human and assistant policy pair π_H, π_A must be outputted to maximize the expected reward when preferences are drawn from $\mathcal{D}$:

$$\mathop{\mathbb{E}}_{\theta \sim \mathcal{D}}[r(\pi_H(\theta), \pi_A(\pi_H(\theta)); \theta)].$$

We start by showing that achieving an approximation better than $1 - 1/e$ in the offline problem is NP-hard.

Lemma C.1 (Assistance is NP-hard)**.** *Computing any (potentially stochastic) human-assistant policy pair that is an α-approximation of the optimal strategy, where $\alpha > 1 - 1/e$, in an offline assistance game, even with rewards restricted to be 0 or 1, is NP-hard.*

Proof. We will in fact show this offline hardness result for games where the human's action has no effect on utility.

Recall that the problem of finding α-approximations to the max k-coverage problem with $\alpha > 1 - 1/e$ is NP-hard (Feige, 1998). Then, we will reduce this to finding an α-approximation of optimal play in an assistance game. The rewards in this game will be 0 or 1 only.

We start by recapping the k-maximum coverage problem. Given a value k and a collection of sets $S = \{S_1, \ldots, S_m\}$ each with elements in a universe $\mathcal{U}$, find a collection of k sets $S_{i_1}, \ldots, S_{i_k}$ such that $\left|\bigcup_{\ell=1}^{k} S_{i_\ell}\right|$ is maximized. Feige (1998) proved that it is NP-hard to get $(1 - 1/e)$-approximations of the maximum k coverage of S.

We now describe a reduction from a maximum coverage problem to solving offline assistance games. Consider an arbitrary k-maximum coverage problem over a collection of sets $S = \{S_1, \ldots, S_m\}$ with elements in $\mathcal{U}$. Create an assistance game where the human's action set is an arbitrary set of size $M_H = k$, and the space of preferences is $\mathcal{U}$. Make the assistant's action space S, so that each S_i is an assistant action, Set the reward function to

$$r(a_H, S_i; e) = \begin{cases} 1 & \text{if } e \in S_i, \\ 0 & \text{otherwise.} \end{cases}$$

The human's action has no effect on utility and serves only to signal the preference.

Let the underlying distribution $\mathcal{D}$ assign an equal probability to each state, $1/\left|\mathcal{U}\right|$.

We will prove two useful facts about this game.

1. Every solution to the max-M-coverage problem induces a solution to this assistance game with the same value. More precisely, every collection of sets $S_{i_1}, \ldots, S_{i_M}$, with union $S = \bigcup_{j=1}^{M} S_{i_j}$, induces an human-assistant policy pair π_H, π_A that achieves utility that is at least the value of the sets in the maximum-M-coverage problem: $|S| \,/\, |\mathcal{U}|$. Indeed, let the human sends message m_j when observing a state in $S_{i_j} \setminus \bigcup_{\ell=1}^{j-1} S_{i_\ell}$. The assistant plays set S_{i_j} upon receiving message m_j. Let $S = \bigcup_{j=1}^{M} S_i$. Whenever a preference in S appears, the agents get a reward of 1, and so the expected reward of this human-assistant policy pair is $|S| \,/\, |\mathcal{U}|$, precisely the value of the sets in the maximum-M-coverage problem.

2. Every solution with a deterministic assistant policy to this assistance game induces a solution to the max-M-coverage problem with at least the same value. More precisely, given a deterministic assistant policy π_A, we will construct a collection of sets $S_{i_1}, \ldots, S_{i_M}$, with union $S = \bigcup_{j=1}^{M} S_{i_j}$, so that for any human policy π_H, the utility π_H, π_A achieves utility is at most the value of the sets in the maximum-M-coverage problem: $|S| \,/\, |\mathcal{U}|$.

   π_A maps each message to an action S_i. There are M_H possible messages, so simply take the M_H actions, $S_{i_1}, \ldots, S_{i_M}$, that the assistant chooses to play as the solution to the maximum coverage problem. Let $S = \bigcup_{j=1}^{M} S_{i_j}$. The value π_A achieves with any human policy π_H is at most the probability that elements in S appear: $|S| \,/\, |\mathcal{U}|$.

These reductions back and forth imply that the optimal values of both problems are equal.

In general assistance games, given a human policy π_H that is not necessarily deterministic, there exists a deterministic best-response assistant policy π_A^* that can be found by an efficient algorithm. Indeed, given a human action $a_H \in \mathcal{A}_H$ the value of playing action $a_A \in \mathcal{A}_A$ for the assistant is

$$\begin{aligned} \mathop{\mathbb{E}}_{\theta \sim \mathcal{D}}[r(a_H, a_A, \theta) \mid \pi_H(\theta) = a_H] &= \sum_{\theta \in \Theta} \mathbb{P}(\theta \mid \pi_H(\theta) = a_H) r(a_H, a_A, \theta) \\ &= \sum_{\theta \in \Theta} \frac{\mathbb{P}(\theta)\mathbb{P}(\pi_H(\theta) = a_H)}{\sum_{\theta' \in \Theta} \mathbb{P}(\theta')\mathbb{P}(\pi_H(\theta') = m)} r(a_H, a_A, \theta), \end{aligned}$$

which can be computed in polynomial time in the number of actions and states. The reward any response achieves is linear, and the optimal deterministically plays the action with largest expected reward given the human action observed.

Using these two facts, we can reduce the problem of finding an α-approximation of the k-maximum coverage problem to finding an α-approximation of the optimal human-assistant policy pair in this assistance game. Suppose (π_H, π_A) are a human-assistant policy pair that achieve an α-approximation of optimal play in this assistance game. (π_H, π_R) may not necessarily be deterministic. But, by the above, we can construct in polynomial time a deterministic assistant policy π'_A that achieves at least the same reward when paired with π_H.

Applying fact 2 on π'_A this means that we have found a collection of sets $S_{i_1}, \ldots, S_{i_k}$ that achieve at least α times the max reward of the assistance game. So we have shown that the optimal reward in this game is equal to the optimal reward in the maximum-k-coverage problem, and we are done! □

With this tool in hand, we can show that, unless RP = NP, efficient learning algorithms for assistance games that achieve approximate utility better than $1 - 1/e$ cannot exist.

Theorem 4.3. *Unless RP = NP, for any $\alpha > 1 - 1/e$, any algorithm that runs in time poly(N, M_H, M_A) per iteration has α-approximate assistance regret such that either $R_T^\alpha \in \omega(T^{1-\epsilon})$ for all $\epsilon > 0$ or $R_T^\alpha \notin$ poly(N, M_H, M_A).*

Proof of Theorem 4.3. Suppose for the sake of contradiction that $R_T \in \mathcal{O}\left(T^{1-\epsilon}\right)$ for some $\epsilon > 0$.

We can write for some constants $c_1, c_2, a, b, c \in \mathbb{R}$, that $R_T(N, M) \leq c_1 N^a M_H^b M_A^c T^{1-\epsilon} + c_2$. This means that the average regret per time step is at most

$$\frac{c_1 N^a M_H^b M_A^c}{T^\epsilon} + \frac{c_2}{T}.$$

The rest of the proof will follow the lead of Kapralov et al. (2013). Given a value $k \in \mathbb{N}$ and a collection of sets $S = \{S_1, \ldots, S_m\}$ each with elements in a universe $\mathcal{U}$, Feige (1998) proved that, for any $\delta > 0$, it is NP-hard to distinguish between the following two cases:

1. Yes case: Some choice of k sets covers $\mathcal{U}$.
2. No case: No choice of k sets covers a $1 - 1/e + \delta$ proportion of the elements in $\mathcal{U}$.

Therefore, if there is a polynomial-time algorithm that outputs yes with constant probability when in the yes case, and outputs no always when in the no case, it must be that RP = NP.

Suppose there exists an efficient no-α-approximate regret assistance game learning algorithm. We will show that the problem above for $\delta = \frac{\alpha - (1 - 1/e)}{2}$ can be solved with a constant probability of success in the yes case, and always in the no case. This would prove RP = NP.

Given an instance of the set-cover problem, run the algorithm on the corresponding assistance game derived in Lemma C.1. As we show in Lemma C.1, the optimal value in this game is equal to that of the set-cover problem.

