Replication-proof Bandit Mechanism Design with Bayesian Agents

Suho Shin[1], **Seyed A. Esmaeili**[2], **MohammadTaghi Hajiaghayi**[1]

[1] University of Maryland
[2] University of Chicago
(suhoshin,hajiagha)@umd.edu,esmaeili@uchicago.edu

Abstract

We study the problem of designing replication-proof bandit mechanisms when agents strategically register or replicate their own arms to maximize their payoff. Specifically, we consider Bayesian agents who only know the distribution from which their own arms' mean rewards are sampled, unlike the original setting of by Shin, Lee and Ok AISTATS'22. Interestingly, with Bayesian agents in stark contrast to the previous work, analyzing the replication-proofness of an algorithm becomes significantly complicated even in a single-agent setting. We provide sufficient and necessary conditions for an algorithm to be replication-proof in the single-agent setting, and present an algorithm that satisfies these properties. These results center around several analytical theorems that focus on *comparing the expected regret of multiple bandit instances*, and therefore might be of independent interest since they have not been studied before to the best of our knowledge. We expand this result to the multi-agent setting, and provide a replication-proof algorithm for any problem instance. We finalize our result by proving its sublinear regret upper bound which matches that of Shin, Lee and Ok AISTATS'22.

1 Introduction

Multi-armed bandit (MAB) algorithms are an important paradigm for interactive learning that focuses on quantifying the trade-off between exploration and exploitation (Slivkins et al. 2019), with various real-world applications such as recommender systems (Barraza-Urbina and Glowacka 2020; Tang et al. 2014), dynamic pricing by (Badanidiyuru, Kleinberg, and Slivkins 2018; Badanidiyuru, Langford, and Slivkins 2014), and clinical trials by (Djallel and Irina 2019) to name a few. They are used to model various practical applications where there is a set of actions (arms) to be selected (pulled) over a collection of rounds. Pulling an arm yields a reward sampled from its distribution which is generally different from the other arms' distributions. The objective is to maximize the sum of rewards accumulated through the rounds or equivalently minimize the *regret* with respect to the optimal clairvoyant arm choice.

In deploying MAB algorithms in real-world applications, however, the rewards are often not coming from stochastic distributions, but from a collection of rational individuals (agents) who seek to maximize their own payoff. Consequently, there has been a surge of interest in bandit problem from the mechanism design perspective, which studies MAB algorithms that induce a good outcome with respect to agents' strategic behavior, while maintaining its efficiency. Examples of such settings include two-sided matching markets, where agents learn their preferences through sequential interactions (Liu et al. 2021; Liu, Mania, and Jordan 2020), and learning in environments where agents can strategically modify the realized rewards of their arms (Braverman et al. 2019; Feng, Parkes, and Xu 2020).

Recently, (Shin, Lee, and Ok 2022) considered a problem where agents register arms to a platform that uses a MAB algorithm to minimize regret. Each agent has an original set of arms, and the mean reward of each arm is known to him a priori, *i.e.,* agents are *fully-informed*. Each agent receives a *fixed constant portion* of the reward whenever its arm is pulled, and tries to maximize cumulative reward. Importantly, each agent can potentially register the same arm *more than once* (replication) at free cost, which possibly ruins the exploration-exploitation balance of the bandit algorithm. Indeed, (Shin, Lee, and Ok 2022) show that "ordinary" algorithms such as upper-confidence-bound (UCB) induce rational agents to replicate infinitely many times, and thus the algorithm suffers linear regret. To address this problem, (Shin, Lee, and Ok 2022) propose a hierarchical algorithm with two phases, and prove its truthfulness such that each agent's dominant strategy is not to replicate.

Although the introduction of replication-proof mechanism is remarkable, the assumption that each agent is fully informed about its own arms is theoretically strong and practically limited. Intuitively, if an agent knows all of its own arms' mean rewards, there exists no reason for the agent to *consider suboptimal arms*. Indeed, this assumption significantly simplifies the equilibrium analysis so that the problem simply reduces to the case under which each agent only has a *single arm*. Then, it is immediate to check that any bandit algorithm is replication-proof under a single-agent setting. Moreover, in practice, it is more reasonable to assume that each agent only has partial information regarding its own arms. For instance, in online content platforms, each content provider usually does not know the quality of each content that has not been posted yet, but only can guess its outcome, based on the data possibly acquired from the previous con-

tents he/she may have.

