The Assistive Multi-Armed Bandit

Lawrence Chan
University of California, Berkeley
chanlaw@berkeley.edu

Dylan Hadfield-Menell
University of California, Berkeley
dhm@berkeley.edu

Siddhartha Srinivasa
University of Washington
siddh@cs.washington.edu

Anca Dragan
University of California, Berkeley
anca@berkeley.edu

***Abstract*—Learning preferences implicit in the choices humans make is a well studied problem in both economics and computer science. However, most work makes the assumption that humans are acting (noisily) optimally with respect to their preferences. Such approaches can fail when people are themselves learning about what they want. In this work, we introduce the assistive multi-armed bandit, where a robot assists a human playing a bandit task to maximize cumulative reward. In this problem, the human does not know the reward function but can learn it through the rewards received from arm pulls; the robot only observes which arms the human pulls but not the reward associated with each pull. We offer sufficient and necessary conditions for successfully assisting the human in this framework. Surprisingly, better human performance in isolation does not necessarily lead to better performance when assisted by the robot: a human policy can do better by effectively communicating its observed rewards to the robot. We conduct proof-of-concept experiments that support these results. We see this work as contributing towards a theory behind algorithms for human-robot interaction.**

***Index Terms*—preference learning, assistive agents**

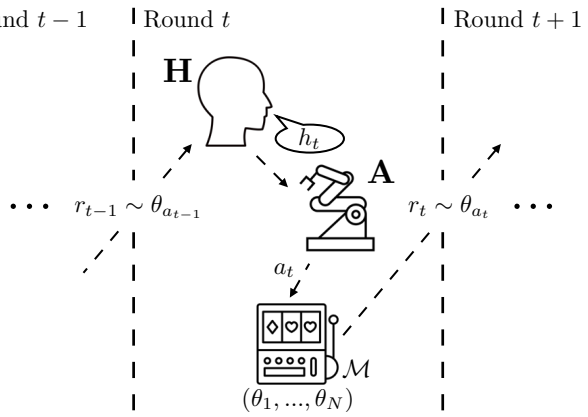

Fig. 1. We introduce the assistive multi-armed bandit: a formalism for the problem of helping a learning agent optimize their reward. In each round, the human observes reward and tells the robot which arms they would like it to pull. The robot observes these requests, attempts to infer the reward values, and selects an arm to pull.

I. Introduction

Preference learning [1] seeks to learn a predictive model of human preferences from their observed behavior. These models have been applied quite successfully in contexts like personalized news feeds [2]–[4], movie recommendations [5]–[7], and human robot interaction [8]–[12]. We can learn this predictive model by fitting a utility function to revealed preferences [13]–[15], fitting parameters in a pre-specified human model [16], and applying contextual-bandit algorithms [3].

Central to all of these approaches is a fundamental assumption: human behavior is *noisily-optimal* with respect to a set of *stationary* preferences. Under this assumption, the problem can then be elegantly cast and analyzed as an *inverse optimal control* (IOC) [17] or *inverse reinforcement learning* (IRL) problem [18]. Here, the human selects an action, takes it, and receives a reward, which captures their internal preference (e.g. the enjoyment of having their desk organized a particular way). The robot only observes human actions and attempts to learn their preference under the assumption that the human likes the actions they selected; if you go for a particular desk configuration more frequently, the robot will assume that you like that configuration more.

Unfortunately, in practice, this natural inference assumes stationarity, which is often violated. We have all experienced situations where our preferences change with experience and time [19], [20]. This is particularly true in situations where we are *ourselves learning* about our preferences as we are providing them [21], [22]. For example, as we are organizing our desk, we might be experimenting with different configurations over time, to see what works.

Now imagine a personal robot is trying to help you organize that same desk. If the robot believes you are optimal, it will infer the wrong preferences. Instead, if the robot accounts for the fact that you are learning about your preferences, it has a better shot at understanding what you want. Even more crucially, the robot can expose you to new configurations – ones that might improve your posture, something you hadn't considered before and you were not going to explore if left to your own devices.

In this work, we formalize how a robot can actively assist humans who are themselves learning about their preferences. Our thesis is that by *modeling* and *influencing* the dynamics of human learning, the robot can enable the human-robot team to learn more effectively and outperform a human learning suboptimally in isolation.

To this end, we introduce the *assistive multi-armed bandit*: an extension of the classical *multi-armed bandit* (MAB) model of learning. In each round, the human selects an action, referred to in the bandit setting as an arm. However, the robot *intercepts* their intended action and chooses a (potentially different) arm to pull. The human then observes the pulled arm and corresponding reward, and the process repeats (Figure 1). We find this model surprisingly rich and fascinating. It captures the heart of collaboration: *information asymmetry* and the cost of equalizing it. As the human learns about their preferences,

they are compelled to communicate them to the robot as it decides their eventual reward. Analyzing this model allows us to understand the theoretical limits to assisting learning agents and the properties that make learners easier to assist.

Our contributions are the following: 1) we formalize the assistive multi-armed bandit; 2) we give weak sufficient conditions under which a human and robot team learns consistently, a lower bound on the cost of assuming noisy-optimality, and a mutual-information based upper bound on team performance; and 3) we use policy optimization [23] to conduct an in-depth empirical validation of our theoretical results and investigate the effect of incorrectly modeling the human's learning strategy. We train A against a fixed learning strategy, e.g., ϵ-greedy, and test it against a different learning strategy, e.g., Thompson sampling. *Our analysis shows a person that is better at learning does not necessarily lead to the human-robot team performing better - there are human learning strategies that are ineffective in isolation but communicate well and enable the robot to effectively assist.* In fact, human learning strategies that are *inconsistent* in isolation, that is, failing a weak notion of asymptotic optimality, can allow the human-robot team to match *optimal* performance in a standard multi-armed bandit. Our results advance the theory behind algorithmic preference learning and provide guidance for structuring algorithms for human-robot interaction.

II. A Family of Bandits

A. The Standard Multi-Armed Bandit

A *multi-armed bandit* (MAB) $\mathcal{M}$ is defined by:

- Θ: a space of reward distributions parameters; $\theta \in \Theta$;
- N: an integer representing the number of arms;
- p: a distribution over Θ.