Run the no-α-approximate regret learning algorithm for

$$T = \max\left(c_2, \left(\left(\frac{\alpha - (1 + 1/e)}{4}\right) c_1 N^a M_H^b M_A^c\right)^{1/\epsilon}\right) \in \text{poly}(N, M)$$

steps, and the average regret per iteration becomes at most $\frac{\alpha - (1 - 1/e)}{4}$.

First, notice that we may as well pretend that the human and assistant output their full policy at every step $\pi_H^{(t)}$ and $\pi_A^{(t)}$ at every step, not just $\pi_H^{(t)}(\theta^{(t)})$ and $\pi_A^{(t)}(\pi_H^{(t)}(\theta^{(t)}))$. This is because at every step of the learning algorithm, we can create M_H and M_A copies and ask for the distribution over actions on every counterfactual input.

The reduction will be as follows. Run the algorithm on T i.i.d draws $\theta^{(1)}, \ldots, \theta^{(T)} \sim \mathcal{D}^T$, induce a sequence of human-assistant policy pairs $(\pi_H^{(1)}, \pi_A^{(1)}), \ldots, (\pi_H^{(T)}, \pi_A^{(T)})$, compute the expected reward of each human-assistant policy pair

averaged over $\mathcal{D}$ (which we can do in at most $\mathcal{O}(N)$ time each), and output yes if some pair $\pi_H^{(t)}, \pi_A^{(t)}$ exists with expected reward at least $\alpha - \frac{3(\alpha-(1-1/e))}{8}$, and no otherwise.

When we are in the no case, no solution will have expected reward in the assistance game that is at least $\alpha - \frac{3(\alpha-(1-1/e))}{8}$, so we will always output no.

Suppose we are in the yes case, so that the optimal utility is 1. Because we run for long enough that average regret is $\frac{\alpha-(1-1/e)}{4}$, we can say

$$\mathop{\mathbb{E}}_{\theta^{1:T}\sim\mathcal{D}^T}\left[\frac{1}{T}\sum_{t=1}^{T} r\left(\pi_H^{(t)}(\theta^{(t)}),\ \pi_A^{(t)}(\pi_H^{(t)}(\theta^{(t)}));\ \theta^{(t)}\right)\right] \geq \alpha - \frac{\alpha-(1-1/e)}{4}. \tag{3}$$

To move forward, we will need to remove the dependence on $\theta^{(t)}$ in the expected reward. Notice that $\theta^{(t)} \perp (\pi_H^{(t)}, \pi_A^{(t)})$, as $\theta^{(t)}$ is sampled from $\mathcal{D}$, and $\pi_H^{(t)}, \pi_A^{(t)}$ depends only on $\theta^{(1)}, \ldots, \theta^{(t-1)}$, which themselves are independent of $\theta^{(t)}$. Therefore, for any t, it must be that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\theta^{1:T}\sim\mathcal{D}^T}\left[r\left(\pi_H^{(t)}(\theta^{(t)}),\ \pi_A^{(t)}(\pi_H^{(t)}(\theta^{(t)}));\ \theta^{(t)}\right)\right] \\
&= \mathop{\mathbb{E}}_{\theta^{1:(t-1)}\sim\mathcal{D}^{(t-1)}}\ \mathop{\mathbb{E}}_{\theta^{(t)}\sim\mathcal{D}}\left[r\left(\pi_H^{(t)}(\theta^{(t)}),\ \pi_A^{(t)}(\pi_H^{(t)}(\theta^{(t)}));\ \theta^{(t)}\right) \mid \theta^{(1)}, \ldots, \theta^{(t-1)}\right] \\
&= \mathop{\mathbb{E}}_{\theta^{1:(t-1)}\sim\mathcal{D}^{(t-1)}}\ \mathop{\mathbb{E}}_{\theta\sim\mathcal{D}}\left[r\left(\pi_H^{(t)}(\theta),\ \pi_A^{(t)}(\pi_H^{(t)}(\theta));\ \theta\right)\right].
\end{aligned} \tag{4}$$

And so by the linearity of expectation,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\theta^{1:T}\sim\mathcal{D}^T}\left[\frac{1}{T}\sum_{t=1}^{T} r\left(\pi_H^{(t)}(\theta^{(t)}),\ \pi_A^{(t)}(\pi_H^{(t)}(\theta^{(t)}));\ \theta^{(t)}\right)\right] \\
&= \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\theta^{1:T}\sim\mathcal{D}^T}\left[r\left(\pi_H^{(t)}(\theta^{(t)}),\ \pi_A^{(t)}(\pi_H^{(t)}(\theta^{(t)}));\ \theta^{(t)}\right)\right] \\
&= \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\theta^{1:(t-1)}\sim\mathcal{D}^{(t-1)}}\ \mathop{\mathbb{E}}_{\theta\sim\mathcal{D}}\left[r\left(\pi_H^{(t)}(\theta),\ \pi_A^{(t)}(\pi_H^{(t)}(\theta));\ \theta\right)\right] \qquad \text{(By Equation (4))} \\
&= \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\theta^{1:T}\sim\mathcal{D}^T}\ \mathop{\mathbb{E}}_{\theta\sim\mathcal{D}}\left[r\left(\pi_H^{(t)}(\theta),\ \pi_A^{(t)}(\pi_H^{(t)}(\theta));\ \theta\right)\right] \\
&= \mathop{\mathbb{E}}_{\theta^{1:T}\sim\mathcal{D}^T}\left[\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\theta\sim\mathcal{D}}\left[r\left(\pi_H^{(t)}(\theta),\ \pi_A^{(t)}(\pi_H^{(t)}(\theta));\ \theta\right)\right]\right]
\end{aligned}$$

Plugging this into (3),

$$\mathop{\mathbb{E}}_{\theta^{1:T}\sim\mathcal{D}^T}\left[\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\theta\sim\mathcal{D}}\left[r\left(\pi_H^{(t)}(\theta),\ \pi_A^{(t)}(\pi_H^{(t)}(\theta));\ \theta\right)\right]\right] \geq \alpha - \frac{\alpha-(1-1/e)}{4}.$$

Let $\bar{X} = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\theta\sim\mathcal{D}}\left[r\left(\pi_H^{(t)}(\theta),\ \pi_A^{(t)}(\pi_H^{(t)}(\theta));\ \theta\right)\right]$. Since rewards lie in $[0,1]$, the random variable $1-\bar{X}$ is non-negative, and the displayed inequality above bounds $\mathbb{E}[1-\bar{X}] \leq 1 - \alpha + \frac{\alpha-(1-1/e)}{4}$. By Markov's inequality applied to $1-\bar{X}$,

$$\Pr\left[\bar{X} < \alpha - \frac{3(\alpha-(1-1/e))}{8}\right] = \Pr\left[1-\bar{X} > 1-\alpha+\frac{3(\alpha-(1-1/e))}{8}\right] \leq \frac{1-\alpha+\frac{\alpha-(1-1/e)}{4}}{1-\alpha+\frac{3(\alpha-(1-1/e))}{8}},$$

and the right-hand side is a constant strictly less than 1 for any fixed $\alpha > 1 - 1/e$. Therefore, with constant probability,

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\theta\sim\mathcal{D}}\left[r\left(\pi_H^{(t)}(\theta),\ \pi_A^{(t)}(\pi_H^{(t)}(\theta));\ \theta\right)\right] \geq \alpha - \frac{3(\alpha-(1-1/e))}{8}.$$

Because it is an average, we can say in this case that there must exist some t such that

$$\mathop{\mathbb{E}}_{\theta\sim\mathcal{D}}\left[r\left(\pi_H^{(t)}(\theta),\ \pi_A^{(t)}(\pi_H^{(t)}(\theta));\ \theta\right)\right] \geq \alpha - \frac{3(\alpha - (1 - 1/e))}{8},$$

and our reduction will find it.