In this context, we study a *Bayesian extension*[1] of the replication-proof bandit mechanism design problem, where each agent only knows such a distribution from which its own arm's mean reward is sampled.[2] Given the prior distribution, each agent computes its ex-ante payoff by taking expectation over all the possible realization of its own arms, and commits to a strategy that maximizes the payoff. Importantly, we aim to obtain a *dominant strategy incentive-compatible* (DSIC) mechanism such that each agent has no incentive to deviate from its truthful strategy, *i.e.,* submitting all the original arms without any replication, regardless of the others' strategies.

While our extension itself is intuitive and simple, this brings *significant challenges* in analyzing an algorithm's equilibrium compared to the fully-informed agent setting. Notably, even in a single-agent case, it is not trivial to obtain a replication-proof algorithm. This is in stark contrast to the fully-informed setting under which any bandit algorithm immediately satisfies replication-proofness. The question of designing replication-proofness in the single-agent setting spawns an intriguing question of *comparing expected regret of multiple bandit instances*, which will be elaborated shortly.

Outline of the paper. In the following subsection, we present our main contributions. Due to the page limit, we defer our discussion on related works to Appendix A. Then, we introduce the formal problem setup and preliminaries, and the failure of existing algorithms and its implication. Next we develop our main intuition on constructing replication-proof algorithm by investigating the single-agent setting. Correspondingly, we extend this result and provide our main result on replication-proof algorithm with sublinear regret. Practical motivations, discussion on an extension to asymmetric setting, details on the failure of H-UCB, and all the proofs are deferred to the appendix. Due to the page limit, all the appendices can be found in the full paper (Esmaeili, Hajiaghayi, and Shin 2023).

1.1 Contributions and Techniques

Overall, our main results for replication-proof algorithms are presented in a step-by-step manner. We start with investigating the previous algorithm H-UCB suggested by (Shin, Lee, and Ok 2022), which mainly builds upon standard UCB along with the idea of hierarchical structure. Unlike from the fully-informed setting under which any bandit algorithm is replication-proof in the single-agent setting, we reveal that the standard UCB algorithm fails to be replication-proof even with a single-agent, regardless of the exploration parameter f, due to its deterministic (up to tie-break) and adaptive nature. This phenomenon carries over to the multi-agent setting, so that UCB with hierarchical structure (H-UCB) fails to be replication-proof, regardless of the number of agents.

To tackle this challenge, we first thoroughly investigate the single-agent setting, and characterize necessary and sufficient conditions for an algorithm to be replication-proof. Let us provide a simple example to explain the main technical challenges one would face in analyzing equilibriums.

Example 1. Consider a single agent with two Bernoulli arms, say arm a and b. The prior distribution is given by $Bern(0.5)$, *i.e.,* the expectation of both the arms' distributions are sampled from the prior distribution $Bern(0.5)$. The agent receives a constant portion, say 40% of the realized reward, whenever their arms are pulled. There exist four cases in terms of the pair of realized mean rewards (μ_a, μ_b): (i) $(1, 1)$, (ii) $(1, 0)$, (iii) $(0, 1)$, and (iv) $(0, 0)$, all of which occurs with probability 0.25. Due to the nature of Bayesian agents, the agent cannot observe which arm has larger mean rewards, and suppose that the agent somehow commits to a strategy that only replicates the arm a, say the replicated arm a'. For cases (i) and (iv), the replicated arm has no effect since all the arms have the same mean rewards, *i.e.,* pulling any arm yields the same expected reward, so let's focus on case (ii) and (iii). In case (ii), the resulting bandit instance from the agent's strategy will be $\mathcal{I}_A : (\mu_a, \mu_{a'}, \mu_b) = (1, 1, 0)$ whereas in case (iii), it will be $\mathcal{I}_B : (0, 0, 1)$ Hence, compared to the strategy with no replication, the agent's gain from replicating arm a is equal to the gain from the average of expected payoff in case (ii) and (iii), compared to the case in which the algorithm runs with bandit instance $\mathcal{I}_O : (1, 0)$ that does not have any replicated arm. Hence, if the agent has no incentive to replicate arm a once, the expected regret under $\mathcal{I}_O$ should be at most the average of that under $\mathcal{I}_A$ and $\mathcal{I}_B$.

Even under simplistic and standard algorithms such as UCB, it is not straightforward to answer the above question. Intuitively, instance $\mathcal{I}_A$ will incur smaller regret as the number of optimal arm increases, but the instance $\mathcal{I}_B$ will have larger regret due to more suboptimal arms with mean reward 0.[3] Thus, the question is whether the loss from $\mathcal{I}_B$ is larger than the gain from $\mathcal{I}_A$ in average, compared to $\mathcal{I}_O$.