At the start of the game, θ is sampled from Θ according to the prior p. At each timestep t, an arm $a_t \in [1, \ldots, N]$ is chosen. A reward $r_t \sim \theta_{a_t}$ is sampled from the corresponding arm distribution. A *strategy* is a mapping that determines the correct distribution to sample from given a history of reward observations and previous arm pulls: $K_t(a_1, r_1, \ldots, a_{t-1}, r_{t-1})$.

We use μ_k to represent the mean of arm k, with parameters θ_k. We use j^* to represent the index of the best arm and μ^* to represent its mean. $T_k(t)$ represents the number of pulls of arm k up to and including time t. The goal of this game is to maximize the sum of rewards over time, or alternatively, to minimize the expectation of the regret $\bar{R}(t)$, defined as:

$$\bar{R}(t) = \sum_t (\mu^* - \mu_{a_t}) = \sum_k (\mu^* - \mu_k)\, T_k(t). \qquad (1)$$

B. Stationary Inverse Optimal Control

In preference learning, e.g., inverse reinforcement learning (IRL) [24], [25] and inverse optimal control (IOC) [17], an AI system observes (noisily-)optimal behavior and infers the reward function or preferences of that agent. This relies on a key assumption that the agent being observed knows the value of actions it can take, at least in the sense that they are able to select optimal actions. In a multi-armed bandit setting, this set of assumptions corresponds to assuming that the human knows the parameters of the bandit, but has some small probability of picking a suboptimal arm. We refer to a human with this knowledge state and policy as implementing the *ϵ-optimal policy*. Inferring the reward of an ϵ-optimal, or noisily-optimal, human can be thought of as solving a stationary IOC problem.

C. The Inverse Multi-Armed Bandit

Before formalizing the problem of *assisting* a human who is learning, rather than noisily-optimal, we look at *passively inferring* the reward from their actions. We call this the *inverse bandit* problem. Each Inverse Bandit problem is defined by:

- $\mathcal{M}$: a multi-armed bandit problem
- H: a bandit strategy employed by the human, that maps histories of past actions and rewards to distributions over arm indices. $H_t : h_1 \times r_1 \times \cdots \times h_{t-1} \times r_{t-1} \to \Pi(N)$

The goal is to recover the reward parameter θ by observing *only* the arm pulls of the human over time $h_1, ..., h_t$.

Unlike the stationary IOC case, H does not have access to the true reward parameters. H receives the reward signal r_t sampled according to θ. As a consequence, the human arm pulls are not i.i.d.; the distribution of human arm pulls changes as they learn more about their preferences.

D. The Assistive Multi-Armed Bandit

In the *assistive multi-armed bandit*, we have a joint system $A \circ H$ that aims to do well in an MAB $\mathcal{M}$. This strategy consists of two parts: the human player H and robot player A. As in an MAB, the goal is to minimize the expected regret. The key difference between an assistive MAB and the standard MAB is that the policy is decomposed into a human component and a robot component. The goal is to capture scenarios where our goal, as designers of the robot A_t, is to optimize a reward signal which is only observed *implicitly* through the actions of a human who is themselves learning about the reward function.

The human and robot components of the policy are arranged in a setup similar to teleoperation. In each round:

1) The human player H selects an arm to suggest based on the history of previous arm pulls *and rewards*: $H_t(a_1, r_1, \ldots, a_{t-1}, r_{t-1}) \in [1, \ldots, N]$.
2) The robot player A selects which arm to actually execute based on the history of the human's attempts and the actual arms chosen: $A_t(h_1, a_1, \ldots, h_{t-1}, a_{t-1}, h_t) \in [1, \ldots, N]$.
3) The human player H observes the current round's arm and corresponding reward: $(a_t, r_t \sim \theta_{A_t})$.

Unlike the inverse MAB or (stationary) IOC, the assistive MAB formalizes the problem of actually using learned preference knowledge to assist a human. Even if we are able to solve the inverse MAB, this is not useful if we can't actually help a learner reduce regret. We expect an optimal solution to the assistive MAB to improve on suboptimal learning, guide exploration, and correct for noise.

III. Theoretical Results

A. Hardness of Assistive MABs

We consider the relative difficulty of assisting a person that knows what they want with assisting a person that is learning. We model the first situation as an assistive stationary IOC problem, and the second as an assistive MAB. First, we show that assistive stationary IOC is, as one might expect, quite easy in theory; we show that it is possible to infer the correct arm while making finitely many mistakes in expectation.

Proposition 1. *Suppose that H's arm pulls are i.i.d and let f_i be the probability H pulls arm i. If H is noisily optimal, that is, $f_{j^*} > f_i$ for all sub-optimal i, there exists a robot policy A that has finite expected regret for every value of θ:*

$$\mathbb{E}[\bar{R}(T)] \leq \sum_{i \neq j^*} \frac{\mu^* - \mu_i}{(\sqrt{f_{j^*}} - \sqrt{f_i})^2}$$

Proof. (Sketch) Our robot policy A simply pulls the most commonly pulled arm.

Let $\hat{f}_i(t) = \frac{1}{t}\sum_{k=1}^{t}[h_t = i]$ be the empirical frequency of H's pulls of arm i up to time t. Note that $A_t = i$ only if $\hat{f}_{j^*}(t) \leq \hat{f}_i(t)$. We apply a Chernoff bound to the random variable $\hat{f}_{j^*}(t) - \hat{f}_i(t)$. This gives that, for each i,

$$\Pr(\hat{f}_i(t) \leq \hat{f}_j(t)) \leq e^{-t(\sqrt{f_i} - \sqrt{f_j})^2}. \quad (2)$$

Summing Eq. 2 over t and suboptimal arms gives the result. □

This is in contrast to the standard results about regret in an MAB: for a fixed, nontrivial MAB problem $\mathcal{M}$, any MAB policy has expected regret at least logarithmic in time on some choice of parameter θ [26], [27]:

$$\mathbb{E}[\bar{R}(T)] \geq \Omega(log(T)).$$

Several approaches based on Upper Confidence Bounds (UCB) have been shown to achieve this bound, implying that this bound is tight [26], [28], [29]. Nonetheless, this suggests that the problem of assisting a noisily-optimal human is significantly easier than solving a standard MAB.

The assistive MAB is at least as hard as a standard MAB. For the same sequence of arm pulls and observed rewards, the amount of information available to A about the true reward parameters is upper bounded by the corresponding information available in a standard MAB. From a certain perspective, actually *improving* on human performance in isolation is hopelessly difficult – A does not get access to the reward signal, and somehow must still assist a person who does.