Thus, we can efficiently determine whether we are in the yes case with constant probability, and we are done! □

D. Missing proofs

D.1. Proof of Lemma D.1

Lemma D.1 (Assistance Matroid is a Partition Matroid, restated)**.** *For any assistance game G, the corresponding assistance matroid $\mathcal{M}_G = (\mathcal{U}_G, \mathcal{I}_G)$ defined in Definition 5.2 is a partition matroid with rank M_H (Definition A.3).*

Proof. The maximum size of $S \in \mathcal{I}_G$ is M_H because there can be at most one action pair for each human action. So the rank of $\mathcal{M}_G$ is M_H. To see why this is a partition matroid, consider partitioning the ground set $\mathcal{U}_G$ into disjoint sets $\mathcal{U}_{a_H} = \{a_H\} \times \mathcal{A}_A$ with capacities $d_{a_H} = 1$ for $a_H \in \mathcal{A}_H$. Since no two human actions are in the same independent set, we have $|\mathcal{I}_G \cap \mathcal{U}_{a_H}| = d_{a_H} = 1$. This completes the proof. □

D.2. Proof of Proposition 5.4

For every independent set $I \in \mathcal{I}_G$, let $\mathcal{A}_H^I = \{a_H \in \mathcal{A}_H \mid \exists a_A \in \mathcal{A}_A : (a_H, a_A) \in I\}$ be the set of human actions paired with an assistant action in I. If $I = \emptyset$, then $V_\theta(I) = 0$, and any policy pair achieves reward at least 0 by definition.

Otherwise, we can define the associated assistant policy as playing, in response to any human action $a_H \in \mathcal{A}_H^I$ the assistant action paired with a_H in I, and in response to any human action $a_H \notin \mathcal{A}_H^I$ an arbitrary assistant action $\overline{a_A} \in \mathcal{A}_A$:

$$\pi_A^I(a_H) = \begin{cases} a_A & \text{if } a_H \in \mathcal{A}_H^I \text{ and } (a_H, a_A) \in I, \\ \overline{a_A} & \text{if } a_H \notin \mathcal{A}_H^I. \end{cases}$$

This is well-defined: because the set I is independent, every human action is paired with at most one assistant action.

Defining the associated human policy $\pi_H^I : \Theta \to \mathcal{A}_H$ as being the one that plays the optimal response to π_A^I with actions in $\mathcal{A}_H^I$,

$$\pi_H^I(\theta) = \mathop{\mathrm{argmax}}_{a_H \in \mathcal{A}_H^I} r(a_H, a_A; \theta),$$

the reward the policies π_H^I and π_A^I achieve when paired together on preference θ is exactly the value $V_\theta(I)$:

$$r(\pi_H^I(\theta), \pi_A^I(\pi_H^I(\theta)); \theta) = \max_{a_H \in \mathcal{A}_H^I} r(a_H, \pi_A^I(a_H); \theta) = \max_{(a_H, a_A) \in I} r(a_H, a_A; \theta) = V_\theta(I).$$

This proves the first condition. For the second, given any assistant policy $\pi_A \in \Pi_A$ define the set

$$I_{\pi_A} = \{(a_H, \pi_A(a_H)) \in \mathcal{U}_G : a_H \in \mathcal{A}_H\}.$$

Because every a_H appears in one pair in I_{π_A}, I_{π_A} is an independent set in $\mathcal{M}_G$. For every preference $\theta \in \Theta$ and every human policy $\pi_H : \Theta \to \mathcal{A}_H$, it follows that

$$r(\pi_H(\theta), \pi_A(\pi_H(\theta)); \theta) \leq \max_{a_H \in \mathcal{A}_H} r(a_H, \pi_A(a_H); \theta) = \max_{a_H, a_A \in I_{\pi_A}} r(a_H, a_A; \theta) = V_\theta(I_{\pi_A}).$$

□

D.3. Structural properties of the value of assistance function

Lemma D.2 (restated)**.** *The value of assistance function V_θ is a weighted threshold potential (Definition A.2).*

Proof. Fix a $\theta \in \Theta$. We will show that V_θ is a weighted threshold potential. Sort the values $r(a_H, a_A; \theta)$ from largest to smallest, $\alpha_1, \ldots, \alpha_{M_H M_A}$, where α_i is the ith largest reward a human-assistant action pair can achieve, and let $\left(a_H^{(1)}, a_A^{(1)}\right), \ldots, \left(a_H^{(M_H M_A)}, a_A^{(M_H M_A)}\right)$ be the corresponding ordering of human-assistant action pairs from highest to lowest. Set $\alpha_{M_H M_A + 1} = 0$.

The key trick is to rewrite the difference between the max over any non-empty S and the overall minimum over $\mathcal{A}_H \times \mathcal{A}_A$ as a telescoping sum:

$$\max_{(a_H, a_A) \in S} r(a_H, a_A; \theta) = \sum_{k=1}^{M_H M_A} (\alpha_k - \alpha_{k+1}) \cdot \mathbb{1}\left(\exists i \leq k : \left(a_H^{(i)}, a_A^{(i)}\right) \in S\right).$$

The indicator function $\mathbb{1}\left(\exists i \leq k : \left(a_H^{(i)}, a_A^{(i)}\right) \in S\right)$ is a budget-additive function

$$\mathbb{1}\left(\exists i \leq k : \left(a_H^{(i)}, a_A^{(i)}\right) \in S\right) = \min\left\{1, \sum_{i \in S} w_i^k\right\},$$

where $w_i^k = \mathbb{1}(i \leq k)$. So, defining the modular function $g_k(S) = \sum_{i \in S} w_i^k$, we can write

$$\max_{(a_H, a_A) \in S} r(a_H, a_A; \theta) = \sum_{k=1}^{M_H M_A} (\alpha_k - \alpha_{k+1}) \cdot \min\left\{1,\ g_k(S)\right\}.$$

Because the list of α_i is ordered, it follows that $\alpha_k \geq \alpha_{k+1}$, and so all the coefficients are non-negative. By definition, $\max_{(a_H, a_A) \in S} r(a_H, a_A; \theta) = V_\theta(S)$. So, all in all, when S is non-empty, we can write the value of assistance function as a non-negative linear combination of weighted threshold potentials

$$V_\theta(S) = \sum_{k=1}^{M_H M_A} (\alpha_k - \alpha_{k+1}) \cdot \min\left\{1,\ g_k(S)\right\}. \tag{5}$$

When $S = \emptyset$, both sides are 0, so the equality always holds. □

For the value of assistance function V_θ written in weighted threshold potential form (Equation (5)), the concave relaxation over the matroid polytope $\mathcal{Y} = \operatorname{conv}\{\mathbb{1}_S : S \in \mathcal{I}_G\}$ is

$$\tilde{V}_\theta(\mathbf{y}) = \sum_{k=1}^{M_H M_A} (\alpha_k - \alpha_{k+1}) \cdot \min\left\{1, \sum_{i=1}^{k} y_{(a_H^{(i)}, a_A^{(i)})}\right\},$$

where $\mathbf{y} \in \mathcal{Y} \subseteq [0,1]^{M_H M_A}$.

Lemma D.3 (restated)**.** *For any preference $\theta \in \Theta$, the concave relaxation $\tilde{V}_\theta$ of the value of assistance function is 1-Lipschitz with respect to the ℓ_1 norm over $\mathcal{Y} = [0,1]^{M_H M_A}$.*

Proof. Consider θ and the corresponding concave relaxation $\tilde{V}_\theta$ defined in Definition A.4. We can write $\tilde{V}_\theta(\mathbf{y}) = \sum_{k=1}^{M_H M_A} |c_k| \min\{1, \mathbf{w}_k^\top \mathbf{y}\}$ where $|c_k| = |\alpha_k - \alpha_{k+1}|$ and $\mathbf{w}_k \in \{0,1\}^{M_H M_A}$ are coefficients and weight vectors derived in the proof of Lemma D.2 where we express the value of assistance functions in the form of weighted threshold potentials.