More formally, we can refine the above observation as the following fundamental question in comparing the expected regret of multiple bandit instances.

> *Given a bandit instance, consider an adversary who uniformly randomly selects an arm and replicates it. Which bandit algorithms guarantee that the regret does not increase against the adversary?*

We affirmatively answer this question by proving that, there exists an algorithm that satisfies the above questions for *any* bandit instance under some mild assumptions. To this end, we introduce a notion of *random-permutation regret*, which captures the expected regret of an agent's strategy with replication without observing the ex-post realization. Then, we

[1] We remark that our notion of Bayesian differs from that of Bayesian mechanism in the literature. While the Bayesian mechanism computes the equilibrium by considering ex-interim payoff of the players in the incomplete information setting, *i.e.,* ex-interim incentive-compatible, our mechanism is DSIC in an ex-post manner..

[2] We refer to Appendix B for more discussion on the practical motivation of our problem setup.

[3] This can be shown using a coupling-like argument between random reward tapes of the arms, combined with a regret decomposition lemma. See (Shin, Lee, and Ok 2022) for more details.

show that if an algorithm satisfies (i) *truthfulness under random permutation* (TRP) such that the random-permutation regret does not increase with respect to replication, and (ii) permutation invariance such that the algorithm's choice does not depend on the arm indices, then the algorithm is replication-proof. More surprisingly, we prove that these conditions are indeed *necessary* for any algorithm to be replication-proof.[4] In fact, the failure of UCB comes from violating TRP. Meanwhile, we prove that exploration-then-commit (henceforth ETC) satisfies these properties, thus it is replication-proof in the single-agent setting.

For the multi-agent setting, we combine several techniques upon the observations above. First, we borrow the hierarchical structure of (Shin, Lee, and Ok 2022), and analyze hierarchical algorithm with each phase running ETC, which effectively decomposes each agent's payoff from each other, thanks to its non-adaptive nature. It remains as a major open problem whether it is necessary for the algorithm to have non-adaptive structure in order to satisfy TRP. In addition, we introduce a novel restarting round to re-initialize the intra-agent statistics. This largely simplifies the analysis of the selected agent in the exploitation rounds in the first phase, since the second phase simply reduces to the single-agent case with the highest empirical average reward. We cement our result by proving that our algorithm has sublinear regret.

2 Model

We consider a simultaneous game between a set of agents $[n] = \{1, 2, \dots, n\}$ and a principal. Each agent $i \in \mathcal{N}$ has a set of original arms $\mathcal{O}_i = \{o_{i,1}, \dots, o_{i,l_i}\}$ given a constant $l_i \in \mathbb{N}$. Each agent i is equipped with a cumulative distribution F_i from which each original arm $o_{i,k}$'s mean reward $\mu(o_{i,k})$ is sampled for $k \in [l_i]$. Each arm $o_{i,k}$ further follows a reward distribution $G_{i,k}$ with the mean reward sampled as noted above. We use f_i and $g_{i,k}$ to denote the density function of F_i and $G_{i,k}$,[5] and we assume that all the distributions are supported on $[0, 1]$. We further assume that F_i is independent from F_j for $i \neq j \in [n]$. For instance, F_i might be a uniform distribution over $[0, 1]$, and each arm $a \in \mathcal{O}_i$ has a Bernoulli reward distribution with the mean sampled from F_i. We often deal with a *single agent* setting with $n = 1$.

Bayesian agents. Importantly, we consider a *Bayesian scenario* in which each agent is only aware of its own distribution F_i,[6] but not the exact realization of μ_a for $a \in \mathcal{O}_i$. We note that such assumption is indeed practical since content providers (agents) in contents platform typically only have a partial information on how much their contents may attract the users, *e.g.,* based on his/her previous contents and outcomes therein. This is in stark contrast to the setting of (Shin, Lee, and Ok 2022) in which they assume *fully-informed* agents such that any strategic agent does not have any incentive to register the arms except the best one reducing the problem into the single-arm case, which is not practical in many real-world scenarios. We highlight that the equilibrium analysis in the fully-informed setting simply boils down to the case under which each agent only has a single original arm and decides how much to replicate that unique arm, whereas our extension significantly complicates the analysis of equilibrium, as will be elaborated shortly.