B. Consistent Assisted Learning

We begin with the simplest success criterion from the bandit literature: consistency. Informally, consistency is the property that the player eventually pulls suboptimal arms with probability 0. This can be stated formally as the average regret going to 0 in the limit: $\lim_{t\to\infty} \bar{R}(t)/t = 0$. In an MAB, achieving consistency is relatively straightforward: any policy that is greedy in the limit with infinite exploration (GLIE) is consistent [30], [31]. In contrast, in an assistive MAB, it is not obvious that the robot can implement such a policy when the H strategy is inconsistent. The robot observes no rewards and thus cannot estimate the best arm in hindsight.

However, it turns out a weak condition on the human allows the robot-human joint system to guarantee consistency:

Proposition 2. *If the human H implements a noisily greedy policy, that is, a policy that pulls the arm with highest sample mean strictly most often, then there exists a robot policy A such that $A \circ H$ is consistent.*

Proof. (Sketch) Fix a set of decaying disjoint exploration sequences E_k, one per arm, such that $\lim_{t\to\infty} \frac{1}{t}|E_k \cap \{1, ..., t\}| \to 0$ and $\lim_{t\to\infty} |E_k \cap \{1, ..., t\}| \to \infty$. In other words, each arm is pulled infinitely often, but at a decaying rate over time.

Let i_t be the arm most commonly pulled by H up until time t, and A be defined by

$$a_t = \begin{cases} k & t \in E_k \\ i_t & \text{otherwise} \end{cases}.$$

Note that this implies that for suboptimal k, $\frac{1}{t}T_k(t) \to 0$ in probability as $t \to \infty$, as the sample means of all the arms converge to the true means, and the rate of exploration decays to zero. This in turn implies that $A \circ H$ achieves consistency. □

In other words, assistance is possible if the human picks the best actions in hindsight. This robot A assists the human H in two ways. First, it helps the human explore their preferences – $A \circ H$ pulls every arm infinitely often. This fixes possible under-exploration in the human. Second, it stabilizes their actions and helps ensure that H does not take too many suboptimal actions - eventually, $A \circ H$ converges to only pulling the best arm. This helps mitigate the effect of noise from the human.

1) modeling learning as ϵ-optimality leads to inconsistency: We now investigate what occurs when mistakenly we model learning behavior as noisy-optimality.

A simple way to make $A \circ H$ consistent when H is noisily optimal is for A to pull the arm most frequently pulled by H.

Proposition 3. *If H plays a strategy that pulls the best arm most often and A plays H's most frequently pulled arm, then $A \circ H$ is consistent.*

Proof. (Sketch) Eventually, H's most frequent arm converges to the best arm with probability 1 by hypothesis. At this point, A will pull the best arm going forward and achieve a per-round regret of 0. □

Next we consider the impact of applying this strategy when its assumptions are incorrect, i.e., H is learning. For simplicity, we assume H is greedy and pulls the best arm given the rewards so far. We will consider a $1\frac{1}{2}$-arm bandit: a bandit with two arms, where one has a known expected value and the other is unknown. We show that pairing this suboptimal-learner with the 'most-frequent-arm' strategy leads the joint system $A \circ H$ to be *inconsistent*:

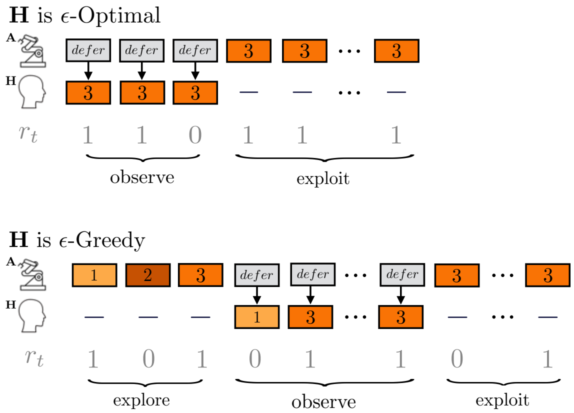

Fig. 2. A comparison between assisting an ϵ-optimal H and an ϵ-greedy H in a modified assistive MAB (defined in Section V-E) where the robot A has to choose between acting and letting H act. This creates a direct exploration-exploitation tradeoff that makes it easier to qualitatively analyze A's behavior. At the top is whether A defers to the human or pulls an arm, followed by what H pulls (if the robot defers), followed by the reward H observes. When the robot models learning, the policy it learns has a qualitative divide into three components: explore, where the robot explores for the human; observe, when the robot lets the human pull arms; and exploit, when the robot exploits this information and pulls its estimate of the best arm. Crucially, the explore component is only found when learning is modeled. This illustrates Proposition 4, which argues that assisting an ϵ-optimal H is different from assisting a learning H.

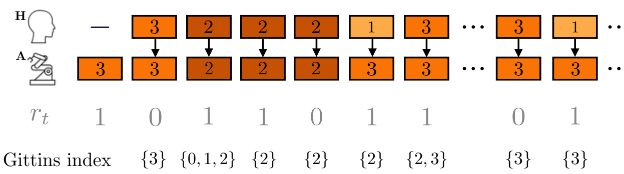

Fig. 3. In Proposition 5 we show that it is possible to match the regret from optimal learning in an standard MAB when assisting the 'win-stay-lose-shift' (WSLS) policy. This is because WSLS perfectly communicates the observed rewards to A. Here we show an example trajectory from an approximately optimal policy assisting WSLS (computed with Algorithm 1). At the top is what H suggests, followed by what A pulls, followed by the reward H observes. For comparison, we show the arms selected by the near-optimal Gittins index policy for each belief state. This highlights the importance of communicative learning policies in an assistive MAB.

Proposition 4. *If H is a greedy learner and A is 'most-frequent-arm', then there exists an assistive MAB $\mathcal{M}$ such that $A \circ H$ is inconsistent.*

Proof. (Sketch) The proof consists of two steps. First, we show a variant of a classical bandit result: if H and A output the constant arm in the same round, they will for the rest of time. Second, we show that this occurs with finite probability and get a positive lower bound on the per-round regret of $A \circ H$. □

While this is a simplified setting, this shows that the types of mistakes and suboptimality represented by learning systems *are not* well modeled by the standard suboptimality assumptions used in research on recommendation systems, preference learning, and human-robot interaction. The suboptimality exhibited by learning systems is stateful and self-reinforcing. Figure 2 shows the practical impact of modeling learning. It compares an optimal assistance policy for stationary IOC with an optimal policy for an assistive MAB. In general, assistive MAB policies seem to fit into three steps: *explore* to give H a good estimate of rewards; *observe* H to identify a good arm; and then *exploit* that information.