For any $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$:

$$\begin{aligned}
|\tilde{V}_\theta(\mathbf{y}) - \tilde{V}_\theta(\mathbf{y}')| &\leq \sum_{k=1}^{M_H M_A} |c_k| \left|\min\{1, \mathbf{w}_k^\top \mathbf{y}\} - \min\{1, \mathbf{w}_k^\top \mathbf{y}'\}\right| \\
&\leq \sum_{k=1}^{M_H M_A} |c_k| \left|\mathbf{w}_k^\top (\mathbf{y} - \mathbf{y}')\right| \\
&\leq \sum_{k=1}^{M_H M_A} |c_k| \cdot \|\mathbf{w}_k\|_\infty \cdot \|\mathbf{y} - \mathbf{y}'\|_1 \\
&\leq \left(\sum_{k=1}^{M_H M_A} |c_k|\right) \|\mathbf{y} - \mathbf{y}'\|_1.
\end{aligned}$$

This shows that the function $\tilde{V}_\theta$ is Lipschitz with Lipschitz constant:

$$\sum_{k=1}^{M_H M_A} |c_k| = \sum_{k=1}^{M_H M_A} (\alpha_k - \alpha_{k+1}) = \alpha_1 - \alpha_{M_H M_A + 1} = \alpha_1 \leq 1.$$

Where the last inequality holds because rewards are bounded in $[0, 1]$. □

D.4. Proof of Proposition 5.1

By Proposition 5.4, the centralized assistance game reduces to online maximization of value of assistance functions over the assistance matroid $\mathcal{M}_G$. The assistance matroid has ground set size $|\mathcal{U}_G| = M_H M_A$ and rank M_H (Lemma D.1). By Lemma D.2 and Lemma D.3, the value of assistance functions are weighted threshold potentials and their concave relaxations are 1-Lipschitz. Applying the RAOCO algorithm of Salem et al. (2024) with these parameters yields a $\text{poly}(M_H, M_A, N, T)$ time algorithm with expected $(1 - 1/e)$-approximate regret $\mathcal{O}\left(M_H \sqrt{T \log M_A}\right)$. This is a direct result of the regret analysis done in (Salem et al., 2024) which is restated as Proposition E.2 in the appendix. □

D.5. Proof of Lemma 4.4

Let $\left(\overline{\pi}_H^{(t)}, \overline{\pi}_A^{(t)}\right)$ be the policy pair output by Alg_C at round t. By our construction of Alg_H, the human policy played in round t is $\overline{\pi}_H^{(t)}$. Let $\pi_A^{(t)}$ denote the assistant policy played by Alg_A in round t. The assistance regret of the decentralized pair is

$$R_T^\alpha = \alpha \max_{\pi_H, \pi_A} \sum_{t=1}^T r_t(\pi_H, \pi_A) - \sum_{t=1}^T r_t\left(\overline{\pi}_H^{(t)}, \pi_A^{(t)}\right).$$

We decompose this into two terms by adding and subtracting the centralized algorithm's reward:

$$R_T = \underbrace{\alpha \max_{\pi_H, \pi_A} \sum_{t=1}^T r_t(\pi_H, \pi_A) - \sum_{t=1}^T r_t\left(\overline{\pi}_H^{(t)}, \overline{\pi}_A^{(t)}\right)}_{R_{\text{central}}} + \underbrace{\sum_{t=1}^T r_t\left(\overline{\pi}_H^{(t)}, \overline{\pi}_A^{(t)}\right) - \sum_{t=1}^T r_t\left(\overline{\pi}_H^{(t)}, \pi_A^{(t)}\right)}_{R_{\text{coord}}}.$$

The first term R_{central} is the external regret of the centralized algorithm Alg_C, i.e., $R_T^{\alpha,\text{ext}}(\text{Alg}_C)$.

For the second term, let p denote the number of times the sequence $\left(\overline{\pi}_H^{(t)}, \overline{\pi}_A^{(t)}\right)_{t=1}^T$ switches, and let $s_1, \ldots, s_p$ be the time indices at which switches occur (with $s_1 = 1$ and $s_{p+1} = T + 1$). Within each segment $[s_i, s_{i+1})$, the centralized algorithm

plays the same policy pair. We can bound:

$$\begin{aligned}
R_{\text{coord}} &= \sum_{t=1}^{T} r_t\left(\overline{\pi}_H^{(t)}, \overline{\pi}_A^{(t)}\right) - \sum_{t=1}^{T} r_t\left(\overline{\pi}_H^{(t)}, \pi_A^{(t)}\right) \\
&= \sum_{i=1}^{p} \sum_{t=s_i}^{s_{i+1}-1} r_t\left(\overline{\pi}_H^{(s_i)}, \overline{\pi}_A^{(s_i)}\right) - \sum_{i=1}^{p} \sum_{t=s_i}^{s_{i+1}-1} r_t\left(\overline{\pi}_H^{(s_i)}, \pi_A^{(t)}\right) \\
&\le \max_{\substack{s_1,\ldots,s_p \\ \pi_A^{(1)},\ldots,\pi_A^{(p)}}} \sum_{i=1}^{p} \sum_{t=s_i}^{s_{i+1}-1} \left(r_t\left(\overline{\pi}_H^{(s_i)}, \pi_A^{(i)}\right) - r_t\left(\overline{\pi}_H^{(s_i)}, \pi_A^{(t)}\right)\right) \\
&\le R_T^{\text{track}}(\text{Alg}_A;\ p).
\end{aligned}$$

Since $p \le S_T(\text{Alg}_C; \delta)$ with probability at least $1-\delta$, we have

$$R_T^{\alpha} \le R_T^{\alpha,\text{ext}}(\text{Alg}_C) + R_T^{\text{track}}(\text{Alg}_A;\ S_T(\text{Alg}_C; \delta)) + \delta T,$$

where the δT term accounts for the event that the number of switches exceeds the high-probability bound. □

D.6. Proof of Proposition 6.1

From Proposition E.2, the $(1-1/e)$-approximate regret of the CR-RAOCO algorithm is bounded by $(1-1/e)$ times the regret of the underlying OCO algorithm. By Proposition E.3, the number of switches of CR-RAOCO is at most the number of switches of the underlying OCO algorithm, since the coupled randomness makes the rounding step a deterministic function of the OCO iterate.

We instantiate the inner OCO algorithm with the P-COMW algorithm of Agarwal et al. (2024) (Algorithm 4). The convex domain is $\mathcal{Y} = [0,1]^{M_A M_H}$, with diameter $D = \sqrt{M_A M_H}$ and dimension $d = M_A M_H$. By Lemma D.3, the concave relaxation $\tilde{V}_\theta$ is 1-Lipschitz with respect to $\|\cdot\|_1$, hence G-Lipschitz with respect to $\|\cdot\|_2$ for $G \le \sqrt{M_A M_H}$. Setting the switching budget $S = \sqrt{M_A M_H T}$ in Theorem E.4 gives

$$R_T \;\le\; GD\sqrt{2T} \;+\; 16\,GD\log(T)\cdot\frac{\sqrt{dT}}{S} \;+\; 13\,GD \;\in\; O\big(M_A M_H \sqrt{T\log T}\big),$$

where the last bound substitutes $G = D = \sqrt{M_A M_H}$, $d = M_A M_H$, and $S = \sqrt{M_A M_H T}$. Multiplying by the $(1-1/e)$ factor from Proposition E.2 preserves the rate, giving the expected $(1-1/e)$-approximate assistance regret bound stated in Proposition 6.1.

For the switching bound, Theorem E.4 gives $\Pr[S_T \ge 3S] \le e^{-S}$. Choosing $S = \max\big(\sqrt{M_A M_H T},\ \log(1/\delta)\big)$ yields, with probability at least $1-\delta$, $S_T \le 3S \in O\big(\sqrt{M_A M_H T} + \log(1/\delta)\big) \subseteq O\big(\sqrt{M_A M_H T \log(1/\delta)}\big)$, matching the bound stated in Proposition 6.1. By Proposition E.3 this same bound applies to the number of switches made by CR-RAOCO.

Crucially, Theorem E.4 requires only convexity and Lipschitz-ness of the losses—both satisfied by $\tilde{V}_\theta$—and does not require the smoothness assumption needed by FTPRLL (Sherman & Koren, 2021), which fails for the piecewise-linear value-of-assistance relaxation. □

D.7. Proof of Proposition 6.2

We construct Alg_A using an algorithm $\text{Alg}_{\text{track}}$ as a subroutine in a blackbox manner. We will choose $\text{Alg}_{\text{track}}$ to be an algorithm designed to minimize tracking regret in general online learning problems (such as algorithms from (Herbster & Warmuth, 1998; Cesa-Bianchi et al., 2012)).

We will first describe how Alg_A is constructed using $\text{Alg}_{\text{track}}$. Then we will express the tracking regret as of Alg_A in terms of tracking regret of $\text{Alg}_{\text{track}}$. Finally we will apply bounds on tracking regret from previous work to obtain the bound of the proposition.