Agent's strategy. Given the prior distribution F_i, the set of original $O_{i,k}$ and the principal's algorithm $\mathfrak{A}$, each agent i decides how many times to replicate arm k, including 0 of no replication, for each $k \in [l_i]$. Precisely, agent i commits to a set $\mathcal{S}_i = \{s_{i,k}^{(c)} : c \in [c_{i,k}], k \in [l_i]\}$ in a strategic manner, where $c_{i,k}$ denotes the total number of replica for arm $o_{i,k}$. The replicated arm $s_{i,k}^{(c)}$ has the same mean reward to the original arm, *i.e.,* $\mu(s_{i,k}^{(c)}) = \mu(o_{i,k})$ for any $i \in [n]$ and $k \in [l_i]$. We often use $s_{i,k}^{(0)}$ to denote $o_{i,k}$ for simplicity. Define $\mathcal{S} = (\mathcal{S}_1, \dots, \mathcal{S}_n)$ and $\mathcal{O} = (\mathcal{O}_1, \dots, \mathcal{O}_n)$. Given all the agents' strategies $\mathcal{S}$, the bandit algorithm $\mathfrak{A}$ runs and corresponding rewards are realized.

Mechainsm procedure. The overall mechanism proceeds as follows: (i) The principal commits to a bandit algorithm (mechanism), (ii) Each agent $i \in \mathcal{N}$ decides an action $\mathcal{S}_i$ to register given $\mathfrak{A}$ to maximize own payoff, (iii) The mean rewards μ_a for each $a \in \mathcal{O}$ are sampled, (iv) The bandit algorithm $\mathfrak{A}$ runs and rewards are realized, (v) The principal and the agent realize their own utility.[7]

Agent's payoff. Once an arm is selected by $\mathfrak{A}$, a fixed portion $\alpha \in (0, 1)$ of the reward is paid to the agent who registers the arm. We write $\vec{\mu}_i$ to denote $(\mu_{i,1}, \mu_{i,2}, \dots, \mu_{i,l_i})$, *i.e.,* the vector of agent i's (realized) mean rewards of the arms. Let $E_i(\vec{\mu}_i)$ be an event that the mean reward of agent i's arms are realized to be $\vec{\mu}_i$. Recall that $E_i(\cdot)$ are mutually independent for for $i \in [n]$. We often abuse E_i to denote $E_i(\vec{\mu}_i)$ if the realization $\vec{\mu}_i$ is clear from the context. We also write $\mathcal{S}_{-i}$ to denote $(\mathcal{S}_1, \dots, \mathcal{S}_{i-1}, \mathcal{S}_{i+1}, \dots, \mathcal{S}_n)$, and $E(\vec{\mu})$ for $\vec{\mu} = (\vec{\mu}_1, \vec{\mu}_2, \dots, \vec{\mu}_n)$ to denote the event such that $\cap_{i \in [n]} E_i(\vec{\mu}_i)$.

We further introduce some notations given the principal's algorithm $\mathfrak{A}$. Conditioned on the $E_i(\vec{\mu}_i)$, agent i's *ex-post payoff* of selecting strategy $\mathcal{S}_i$ given the others' strategies is defined as

$$U_i(\mathcal{S}_i, \mathcal{S}_{-i}; E(\vec{\mu})) = \alpha \sum_{t=1}^{T} R_t 1\left\{ I^{(t)} \in \mathcal{S}_i \right\},$$

where R_t refers to the reward at round t and I_t denotes the arm selected at round t.

Given the ex-post utility, each agent computes its *ex-ante payoff* by marginalizing over the realization of its own arms' mean rewards, defined as follows

$$u_i(\mathcal{S}_i, \mathcal{S}_{-i}) = \mathbb{E}\left[U_i(\mathcal{S}_i, \mathcal{S}_{-i}; E(\vec{\mu})) \right],$$

where the randomness in the expectation comes from $E(\vec{\mu})$, i.e. the realization of the mean rewards, and possibly from the random bits of the algorithm $\mathfrak{A}$. In words, each agent

[4] We refer to Section 2 for precise definitions.

[5] We assume the existence of density function for ease of exposition, but all the results will easily be generalized.

[6] Discussion on the generalization to the asymmetric setting is presented in Appendix C.

[7] Note that if all the agents do not strategize and simply register $\mathcal{O}_i$ in step (ii), our problem reduces to the standard stochastic multi-armed bandit problem.

is rational *expected utility maximizer* who maximizes its expected payoff.

Information and desiderata. We now describe the information structure of the mechanism, and how each agent decides its own strategy based on its ex-ante payoff. We first assume that the principal does not have any information regarding the agents' distributions nor the number of original arms, but can observe by whom each arm is registered. This is often called *prior-independent mechanism* in the mechanism design, *c.f.,* (Hartline et al. 2013). In addition, we assume that the agent does not have any information regarding the others. Instead, our objective is to construct a *incentive-compatible* mechanism such that each agent's *dominant strategy* is to register the arms without any replication, without regard to what others may have proposed. To this end, we first define a dominant strategy as follows.