MABs are the standard theoretical model of reinforcement learning and so this observation highlights the point that the term inverse reinforcement learning is somewhat of a misnomer (as opposed to inverse optimal control): IRL's assumptions about an agent (noisy optimality) lead to very different inferences than *actually* assuming an agent is learning.

C. Regret in Assistive Multi-Armed Bandits

Having argued that we can achieve consistency for such a broad class of human policies in an assistive MAB, we now return to the question of achieving low regret. In particular, we investigate the conditions under which $A \circ H$ achieve $O(\log(T))$ expected regret, as is possible in the standard MAB.

For any given human H, there exists a robot A such that $A \circ H$ does as well as H: let A copy the H's actions without modification; that is, $a_t = h_t$ for all t. So in the case where H achieves $O(\log(T))$ regret by itself, $A \circ H$ can as well.

However, a more interesting question is that of when we can successfully assist an suboptimal H that achieves $\omega(\log(T))$ regret. *While one may hypothesize that better human policies lead to better performance when assisted, this is surprisingly not the case, as the next section demonstrates.*

1) An inconsistent policy that is easy to assist: Consider a Beta-Bernoulli assistive MAB where rewards are binary: $r_t \in \{0, 1\}$. A classic bandit strategy here is 'win-stay-lose-shift' (WSLS) [32] which, as the name suggests, sticks with the current arm if the most recent reward is one:

$$h_t = \begin{cases} a_{t-1} & r_{t-1} = 1 \\ \textbf{Unif}(\{k | k \neq a_{t-1}\}) & r_{t-1} = 0 \end{cases} \quad (3)$$

This a simple strategy that performs somewhat well empirically – although it is easy to see that it is not consistent in isolation, let alone capable of achieving $O(\log(T))$ regret. Indeed, it achieves $\Theta(T)$ regret, as it spends a fixed fraction of its time pulling suboptimal arms. However, if H can implement this strategy, the combined system can implement an arbitrary MAB strategy from the standard MAB setting, including those that achieve logarithmic regret. In other words, the robot can successfully assist the human in efficiently balancing exploration and exploitation *despite only having access to the reward parameter through an inconsistent human.*

Proposition 5. *Let R^* be the optimal regret for a Beta-Bernoulli multi-armed bandit. If H implements the WSLS strategy in the corresponding assistive MAB, then there exists a robot strategy A such that $A \circ H$ achieves regret R^*.*

Proof. (Sketch) The pair (a_{t-1}, h_t) directly encodes the previous reward r_{t-1}. This means that A_t can be an arbitrary function of the history of arm pulls and rewards and so it can implement the MAB policy that achieves regret R^*. □

Figure 3 compares a rollout of this $A \circ H$(we describe the approach in Section V-B1) with a rollout of a near-optimal policy for a standard MAB.

2) Communication upper bounds team performance: The WSLS policy is not unique in that it allows $A \circ H$ to obtain logarithmic regret. A less interesting, but similarly effective policy, is for the human to directly encode their reward observations into their actions; the human need not implement a sensible bandit policy. For example, the following purely communicative H also works for a Beta-Bernoulli bandit:

$$h_t = \begin{cases} 0 & r_{t-1} = 0 \\ 1 & r_{t-1} = 1 \end{cases} \tag{4}$$

We can generalize the results regarding communicative policies using the notion of mutual information, which quantifies the amount of information obtained through observing the human arm pulls.

Let $\mathbf{I}(X;Y)$ be the mutual information between X and Y, $\mathbf{H}(X)$ be the entropy of X, and $\mathbf{H}(X|Y)$ be the entropy of X given Y.

Proposition 6. *Suppose that the probability the robot pulls a suboptimal arm at time t is bounded above by some function $f(t)$, that is $P(A_t \neq j^*) \leq f(t)$. Then the mutual information $\mathbf{I}(j^*; h_1 \times \cdots \times h_t)$ between the human actions up to time t and the optimal arm must be at least $(1 - f(t)) \log N - 1$.*

Proof. We can consider the multi-armed bandit task as one of deducing the best arm from the human's actions. This allows us to apply Fano's inequality [33] to $P(A_t \neq j^*)$, and using the fact that the entropy of a Bernoulli random variable is bounded above by 1, we get

$$\begin{aligned} P(A_t \neq j^*) \log(N-1) &\geq \mathbf{H}(\hat{j}^* | j^*) - \log 2 \\ &= \mathbf{H}(\hat{j}^*) - \mathbf{I}(\hat{j}^*; j^*) - 1 \\ &\geq \log N - \mathbf{I}(j^*; h_1 \times \cdots \times h_t) - 1. \end{aligned}$$

Rearranging terms and using $P(A_t \neq j^*) \leq f(t)$, we get

$$\begin{aligned} \mathbf{I}(j^*; h_1 \times \cdots \times h_t) &\geq \log N - f(t) \log(N-1) - 1 \\ &\geq (1 - f(t)) \log N - 1. \end{aligned}$$

□

Intuitively, since the probability of error is bounded by $f(t)$, in $(1 - f(t))$ cases the human actions conveyed enough information for A to successfully choose the best action out of N options. This corresponds to $\log N$ bits, so there needs to be at least $(1 - f(t)) \log N$ bits of information in H's actions.