The assistant's algorithm Alg_A maintains M_H copies of $\text{Alg}_{\text{track}}$, one for each human action $a_H \in \mathcal{A}_H$. Let us denote the copy associated with action a_H by $\text{Alg}_{\text{track}}^{a_H}$. Each copy of $\text{Alg}_{\text{track}}$ optimizes over the space of assistant actions $\mathcal{A}_A$. The

assistant's policy at round t: $\pi_A^{(t)}$ chooses the action selected by the copy $\text{Alg}_{\text{track}}^{a_H^{(t)}}$, where $a_H^{(t)} = \pi_H^{(t)}(\theta^{(t)})$. At the end of round t, the assistant updates this copy of $\text{Alg}_{\text{track}}$ using the bandit feedback it receives of reward $r_t(\pi_H^{(t)}, \pi_A^{(t)})$ upon selecting action $a_A^{(t)}$.

We will show that we can bound the assistant's tracking regret over the space Π_A by the sum of the tracking regrets of each of the M_H copies of $\text{Alg}_{\text{track}}$. We can decompose the reward r_t as $r_t(\overline{\pi}_H^{(t)}, \pi_A) = \sum_{a_H \in \mathcal{A}_H} r_t^{a_H}(\pi_A(a_H))$, where each $r_t^{a_H} : \mathcal{A}_A \to [0,1]$ is defined by $r_t^{a_H}(a_A) = r(a_H, a_A; \theta^{(t)}) \cdot \mathbf{1}\left(\overline{\pi}_H^{(t)}(\theta^{(t)}) = a_H\right)$. $r_t^{a_H}$ is the reward function with which the copy $\text{Alg}_{\text{track}}^{a_H}$ is updated. We can decompose the tracking regret of the assistant as

$$\begin{aligned}
R_T^{\text{track}}(\text{Alg}_A; p) &= \max_{\substack{s_1,\dots,s_{p+1} \\ \pi_A^{(1)},\dots,\pi_A^{(p)} \in \Pi_A}} \sum_{i=1}^{p} \sum_{t=s_i}^{s_{i+1}-1} r_t(\overline{\pi}_H^{(t)}, \pi_A^{(i)}) - \sum_{t=1}^{T} r_t(\overline{\pi}_H^{(t)}, \pi_A^{(t)}) \\
&= \max_{\substack{s_1,\dots,s_{p+1} \\ \pi_A^{(1)},\dots,\pi_A^{(p)}}} \sum_{a_H \in \mathcal{A}_H} \left[\sum_{i=1}^{p} \sum_{t=s_i}^{s_{i+1}-1} r_t^{a_H}(\pi_A^{(i)}(a_H)) - \sum_{t=1}^{T} r_t^{a_H}(\pi_A^{(t)}(a_H)) \right] \\
&\le \sum_{a_H \in \mathcal{A}_H} \max_{\substack{s_1,\dots,s_{p+1} \\ \pi_A^{(1)},\dots,\pi_A^{(p)}}} \left[\sum_{i=1}^{p} \sum_{t=s_i}^{s_{i+1}-1} r_t^{a_H}(\pi_A^{(i)}(a_H)) - \sum_{t=1}^{T} r_t^{a_H}(\pi_A^{(t)}(a_H)) \right] \\
&\le \sum_{a_H \in \mathcal{A}_H} R_T^{\text{track}}(\text{Alg}_{\text{track}}^{a_H}; p) = M_H R_T^{\text{track}}(\text{Alg}_{\text{track}}^{a_H}; p).
\end{aligned}$$

Using regret bounds for a tracking regret minimizing algorithm from previous work like Fixed-Share algorithm (Herbster & Warmuth, 1998; Cesa-Bianchi et al., 2012) that we restate in Section E.3, we get the tracking regret bound in the proposition. □

D.8. Proof of Theorem 4.1

First we describe the decentralized human and assistant yielding this assistance regret rate. The human algorithm is derived from the centralized algorithm Alg_C with the external regret and switching bound provided in Proposition 6.1. In particular, Alg_C has expected $(1 - 1/e)$-external regret at most $\mathcal{O}\left(\sqrt{M_A M_H T}\right)$ and, with probability $1 - 1/T$, switches at most $\mathcal{O}\left(M_A M_H T \log(T)\right)$ times. Alg_H is derived from Alg_C in the standard way described in Algorithm 1. The assistant algorithm Alg_A is the one shown to achieve low tracking regret in Proposition 6.2 and the human algorithm Alg_H derived from the centralized algorithm.

We can immediately apply Lemma 4.4 to achieve the regret guarantee stated in the theorem, for $\alpha = 1 - 1/e$,

$$\begin{aligned}
R_T^{\alpha} &\le R_T^{\alpha,\text{ext}}(\text{Alg}_C) + R_T^{\text{track}}(\text{Alg}_A; S_T(\text{Alg}_C; \delta)) + \delta T && \text{(By Lemma 4.4)} \\
&= R_T^{\alpha,\text{ext}}(\text{Alg}_C) + M_H \sqrt{M_A T \log(M_A T) \cdot S_T(\text{Alg}_C; 1/T)} + 1 && \text{(By Proposition 6.2)} \\
&= \sqrt{M_H M_A T} + M_H \sqrt{M_A T \log(M_A T) \cdot \sqrt{M_A M_H T \log(T)}} + 1 && \text{(By Proposition 6.1)} \\
&\in \mathcal{O}\left(M_H^{5/4} M_A^{3/4} \log(M_A T)^{3/4} T^{3/4}\right).
\end{aligned}$$

□

D.9. Proof of Theorem 4.2

We first describe the human and assistant algorithms Alg_H and Alg_A, and then bound their assistance regret.

Construction of Alg_H and Alg_A. As in the proofs of Lemma 4.4 and Proposition 6.2, the human algorithm Alg_H runs the stable centralized algorithm Alg_C from Proposition 6.1. On most rounds it plays the human component of Alg_C's output. However, whenever Alg_C switches to a new policy pair, Alg_H temporarily deviates in order to communicate the new assistant policy to the assistant. This removes the need for the assistant to relearn the policy through exploration, enabling us to get a better regret rate than the $O(T^{3/4})$ rate in Proposition 6.2.

An assistant policy is a mapping $\pi_A : \mathcal{A}_H \to \mathcal{A}_A$. There are $M_A^{M_H}$ such policies, so any policy can be encoded using $\log_{M_H}(M_A^{M_H}) = M_H \log_{M_H}(M_A)$ human actions. We assume the agents share a common mapping sequences of human actions to assistant policies.

Signaling a switch. To begin communication, both agents must detect that a switch has occurred. The human knows this (since it runs Alg_C), but the assistant does not. We design the following signaling-string protocol in which the human uses a special sequence of actions to indicate that a switch has occurred.

Before the first round of the game, but after nature chooses a sequence of preferences $(\theta^{(1)}, \dots, \theta^{(T)})$, the human and assistant sample a shared random string $\sigma \in \mathcal{A}_H^\ell$, for $\ell = \lceil 2 \log_{M_H} T \rceil$, where each element of σ that is drawn uniformly at random from $\mathcal{A}_H$. The human chooses actions according to string σ to indicate to the assistant the start of a communication phase.

Let us call the rounds outide of the communication and signaling string σ rounds of normal play. These are rounds where the human's action is derived from the centralized algorithm. That is, $a_H^{(t)} = \overline{\pi}_H^{(t)}(\theta^{(t)})$.

Whenever Alg_C switches to a new policy pair, the human instead plays 1) the signaling string σ for ℓ rounds, and 2) the encoded assistant policy for $M_H \log_{M_H}(M_A)$ rounds. The assistant monitors a sliding window of the last ℓ human actions and, upon detecting σ, interprets the subsequent $M_H \log_{M_H}(M_A)$ actions as an encoding of the new assistant policy.

No false positives in assistant's detection of communication phase. We now show it is unlikely that the assistant incorrectly perceives the start of a communication phase. That is σ is unlikely to appear accidentally during normal play.

Because the adversary is oblivious, the sequence $(\theta^{(1)}, \dots, \theta^{(T)})$ is fixed before σ is sampled. Moreover, Alg_C is full-information: its updates depend only on past preferences and its own randomness, and not on the realized actions. Fix any realization ρ of Alg_C's internal randomness. Given $(\theta^{(1)}, \dots, \theta^{(T)})$ and ρ, the sequence of policies output by Alg_C is deterministic. Hence the sequence of human actions during normal play $\mathbf{a_H}^{\mathrm{norm}}(\rho)$ is deterministic and does not depend on σ.