Corollary 7. *Suppose that the probability the robot pulls a suboptimal arm at time t is bounded above by some function $f(t)$, that is $P(A_t \neq j^*) \leq f(t)$. Then the mutual information $\mathbf{I}(a_1 \times r_1 \times \cdots \times a_{t-1} \times r_{t-1}; h_1 \times \cdots \times h_t)$ between the human actions up to time t and the human observations must be at least $(1 - f(t)) \log N - 1$.*

Proof. Since the best arm is independent of the human actions given the human observations, this follows immediately from the data processing inequality and proposition 6. □

In order to achieve regret logarithmic in time, we must have that $P(K_t \neq j^*) \leq \frac{C}{t}$ for some $C > 0$. Applying proposition 6 above implies that we must have

$$\mathbf{I}(j^*; h_1 \times \cdots \times h_t) \geq (1 - \frac{C}{t}) \log N - 1$$

Note that the term $\mathbf{I}(\hat{j}^*; h_1 \times \cdots \times h_t)$ depends on *both* the human policy and the robot policy - no learning human policy can achieve this bound unless the human-robot system $A \circ H$ samples each arm sufficiently often. *As a consequence, simple strategies such as inferring the best arm at each timestep and pulling it, cannot achieve the $\Theta(\log T)$ lower bound on regret.*

IV. Algorithms for Assistive Multi-Armed Bandits

The optimal response to a given human strategy can be computed by solving a partially observed Markov decision process (POMDP) [34]. The state is the reward parameters θ and H's internal state. The observations are the human arm pulls. In this framing, a variety of approaches can be used to compute policies or plans, e.g., online Monte-Carlo planning [35], [36] or point-based value iteration [37].

In order to run experiments with large sample sizes, our primary design criterion was fast online performance. This lead us to use a direct policy optimization approach. The high per-action cost of Monte-Carlo planners makes them impractical for this problem. Further, explicitly tracking θ and H's internal state is strictly harder than solving the inverse MAB.

Algorithm 1 Policy Optimization for the Assistive MAB

```
human policy H
initialize parameterized policy π(w; ·), policy parameters w
for i ≤ nItrs do
    ξs, rs ← SAMPLE-TRAJECTORIES(π(w; ·), Size, T)
    ∂̂w ← POLICY-GRADIENT(ξs, π(w; ·))          ▷ [38]
    w  w + ∂̂w
end for

procedure SAMPLE-TRAJECTORIES(π(w, ·), Size, T)
    initialize empty array ξs
    for i ≤ Size do
        θ ∼ Θ
        for t ≤ T do
            h_t ∼ H_t(h_1, r_1, ..., a_{t-1}, r_{t-1})
            a_t ∼ π(w; h_1, a_1, ..., h_{t-1}, a_{t-1}, h_t)
            r_t ∼ θ_{a_t}
        end for
        ξ  ξ = [(h_1, a_1, r_1), ..., (h_T, a_T, r_T)]
        ξs  ξs+[ξ]
    end for
return ξs
end procedure
```

Our approach applies the policy optimization algorithm of [23] to assistive MABs. Given an assistive MAB $(\mathcal{M}, H)$, we sample a batch of reward parameters θ from the prior $p(\Theta)$; generate trajectories of the form $\xi = [(h_1, a_1, r_1), ..., (h_t, a_t, r_t)]$ from H and the current robot policy $A^{(i)}$; and use the

trajectories to update the robot policy to $A^{(i+1)}$. During this offline training stage, since we are sampling reward parameters rather than using the ground truth reward parameters, we can use the generated rewards r_t to improve on $A^{(i)}$.

We represent A's policy as a recurrent neural network (RNN). At each timestep, it observes a tuple (a_{t-1}, h_t) where a_{t-1} is the most recent robot action and h_t is the most recent human action. In response, it outputs a distribution over arm indices, from which an action is sampled. Given a batch of trajectories, we use an approximate policy gradient method [38] to update the weights of our RNN[1]. We summarize this procedure in Algorithm 1. In our experiments, we used Proximal Policy Optimization (PPO) [39], due to its ease of implementation, good performance, and relative insensitivity to hyperparameters.

V. Experiments

In our experiments, we used a horizon 50 Beta-Bernoulli bandit with four arms. Pulling the ith arm produces a reward of one with probability θ_i and zero with probability $1 - \theta_i$: $\Theta = [0, 1]^4$. We assume a uniform prior over Θ: $\theta_i \sim \text{Beta}(1, 1)$.

We consider 5 classes of human policy:

- **ϵ-greedy**, a learning H that chooses the best arm in hindsight with probability $1 - \epsilon$ and a random arm with probability ϵ.[2]
- **WSLS**, the *win-stay-lose-shift* policy [32] sticks with the arm pulled in the last round if it returned 1, and otherwise switches randomly to another arm.
- **TS**, the *Thompson-sampling* policy [40] maintains a posterior over the arm parameters, and chooses each arm in proportion to the current probability it is optimal. This is implemented by sampling a particle from the posterior of each arm, then pulling the arm associated with the highest value.
- **UCL**, the *upper-credible limit* policy [41] is an algorithm similar to Bayes UCB [42] with softmax noise, used as a model of human behavior in a bandit environment.[3]
- **GI**, the *Gittins index* policy [43] is the Bayesian optimal solution to an infinite horizon discounted objective MAB.[4]

In addition, we also defined the following noisily-optimal human policy to serve as a baseline:

- **ϵ-optimal**, a *fully informed* H that knows the reward parameters θ, chooses the optimal arm with probability $1 - \epsilon$, and chooses a random action with probability ϵ.[5]

A. Inverse Multi-Armed Bandit

Our first experiment investigates the miscalibration that occurs when we do not model learning behavior. A robot that doesn't model human learning will be overconfident. To show this, we use the Metropolis-Hastings algorithm [45] to approximate the posterior over reward parameters θ given human actions. We compare the posterior we get when we model learning with the posterior that assumes H is ϵ-optimal.

TABLE I
Log-density of true reward params in a horizon 5 inverse MAB

	Assumed H Policy	
Actual H Policy	**Correct Policy**	**ϵ-Optimal**
ϵ-greedy	0.49	-0.23
WSLS	0.95	0.13
TS	0.02	-0.23
UCL	0.03	-0.30
GI	0.94	0.20
ϵ-Optimal	–	1.55

We compared the log-density of the true reward parameters in the posterior conditioned on 5 H actions, under both models, when H is actually learning. We report the results in Table I. For every learning human policy, we find that the log-density of the true reward parameters is significantly higher when we model learning than when we do not. In the case of ϵ-greedy and TS, we find that the posterior that fails to model learning assigns negative log-density to the true parameters. This means the posterior is a *worse estimate* of θ than the prior.

B. Assistance is Possible

Propositions 2 and 5 prove that it is possible to assist suboptimal learners in theory. In this section, we show that it is possible in practice. We use Algorithm 1 to train a recurrent policy for each human policy. In Fig. 4, we report the performance with assistance and without.

1) Policy optimization details: To alleviate the problem of exploding and vanishing gradients [46], we use Gated Recurrent Units (GRU) [47] as the cells of our recurrent neural network. The output of the GRU cell is fed into a softmax function, and this output is interpreted as the distribution over actions. To reduce to variance in our policy gradient estimate, we also use a

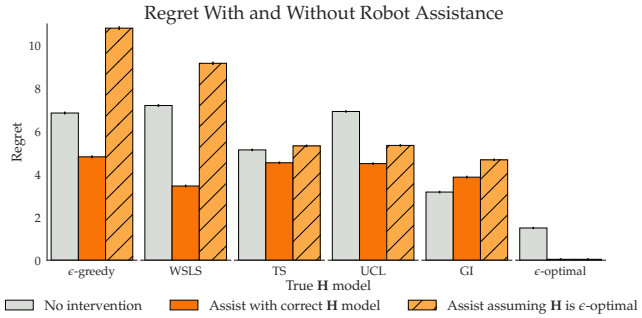

Fig. 4. Averaged regret of various human policies (lower = better) over 100,000 trajectories when unassisted, assisted with the correct human model, and assisted assuming that the human is noisily-optimal. Assistance lowers the regret of most learning policies, but it is important to model learning: ignoring that the human is learning can lead to worse performance than no assistance. Note that assisted WSLS performs almost as well as the Gittins Index policy, an empirical verification of Proposition 5.

[1]Our code is available online at https://github.com/chanlaw/assistive-bandits

[2]We performed grid search to pick an ϵ based on empirical performance, and found that $\epsilon = 0.1$ performed best.

[3]We set $K = 4$ and softmax temperature $\tau = 4$.

[4]We follow the approximations described by Chakravorty and Mahajan in [44], and choose a discount rate ($\gamma = 0.9$) that performs best empirically using grid search.

[5]We set ϵ to match that of the ϵ-greedy policy.

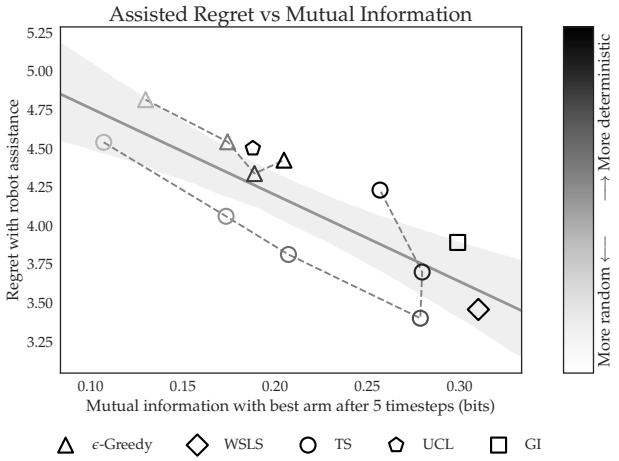

Fig. 5. The assisted regret of various policies, plotted against the mutual information between the best arm and the policy's actions in the first 5 timesteps. We also plot the best-fit line, with 95% confidence interval, for the regression between assisted regret and mutual information. We augmented our policies with variants of ϵ-greedy and Thompson sampling with less randomness. Policies with high mutual information lead to lower regret when assisted, supporting our theoretical findings.

value function baseline [48] and apply Generalized Advantage Estimation (GAE) [49]. We used weight normalization [50] to speed up training. We used a batch size of 250000 timesteps or 5000 trajectories per policy iteration, and performed 100 policy iterations using PPO.

2) Results: To quantitatively test the hypotheses that our learned models successfully assist, we perform a two factor ANOVA. We found a significant interaction effect, $F(3, 999990) = 1778.8$, $p < .001$, and a post-hoc analysis with Tukey HSD corrections showed that we were able to successfully assist the human in all four sub-optimal learning policies ($p < .001$).

Our WSLS results agree with Proposition 5. Assisted WSLS achieves a regret of 3.5, close to the regret of the best-performing unassisted policy, 3.2. The gap in reward is due to our choice to employ approximate policy optimization. We provide an example trajectory in Fig. 3. The actions selected are almost identical to those of the optimal policy.

This suboptimality also accounts for the small increase in regret when assisting the Gittins index policy.

3) Modeling learning matters: Proposition 4 shows that modeling learning matters. We compared assistance assuming that H is ϵ-optimal with assistance with the correct (learning) model. We report the results in Fig. 4. We found that the regret with the wrong model is higher than no intervention in every case but UCL. Assisted WSLS with the wrong model has double the regret of assisted WSLS with the correct model. Proposition 4 shows that in theory, ignoring learning leads to inconsistency. Fig. 4 shows that this mistake leads to higher regret empirically.

TABLE II
INCREASE IN REWARD FROM ROBOT ASSISTANCE

	Assumed H Policy					
Actual H	**ϵ-greedy**	**WSLS**	**TS**	**UCL**	**GI**	**ϵ-optimal**
ϵ-greedy	**2.13**	-0.60	-2.18	-2.20	-0.11	-3.95
WSLS	0.94	**3.75**	0.80	0.10	-2.21	-1.97
TS	0.33	**0.66**	0.60	0.44	-1.53	-0.19
UCL	1.76	-1.19	**2.51**	2.43	0.74	1.28
GI	-1.09	**-0.28**	-0.77	-0.85	-0.71	-1.50
ϵ-optimal	0.24	1.17	1.24	1.28	-3.09	**1.46**

C. Mutual Information Predicts Performance

Proposition 6 implies that high mutual information is required for good team performance. To verify this, we computed the mutual information for a variety of combined policies after 5 timesteps. Fig. 5 plots this against the regret of the combined system. We consider several variants of ϵ-greedy and TS that are more or less deterministic. We consider $\epsilon \in [0, 0.02, 0.05, 0.1]$. To make TS more deterministic, we use the mean of a sample of n particles to select arms. We consider $n \in [1, 2, 3, 10, 30, \infty]$.

Across this data, higher mutual information is associated with lower assisted regret, $r(10) = -.82$, $p < .001$. Furthermore, by looking at the ϵ-greedy and TS results as a sequence, we can observe a clear and distinct pattern. Policies that are more deterministic tend to be easier to help. This is supported by the results in Table I, which shows that it is easier to infer reward parameters for WSLS and GI (i.e., the two policies with the highest mutual information) than TS and ϵ-greedy.