Since σ is drawn uniformly from $\mathcal{A}_H^\ell$ and independently of $\mathbf{a}^{\mathrm{norm}}(\rho)$, the probability that any fixed window of ℓ consecutive normal-play actions equals σ is $M_H^{-\ell}$. Taking a union bound over at most T such windows,

$$\Pr\left[\sigma \text{ appears during normal play}\right] \le T \cdot M_H^{-\ell} \le T \cdot T^{-2} = \frac{1}{T}.$$

This bound holds for every ρ, and hence unconditionally. Therefore, with probability at least $1 - 1/T$, the assistant detects every communication phase correctly and incurs no false positives.

After receiving a communicated policy, the assistant plays it exactly until the next switch, achieving perfect synchronization.

Assistance regret analysis. To get the final bound on assistance regret, we will use an approach similar to the decomposition from Lemma 4.4. Conditioning on the event that σ does not appear during normal play (which holds with probability at least $1 - 1/T$), the policies played by the decentralized algrorithms match the policies selected by Alg_C in all rounds but 1) the rounds where signaling string σ is played to indicate switch, and 2) the rounds used to communicate the switched assistant policy. For a single switch, these rounds occur $\ell = \log_{M_H} T$ times and $M_H \log_{M_H} M_A$ times respectively. Since rewards lie in $[0, 1]$, each such round incurs at most unit regret. If Alg_C makes S_T switches, then these rounds result in a cost of coordination of

$$R_{\mathrm{coord}} \le \left(\ell + M_H \log_{M_H}(M_A)\right) S_T = O\big(M_H \log_{M_H}(M_A) + \log_{M_H} T\big)\, S_T.$$

On the $1/T$-probability event of a false positive, we bound regret by T, contributing at most 1 in expectation.

By Proposition 6.1, the centralized algorithm satisfies

$$R_T^{\mathrm{ext}}(\mathrm{Alg}_C) \in O(\sqrt{M_A M_H T}), \qquad S_T \in O\big(\sqrt{M_A M_H T \log(1/\delta)}\big)$$

with probability at least $1-\delta$. Setting $\delta = 1/T$ and combining terms,

$$\begin{aligned} R_T &\leq R_T^{\text{ext}}(\text{Alg}_C) + R_{\text{coord}} + \delta T \\ &\leq O(\sqrt{M_A M_H T}) + O\big(M_H \log_{M_H}(M_A) + \log_{M_H} T\big) \cdot O\big(\sqrt{M_A M_H T \log T}\big) + O(1) \\ &= O\big(M_H \log_{M_H}(M_A) \cdot \sqrt{M_A M_H T}\big). \end{aligned}$$

E. Results from Previous Work

In this section, we will state the results from previous work that we use for our results. We will also prove any extensions to these results that we need for our results.

E.1. Online Submodular Maximization

A key tool that we use is the reduction of online submodular maximization to online convex optimization constructed in Salem et al. (2024). This reduction relies on the existence of concave relaxations and rounding schemes that satisfy a "sandwich property." The following proposition establishes that this property holds for Lipschitz weighted threshold potentials (Definition A.2), which is the class of functions arising in assistance games.

Proposition E.1 (Sandwich Property for Lipschitz Weighted Threshold Potentials)**.** *Let $\mathcal{X} \subseteq \{0,1\}^n$ and let $\mathcal{Y} = \text{conv}(\mathcal{X})$ be its convex hull. Let $\mathcal{F}$ be a class of weighted threshold potential functions (Definition A.2) over $\mathcal{X}$ that are L-Lipschitz. Then, there exists a randomized rounding $\Xi : \mathcal{Y} \times U \to \mathcal{X}$, where U is a random variable, such that for every $f \in \mathcal{F}$, there exists an L-Lipschitz concave function $\tilde{f} : \mathcal{Y} \to \mathbb{R}$ satisfying:*

1. *$\tilde{f}(\mathbf{x}) \geq f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$, and*
2. *$\mathbb{E}_\Xi[f(\Xi(\mathbf{y}))] \geq (1-1/e) \cdot \tilde{f}(\mathbf{y})$ for all $\mathbf{y} \in \mathcal{Y}$.*

This proposition is stated as Lemma 1 in Salem et al. (2024) and its proof is provided in Appendix E of Salem et al. (2024). The proof involves showing that negatively correlated randomized roundings lead to the property stated in the proposition for weighted threshold potential functions with concave relaxations that are Lipschitz. It then uses results from Chekuri et al. (2010) to show that swap rounding and randomized pipage rounding are negatively correlated for partition matroids $\mathcal{M}$. We describe the swap rounding algorithm and express it as a function over $\mathcal{X}$ and a random variable $U \sim \text{Uniform}([0,1])^{n+1}$.

Algorithm 2 Swap Rounding (Chekuri et al., 2010; Salem et al., 2024)

Require: $\mathbf{y} \in \mathcal{Y} = \text{conv}(\mathcal{X})$, random vector $\mathbf{U} = (U_1, \ldots, U_{n+1}) \sim \text{Unif}([0,1])^{n+1}$
Ensure: $\mathbf{x} \in \mathcal{X}$

```
Decompose y = Σ_{k=1}^{K} γ_k z_k where γ_k ∈ [0,1], Σ_{k=1}^{K} γ_k = 1, and z_k ∈ X are bases of the matroid  ▷ K ≤ n+1 by Carathéodory
β_1 ← γ_1
x ← z_1
for k = 1, ..., K − 1 do
    β_{k+1} ← β_k + γ_{k+1}
    if U_k ≤ β_k/β_{k+1} then
        x ← x                                   ▷ Probability β_k/β_{k+1}: keep current
    else
        x ← z_{k+1}                             ▷ Probability γ_{k+1}/β_{k+1}: switch
    end if
end for
return x
```

The algorithm provided by (Salem et al., 2024) using this reduction is called Randomness Augmented Online Convex Optimization (RAOCO). We use a slightly modified version called the Coupled-Randomness RAOCO (CR-RAOCO) algorithm. CR-RAOCO implements the RAOCO algorithm in Salem et al. (2024) but by coupling the randomness in the

rounding step across the different rounds. This is done by drawing a single random variable $\mathbf{U} \sim \mathrm{Uniform}([0,1])^{n+1}$ and using it to round all the points across all the rounds. We introduce the coupling to enhance the stability of the RAOCO algorithm. We provide a description of the CR-RAOCO algorithm in the following algorithm Algorithm 3.

Algorithm 3 Coupled-Randomness RAOCO (CR-RAOCO)

Require: OCO policy $\mathcal{P}_{\mathcal{Y}}$, randomized rounding $\Xi : \mathcal{Y} \times U \to \mathcal{X}$, randomness distribution $\mathcal{D}_{random}$

$\mathbf{U} \sim \mathcal{D}_{random}$ ▷ Sample randomness once
for $t = 1, 2, \ldots, T$ **do**
  $\mathbf{y}_t \leftarrow \mathcal{P}_{\mathcal{Y},t}((\mathbf{y}_s)_{s<t}, (\tilde{f}_s)_{s<t})$
  $\mathbf{x}_t \leftarrow \Xi(\mathbf{y}_t, \mathbf{U})$ ▷ Round using shared randomness $\mathbf{U}$
  Play $\mathbf{x}_t$ and receive reward $f_t(\mathbf{x}_t)$
  Reward function f_t is revealed
  Construct concave extension $\tilde{f}_t$ from f_t as in Proposition E.1
end for

The following proposition shows that the regret of the CR-RAOCO policy transfers from the OCO regret guarantee. The analysis is basically the same as the regret analysis of RAOCO done by (Salem et al., 2024). The only difference is establishing that the sandwich property shown in Proposition E.1 holds with the coupled rounding procedure as well. The sandwich property is only a condition on the expectation of a function applied to the rounded random variable. Since the marginal distribution of the rounded random variable is unchanged after the coupling, the expectation remains the same and the property continues to hold. We state the regret analysis including this additional argument below.