D. Sensitivity to Model Misspecification

In the previous three experiments, we assumed knowledge of the correct learning policy. In this experiment, we consider the implications of incorrectly modeling learning. We took the policies we trained in Section V-B and tested them with every human policy. We report the net change in reward in Table II. We colored cases where the robot A successfully assists the human H green, and cases where it fails to assist red. We bolded the best performance in each row.

Modeling learning (even with the incorrect model) generally leads to lower regret than assuming ϵ-optimal for every learning H policy. However, when the robot has the wrong model of learning, it can fail to assist the human. For example, ϵ-greedy is only successfully assisted when it is correctly modeled. This argues that research into the learning strategies employed by people in practice is an important area for future research.

An intriguing result is that assuming ϵ-greedy *does* successfully assist all of the suboptimal learning policies. This suggests that, although some learning policies must be well modeled, learning to assist some models can be transferred to other models in some cases. On the other hand, trying to assist GI leads to a policy that hurts performance across the board. In future work, we plan to identify classes of learners which can be assisted by the same robot policy.

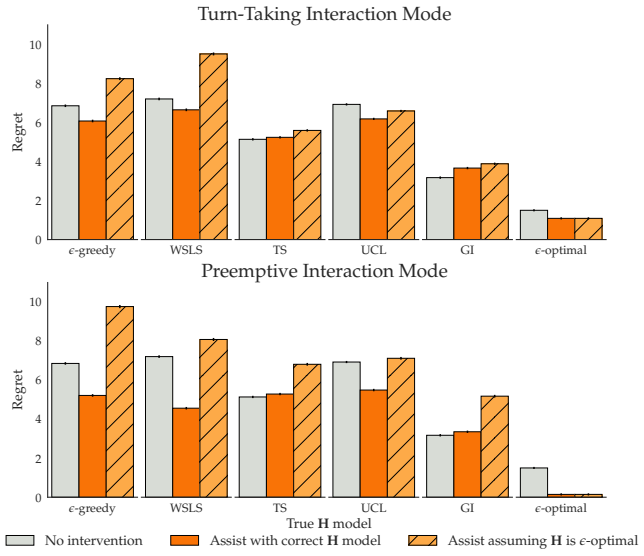

Fig. 6. Averaged regret of various human policies (lower = better) over 100,000 trajectories under different interaction modes. Assistance lowers the regret of ϵ-greedy, WSLS, and UCL in both the preemptive and turn-taking interaction modes. Assistance while ignoring learning is worse than no assistance in almost every case. This offers further support for the importance of modeling learning when assisting humans.

E. Other Paradigms of Assistance

The assistive multi-armed bandit considered so far only captures one mode of interaction. It is straightforward to consider extensions to different modes. We consider two such modes. The first is turn taking, where H and A take turns selecting arms. This can be more difficult because the robot has to act in early rounds, when it has less information, and because the human has to act in later rounds, when H may be noisy and the best arm has already been identified.

The second variant we consider is preemptive interaction. In this case, A goes first and either pulls an arm or lets H act. This creates an exploration-exploitation tradeoff. A only observes H's arm pulls by actually allowing H to pull arms and so it must choose between observing H's behavior and exploiting that knowledge.

Fig. 6 shows the experiment from Section V-B applied to each of these interaction modes. The results are largely similar to those of teleoperation. We are able to assist the suboptimal policies and modeling learning as ϵ-optimality increases regret in all cases. We see that WSLS is a less attractive policy in these settings: because H actions are always executed when they are observed, it no longer makes sense for H to employ a *purely* communicative policy. However, we do still see results that confirm Proposition 6: more deterministic policies that reveal more information are easier to help.

1) Explore, observe, then exploit: In looking at the policies learned for the preemptive interaction mode, we see an interesting pattern emerge. Because the policy has to choose between selecting arms directly and observing H, by looking at rollouts of the learned policy we can determine when it is observing the human. We find that a clear pattern emerges. A initially *explores* for H: it selects arms uniformly to give H a good estimate of θ. Then, A *observes* H's arm pulls to identify the optimal arm. For the final rounds, A *exploits* this information and pulls its estimate of the optimal arm. Fig. 2 compares a representative trajectory with one that is optimized against an ϵ-optimal H.

VI. Discussion

A. Summary

In this work, we studied the problem of assisting a human who is learning about their own preferences. Our central thesis is that by modeling and influencing the dynamics of a human's learning, we can create robots that can better assist people in achieving their preferences. We formalized this as the assistive multi-armed bandit problem, which extends the multi-armed bandit to account for teleoperation and human learning. We analyzed our formalism theoretically, then used policy optimization in proof-of-concept experiments that supported our theoretical results. Surprisingly, we found that a person that is better at learning in isolation does not necessarily lead to a human-robot team that performs better. We highlighted a theoretical connection between the amount of information communicated by the human policy and the best assisted performance, which we validated in our experiments.

B. Limitations and Future Work.

1) Stateful environments: One significant limitation of this work is that we assume the environment the human is acting in is stateless. In practice, the environmental state changes over time, and the state can greatly influence the reward associated with certain actions. This suggests natural extensions of the assistive multi-armed bandit to the contextual bandit [51] and full Markov decision process (MDP) [52] settings.

2) Realistic human policies: Another significant limitation of this work is the use of simple bandit policies in place of actual human policies. In addition, we do not have access to the true human policy in any case. Future work can remedy this by incorporating more realistic policies, and can study to what extent these results generalize to assisting actual humans.

C. Closing Remarks

The assistive multi-armed bandit is representative of a world where robots are supposed to assist people, even though people haven't figured out what they want yet. Laying down the theoretical foundations for these kinds of interaction paradigms is an important and under-served aspect of HRI.

Acknowledgements

We thank the members of the InterACT lab and the Center for Human Compatible AI for helpful advice. This work was partially supported by OpenPhil, AFOSR, NSF, and NVIDIA.

REFERENCES

[1] J. Fürnkranz and E. Hüllermeier, "Preference learning," in *Encyclopedia of Machine Learning*. Springer, 2011, pp. 789–795.

[2] H. Sakagami and T. Kamba, "Learning personal preferences on online newspaper articles from user behaviors," *Computer Networks and ISDN Systems*, vol. 29, no. 8-13, pp. 1447–1455, 1997.

[3] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 661–670.

[4] Z. Zhao, H. Lu, D. Cai, X. He, and Y. Zhuang, "User preference learning for online social recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2522–2534, 2016.

[5] C. Basu, H. Hirsh, W. Cohen *et al.*, "Recommendation as classification: Using social and content-based information in recommendation," in *Aaai/iaai*, 1998, pp. 714–720.

[6] D. Goel and D. Batra, "Predicting user preference for movies using netflix database," *Department of Electrical and Computer Engineering, Carniege Mellon University*, 2009.

[7] H. Wang and H. Zhang, "Movie genre preference prediction using machine learning for customer-based information," in *Computing and Communication Workshop and Conference (CCWC), 2018 IEEE 8th Annual*. IEEE, 2018, pp. 110–116.

[8] B. Akgun, M. Cakmak, K. Jiang, and A. L. Thomaz, "Keyframe-based learning from demonstration," *International Journal of Social Robotics*, vol. 4, no. 4, pp. 343–355, 2012.

[9] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation." in *Robotics: science and systems*, 2012.

[10] K. Fischer, F. Kirstein, L. C. Jensen, N. Krüger, K. Kukliński, T. R. Savarimuthu *et al.*, "A comparison of types of robot control for programming by demonstration," in *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*. IEEE Press, 2016, pp. 213–220.

[11] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. Dragan, "Information gathering actions over human internal state," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 66–73.

[12] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan, "Learning from physical human corrections, one feature at a time," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2018, pp. 141–149.

[13] E. Beigman and R. Vohra, "Learning from revealed preference," in *Proceedings of the 7th ACM Conference on Electronic Commerce*. ACM, 2006, pp. 36–42.

[14] M. Zadimoghaddam and A. Roth, "Efficiently learning from revealed preference," in *International Workshop on Internet and Network Economics*. Springer, 2012, pp. 114–127.

[15] M.-F. Balcan, A. Daniely, R. Mehta, R. Urner, and V. V. Vazirani, "Learning economic parameters from revealed preferences," in *International Conference on Web and Internet Economics*. Springer, 2014, pp. 338–353.

[16] D. C. Kingsley and T. C. Brown, "Preference uncertainty, preference learning, and paired comparison experiments," *Land Economics*, vol. 86, no. 3, pp. 530–544, 2010.

[17] R. E. Kalman, "When is a linear control system optimal?" *Journal of Basic Engineering*, vol. 86, no. 1, pp. 51–60, 1964.

[18] S. Russell, "Learning agents for uncertain environments," in *Proceedings of the eleventh annual conference on Computational learning theory*. ACM, 1998, pp. 101–103.

[19] M. Allais, "The so-called allais paradox and rational decisions under uncertainty," in *Expected utility hypotheses and the Allais paradox*. Springer, 1979, pp. 437–681.

[20] J. Baron, *Thinking and deciding*. Cambridge University Press, 2000.

[21] R. M. Cyert and M. H. DeGroot, "Adaptive utility," in *Adaptive Economic Models*. Elsevier, 1975, pp. 223–246.

[22] J. F. Shogren, J. A. List, and D. J. Hayes, "Preference learning in consecutive experimental auctions," *American Journal of Agricultural Economics*, vol. 82, no. 4, pp. 1016–1021, 2000.

[23] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "Rl2: Fast reinforcement learning via slow reinforcement learning," *arXiv preprint arXiv:1611.02779*, 2016.

[24] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning." in *Icml*, 2000, pp. 663–670.

[25] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 1.

[26] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[27] S. Mannor and J. N. Tsitsiklis, "The sample complexity of exploration in the multi-armed bandit problem," *Journal of Machine Learning Research*, vol. 5, no. Jun, pp. 623–648, 2004.

[28] R. Agrawal, "Sample mean based index policies by o (log n) regret for the multi-armed bandit problem," *Advances in Applied Probability*, vol. 27, no. 4, pp. 1054–1078, 1995.

[29] O. Cappé, A. Garivier, O.-A. Maillard, R. Munos, G. Stoltz *et al.*, "Kullback–leibler upper confidence bounds for optimal sequential allocation," *The Annals of Statistics*, vol. 41, no. 3, pp. 1516–1541, 2013.

[30] H. Robbins, "Some aspects of the sequential design of experiments," in *Bulletin of the American Mathematical Society.*, 1952.

[31] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Machine Learning*, vol. 38, no. 3, pp. 287–308, Mar 2000. [Online]. Available: https://doi.org/10.1023/A:1007678930559

[32] H. Robbins, "Some aspects of the sequential design of experiments," in *Herbert Robbins Selected Papers*. Springer, 1985, pp. 169–177.

[33] R. M. Fano, "Fano inequality," *Scholarpedia*, vol. 3, no. 10, p. 6648, 2008.

[34] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[35] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," in *Advances in neural information processing systems*, 2010, pp. 2164–2172.

[36] A. Guez, D. Silver, and P. Dayan, "Efficient bayes-adaptive reinforcement learning using sample-based search," in *Advances in Neural Information Processing Systems*, 2012, pp. 1025–1033.

[37] J. Pineau, G. Gordon, S. Thrun *et al.*, "Point-based value iteration: An anytime algorithm for pomdps," in *IJCAI*, vol. 3, 2003, pp. 1025–1032.

[38] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[40] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

[41] P. B. Reverdy, V. Srivastava, and N. E. Leonard, "Modeling human decision making in generalized gaussian multiarmed bandits," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 544–571, 2014.

[42] E. Kaufmann, O. Cappé, and A. Garivier, "On bayesian upper confidence bounds for bandit problems," in *Artificial Intelligence and Statistics*, 2012, pp. 592–600.

[43] J. C. Gittins, "Bandit processes and dynamic allocation indices," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 148–177, 1979.

[44] J. Chakravorty and A. Mahajan, "Multi-armed bandits, gittins index, and its calculation," *Methods and Applications of Statistics in Clinical Trials: Planning, Analysis, and Inferential Methods, Volume 2*, pp. 416–435, 2014.

[45] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[46] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[47] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[48] T. Degris, P. M. Pilarski, and R. S. Sutton, "Model-free reinforcement learning with continuous action in practice," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 2177–2182.

[49] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[50] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 901–909.
[51] A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. Schapire, "Taming the monster: A fast and simple algorithm for contextual bandits," in *International Conference on Machine Learning*, 2014, pp. 1638–1646.
[52] R. Bellman, "A markovian decision process," *Journal of Mathematics and Mechanics*, vol. 6, no. 5, pp. 679–684, 1957. [Online]. Available: http://www.jstor.org/stable/24900506