Proposition E.2 (Regret Bound for RAOCO (Salem et al., 2024))**.** *Under the Sandwich Property (Proposition E.1), given an OCO policy $\mathcal{P}_{\mathcal{Y}}$ operating over $\mathcal{Y} = \mathrm{conv}(\mathcal{X})$, the RAOCO policy $\mathcal{P}_{\mathcal{X}}$ described by Algorithm 3 satisfies*

$$\alpha\text{-regret}_T(\mathcal{P}_{\mathcal{X}}) \leq \alpha \cdot \mathrm{regret}_T(\mathcal{P}_{\mathcal{Y}}).$$

Proof. Consider the sequence of reward functions $\{f_1, f_2, \ldots, f_T\} \in \mathcal{F}^T$ and the associated sequence of concave relaxations $\{\tilde{f}_1, \tilde{f}_2, \ldots, \tilde{f}_T\} \in \tilde{\mathcal{F}}^T$. Then,

$$\max_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}) \leq \max_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} \tilde{f}_t(\mathbf{x}) \leq \max_{\mathbf{y}\in\mathcal{Y}} \sum_{t=1}^{T} \tilde{f}_t(\mathbf{y}). \tag{6}$$

The first inequality follows from the upper bound property $\tilde{f}(\mathbf{x}) \geq f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$ (property 1 in Proposition E.1), and the second inequality holds because maximizing over a superset $\mathcal{Y} = \mathrm{conv}(\mathcal{X}) \supseteq \mathcal{X}$ can only increase the objective value attained.

The total expected reward obtained by the RAOCO policy is given by

$$\mathbb{E}_{\Xi}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] = \sum_{t=1}^{T} \mathbb{E}_{\Xi}[f_t(\mathbf{x}_t)] = \sum_{t=1}^{T} \mathbb{E}_{\Xi}[f_t(\Xi(\mathbf{y}_t))] \geq \alpha \sum_{t=1}^{T} \tilde{f}_t(\mathbf{y}_t), \tag{7}$$

where the inequality follows from the rounding property $\mathbb{E}_{\Xi}[f(\Xi(\mathbf{y}))] \geq \alpha \cdot \tilde{f}(\mathbf{y})$ for all $\mathbf{y} \in \mathcal{Y}$ (property 2 in Proposition E.1).

Combining Equations (6) and (7), we obtain

$$\alpha \max_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}) - \mathbb{E}_{\Xi}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] \leq \alpha \max_{\mathbf{y}\in\mathcal{Y}} \sum_{t=1}^{T} \tilde{f}_t(\mathbf{y}) - \alpha \sum_{t=1}^{T} \tilde{f}_t(\mathbf{y}_t).$$

Taking the supremum over all sequences of reward functions in $\mathcal{F}^T$ on both sides yields the desired result:

$$\alpha\text{-regret}_T(\mathcal{P}_{\mathcal{X}}) \leq \alpha \cdot \mathrm{regret}_T(\mathcal{P}_{\mathcal{Y}}).$$

□

The following proposition establishes that the number of switches made by CR-RAOCO is exactly the number of switches made by the underlying OCO policy. This is a key property that allows us to control the switching cost of the algorithm.

Proposition E.3 (Switching Bound for CR-RAOCO)**.** *Let $\mathcal{P}_{\mathcal{Y}}$ be an OCO policy and let $\mathcal{P}_{\mathcal{X}}$ be the CR-RAOCO policy described by Algorithm 3. Let $(\mathbf{y}_1, \dots, \mathbf{y}_T)$ be the sequence of fractional points produced by $\mathcal{P}_{\mathcal{Y}}$ and let $(\mathbf{x}_1, \dots, \mathbf{x}_T)$ be the sequence of integral points produced by $\mathcal{P}_{\mathcal{X}}$. Then, the number of switches satisfies*

$$\sum_{t=2}^{T} \mathbb{1}[\mathbf{x}_t \neq \mathbf{x}_{t-1}] \leq \sum_{t=2}^{T} \mathbb{1}[\mathbf{y}_t \neq \mathbf{y}_{t-1}].$$

Proof. In CR-RAOCO, the randomness $\mathbf{U}$ is sampled once at the beginning and shared across all rounds. Given this fixed $\mathbf{U}$, the rounding function $\Xi(\cdot, \mathbf{U}) : \mathcal{Y} \to \mathcal{X}$ is a deterministic function. Therefore, $\mathbf{x}_t = \Xi(\mathbf{y}_t, \mathbf{U})$ for all t.

Since $\Xi(\cdot, \mathbf{U})$ is deterministic, if $\mathbf{y}_t = \mathbf{y}_{t-1}$, then $\mathbf{x}_t = \Xi(\mathbf{y}_t, \mathbf{U}) = \Xi(\mathbf{y}_{t-1}, \mathbf{U}) = \mathbf{x}_{t-1}$. Equivalently, $\mathbf{x}_t \neq \mathbf{x}_{t-1}$ implies $\mathbf{y}_t \neq \mathbf{y}_{t-1}$. Thus,

$$\mathbb{1}[\mathbf{x}_t \neq \mathbf{x}_{t-1}] \leq \mathbb{1}[\mathbf{y}_t \neq \mathbf{y}_{t-1}]$$

for all $t \geq 2$, and summing over t yields the result. □

E.2. Online Convex Optimization with Few Switches

The CR-RAOCO algorithm uses an OCO algorithm as a black box. To ensure CR-RAOCO has both low regret and few switches, we use the Private Continuous Online Multiplicative Weights (P-COMW) algorithm of Agarwal et al. (2024), a Shrinking-Dartboard-style algorithm based on sampling from a strongly log-concave density. Unlike the lazy OCO algorithm of Sherman & Koren (2021), which requires the loss functions to be β-smooth, P-COMW requires only that the losses be convex and Lipschitz—properties that hold for the value-of-assistance concave relaxation $\tilde{V}_\theta$ used in our setting (Lemma D.3). Compared to the earlier Shrinking-Dartboard variant of Anava et al. (2015), P-COMW additionally provides a flexible switching budget, an improved dimensional dependence in its regret bound, and a high-probability bound on the number of switches that is included directly in the analysis of Agarwal et al. (2024).

Algorithm 4 Private Continuous Online Multiplicative Weights (P-COMW) (Agarwal et al., 2024)

Require: Convex domain $W \subseteq \mathbb{R}^d$, temperature $\beta > 0$, regularization $\lambda > 0$, scale $\Phi > 0$
Sample $x_1 \sim \mu_1$ where $\mu_1(x) \propto \exp\!\big(-\frac{\lambda}{2}\|x\|^2\big)$
for $t = 1, \dots, T$ **do**
  Play x_t and observe loss $g_t : W \to [0, 1]$
  Define the strongly log-concave density $\mu_{t+1}(x) \propto \exp\!\Big(-\beta \sum_{\tau=1}^{t} g_\tau(x) - \frac{\lambda}{2}\|x\|^2\Big)$
  With probability $\min\big\{1,\ \mu_{t+1}(x_t)/(\Phi \cdot \mu_t(x_t))\big\}$, set $x_{t+1} \leftarrow x_t$ ▷ Stay
  Otherwise, sample $x_{t+1} \sim \mu_{t+1}$ independently ▷ Switch
end for

P-COMW samples each iterate from a strongly log-concave density—continuous online multiplicative weights with an added ℓ_2 regularization—and uses a rejection-sampling-style stay-step that ensures few switches while preserving the marginal regret of the underlying continuous multiplicative weights distribution. The strong-convexity term governed by λ is the key innovation that buys the improved dimensional dependence relative to prior non-smooth analyses. The following theorem from Agarwal et al. (2024) bounds both the regret and the number of switches as a function of an arbitrary switching budget S, with no smoothness assumption on the losses.

Theorem E.4 (Lazy OCO Guarantees (Agarwal et al., 2024))**.** *Let $W \subseteq \mathbb{R}^d$ be convex with diameter $D = \sup_{x,y \in W} \|x - y\|$, and let $g_1, \dots, g_T : W \to [0, 1]$ be convex and G-Lipschitz. For any switching budget $S \in [1, T]$, with parameters β, λ, Φ chosen as in Theorem 4.5 of Agarwal et al. (2024), Algorithm 4 satisfies*

$$R_T \ \leq\ GD\sqrt{2T} \ +\ 16\, GD \log(T) \cdot \frac{\sqrt{dT}}{S} \ +\ 13\, GD,$$

and the number of switches satisfies $\mathbb{E}[S_T] \leq S$ together with the high-probability bound

$$\Pr\big[S_T \geq 3S\big] \ \leq\ e^{-S}.$$

Polynomial-time implementation. Algorithm 4 requires sampling from the strongly log-concave density $\mu_{t+1}(x) \propto \exp\left(-\beta \sum_{\tau \le t} g_\tau(x) - \frac{\lambda}{2}\|x\|^2\right)$ over W. When W admits a polynomial-time membership oracle—as is the case for the matroid polytope used in CR-RAOCO—this sampling problem can be solved approximately in time $\mathrm{poly}(d, T, 1/\varepsilon)$ via standard log-concave samplers such as hit-and-run (Lovász & Vempala, 2007). The polynomial degree is higher than that of FTPRLL but remains $\mathrm{poly}(M_H, M_A, N, T)$, matching the runtime claim of our centralized algorithm.

E.3. Tracking regret

The tracking regret framework has been studied to achieve good rewards relative to the baseline action changing p times during the T rounds and is related to adaptive regret (Hazan & Seshadhri, 2007).

Herbster & Warmuth (1998) provide an algorithm Fixed-Share that achieves m-segment tracking regret $\mathcal{O}\left(\sqrt{Tm(\log N + \log T)N}\right)$ in the full information setting. Cesa-Bianchi et al. (2012) show that the fixed share algorithm is equivalent to online mirror descent with a particular projection. Theorem 4.1 of Shalev-Shwartz et al. (2012) shows how online mirror descent with bandit information can be implemented without degradation of regret. This is stated in the following Theorem by Daniely et al. (2015).

Theorem E.5 (Restatement of theorem by Daniely et al. (2015))**.** *Fixed-Share algorithm in the bandit feedback setting achieves expected m-segment tracking regret at most $\mathcal{O}\left(\sqrt{Tm(\log N + \log T)N}\right)$.*

E.4. Minmax lower bound

The tools used to show a minmax lower bound of $\Omega(\sqrt{T})$ regret in the standard online learning regret minimization setting can be used to derive a similar minmax lower bound of $\Omega(\sqrt{T})$ for assistance regret.

That is, we can show that for any pair of human and assistant algorithms, there is an instance of the assistance game that results in assistance regret at least $\Omega(\sqrt{T})$. The lower bound is constructed through showing limitations on distinguishing between two Bernoulli distributions with means $1/2 + \varepsilon$ and $1/2 - \varepsilon$.

At a high level, we can view the general online learning problem as a special case of the assistance game with just a single action that the human can take. The assistant's problem in this case resembles the standard external regret minimization problem.

Proposition E.6 (Assistance regret lower bound)**.** *For every human-assistant learning algorithms, there is an instance of the online assistance game for which assistant regret is $\Omega(\sqrt{T})$.*

Proof. We will show a reduction from the problem of distinguishing between two Bernoulli distributions to an assistance game.

The ε-Bernoulli distinction problem is the problem of given a distribution that is one of $\mathrm{Bern}(1/2 - \varepsilon)$ or $\mathrm{Bern}(1/2 + \varepsilon)$, determining which distribution it is.

Standard lower bounds for this problem state that the probability of success of any algorithm with T samples in the ε-Bernoulli distinction problem, is at most $1 - \exp(-\varepsilon^2 T/2)$.

Now we will show that these lower bounds imply a lower bound for assistance regret by establishing an algorithmic reduction

Algorithm for distinction from algorithm for the assistance game. Consider any human-assistant algorithms for the assistance game. We can construct an algorithm for the ε-Bernoulli distinction problem in the following way.

We can set up a assistance game with a single action for the human, where the preference space is $\Theta = \{0, 1\}$. The assistant's action set is $\mathcal{A} = \{0, 1\}$, and the agents receive a reward of 1 if the assistant correctly predicts the human's preference, and a reward of 0 otherwise. Preferences are drawn at the start of the game by nature according to the fixed distribution D which can be either $\mathrm{Bern}(1/2 - \varepsilon)$ or $\mathrm{Bern}(1/2 + \varepsilon)$.

If the assistant predicts the preference to be 1 more than $1/2$ of the time, we conclude that the distribution is $\mathrm{Bern}(1/2 + \varepsilon)$ and otherwise, we conclude the distribution is $\mathrm{Bern}(1/2 - \varepsilon)$.

Let us analyze the probability of success of this approach and relate it to the assistance regret.

Let the assistant games induced by state generating distributions $\mathrm{Bern}(1/2 - \varepsilon)$, $\mathrm{Bern}(1/2 + \varepsilon)$ be G_ε, G'_ε respectively. For

a human-assistant learning algorithm, let $C_1(T)$ be a random variable denoting the number of times the assistant predicts the state to be 1 in T rounds. For any assistant learning algorithm π, we can write the assistance regret under the games $G_\varepsilon, G'_\varepsilon$ as $R_T(\pi, G_\varepsilon)$ and $R_T(\pi, G'_\varepsilon)$ respectively.

We can bound both regrets in terms of the number of times the assistant predicts the preference 1 in the following way. In game G_ε (state distribution $\mathrm{Bern}(1/2-\varepsilon)$), the optimal fixed assistant policy is to always predict 0, giving expected reward $T(1/2+\varepsilon)$, while the algorithm's expected reward is $T(1/2+\varepsilon) - 2\varepsilon\,\mathbb{E}[C_1(T)]$ (since $a_A^{(t)} \perp \theta^{(t)}$ given the history). Markov's inequality on the non-negative variable $C_1(T)$ then gives

$$R_T(\pi, G_\varepsilon) \;\geq\; 2\varepsilon\,\mathbb{E}[C_1(T)] \;\geq\; T\varepsilon \cdot \mathbb{P}_{D_\varepsilon}(C_1(T) > T/2),$$

and symmetrically (with optimal policy "predict 1 always" and Markov on $T - C_1(T)$),

$$R_T(\pi, G'_\varepsilon) \;\geq\; T\varepsilon \cdot \mathbb{P}_{D'_\varepsilon}(C_1(T) \leq T/2).$$

The events $\{C_1(T) > T/2\}$ under D_ε and $\{C_1(T) \leq T/2\}$ under D'_ε are exactly the events that our test guesses incorrectly, so their sum is the total failure probability of the distinction test. Adding the two regret bounds,

$$\begin{aligned} R_T(\pi, G_\varepsilon) + R_T(\pi, G'_\varepsilon) &\geq T\varepsilon \cdot \left[\mathbb{P}_{D_\varepsilon}(C_1(T) > T/2) + \mathbb{P}_{D'_\varepsilon}(C_1(T) \leq T/2)\right] \\ &\geq T\varepsilon \cdot \text{Probability of failure in distinction problem} \\ &\geq T\varepsilon \cdot \exp(-\varepsilon^2 T/2), \end{aligned}$$

where the last inequality uses the standard distinction lower bound stated above (success $\leq 1 - \exp(-\varepsilon^2 T/2)$, hence failure $\geq \exp(-\varepsilon^2 T/2)$).

Choosing $\varepsilon = \min(1/2, \sqrt{1/(4T)}) = 1/(2\sqrt{T})$ for $T \geq 1$ gives $\exp(-\varepsilon^2 T/2) = \exp(-1/8)$, so the sum of regrets is $\Omega(\sqrt{T})$, and at least one of $R_T(\pi, G_\varepsilon)$, $R_T(\pi, G'_\varepsilon)$ is $\Omega(\sqrt{T})$, which is the lower bound of the proposition.

□

F. Additional Related Work

A particular subclass of assistance games are *communication games*, which are studied in the line of work on Emergent Communication (EC) (Foerster et al., 2016; Lazaridou et al., 2016; Lazaridou & Baroni, 2020). This literature studies how agents learn to communicate in Lewis signaling games when trained using standard training dynamics. Empirical work has investigated how the choice of training parameters impacts the efficiency of communication (e.g.,(Havrylov & Titov, 2017; Kim & Oh, 2021; Ren et al., 2019; Chaabouni et al., 2022; Rita et al., 2020; Li & Bowling, 2019)) Convergence of training dynamics have received a more theoretical treatment in works in game theory and evolutionary biology showing (Franke, 2009b;a; Jäger, 2007; 2012; Trapa & Nowak, 2000; Kirby, 2002; Kirby et al., 2014; Jacob et al., 2023), but the rates of convergence are usually not analyzed.

Training dynamics have received a more theoretical treatment in works in game theory and evolutionary biology (Franke, 2009b;a; Jäger, 2007; 2012; Trapa & Nowak, 2000; Kirby, 2002; Kirby et al., 2014; Jacob et al., 2023). However, these works mostly view language formation as equilibrium computation. There are many possible equilibria and there is no guarantee of convergence to the optimal equilibrium, which is the goal of our work.