Definition 1 (Dominant Strategy)**.** Agent i's strategy $\mathcal{S}_i$ is dominant strategy if $u_i(\mathcal{S}_i \cup \mathcal{S}_{-i}) \geq u_i(\mathcal{S}'_i \cup \mathcal{S}_{-i})$ for any $\mathcal{S}'_i$ and $\mathcal{S}_{-i}$.

Correspondingly, we introduce the following notion of *dominant strategy equilibrium*.

Definition 2 (Equilibrium)**.** An algorithm $\mathfrak{A}$ has dominant strategy equilibrium (DSE) if there exists a set of agent strategies $\mathcal{S} = \{\mathcal{S}_i\}_{i\in[n]}$ such that $\mathcal{S}_i$ is dominant strategy for any agent $i \in [n]$.

As per the definition, the total ordering of the agent's payoff with respect to its own strategy should remain the same regardless of the others' strategies $\mathcal{S}_{-i}$. Thus, we often omit the parameter $\mathcal{S}_{-i}$ in denoting agent i's payoff, *e.g.,* $u_i(\mathcal{S}_i)$ instead of $u_i(\mathcal{S}_i, \mathcal{S}_{-i})$. Note, however, that the realization of the other agents' mean rewards may affect the actual quantity of $u_i(\mathcal{S}_i)$, although it does not affect which strategy agent i will eventually commit to. Since we are interested only in how each agent will "strategize", it does not lose any generality.

We finally introduce the following notion of truthfulness.

Definition 3 (Replication-proof)**.** An algorithm $\mathfrak{A}$ is *replication-proof* if it has a dominant strategy equilibrium $\{\mathcal{O}_i\}_{i\in[n]}$ for any $T \in \mathbb{N}$.

Replication-proofness is desirable as the agents are not motivated to replicate their own arms, thereby reducing the number of arms that a bandit algorithm faces. Since the regret upper bound of a standard bandit algorithm usually depends on the number of arms, maintaining a smaller number of suboptimal arms in the system affects the eventual learning efficiency of the algorithm. We often say that an algorithm is *incentive-compatible* or *truthful* to denote the replication-proofness (Nisan et al. 2007).

On top of designing *prior-independent incentive-compatible* mechanism, we are also interested in a learning efficiency, *i.e.,* learning the reward distribution in a sample-efficient manner. In the literature, this is usually described as cumulative regret, defined as follows.

Definition 4 (Regret)**.** Given time horizon T, strategy $\mathcal{S}$ and the event $E(\vec{\mu})$, the ex-post regret of algorithm $\mathfrak{A}$ is

$$\textsc{Reg}(\mathfrak{A}, T; \mathcal{S}, E(\vec{\mu})) = \left(\max_{a\in\cup_{i\in\mathcal{N}}\mathcal{O}_i} \sum_{t=1}^{T} \mu(a)\right) - \sum_{t=1}^{T} \mu(I_t),$$

where I_t denotes the arm selected by $\mathfrak{A}$ at round t.

We emphasize that the optimal benchmark is selected from $\cup_{i\in\mathcal{N}}\mathcal{O}_i$, not from the registered set of arms $\cup_{i\in\mathcal{S}}\mathcal{S}_i$. This benchmark is indeed important since the principal should incentivize the agents to decide an action that contains own best arm to maintain larger utility. For example, if an algorithm motivates an agent to not register all the arms, then there might be a constant probability that the omitted arm has the largest mean reward, which essentially yields sublinear regret with respect to the regret defined above. More formally, our notion of regret implicitly implies that any sublinear regret algorithm incentivizes all the agents to register all their original arms at least once. We also remark that although the arms' mean rewards are sampled from prior distributions, which is often referred to Bayesian bandits (*e.g.,* see Ch.3 of (Slivkins et al. 2019)), we deal with ex-post regret, *e.g.,* as does in (Agrawal and Goyal 2012) which analyzes the ex-post regret of Thompson Sampling in Bayesian bandits. Obviously, ex-post regret upper bound *implies* Bayesian regret upper bound by simply taking an expectation.

3 Failure of Existing Algorithms

Before getting into our main results, the natural question one might ask is to analyze whether the existing algorithms by (Shin, Lee, and Ok 2022) work. They mainly propose hierarchical UCB (H-UCB), which is a variant of the standard UCB by (Auer, Cesa-Bianchi, and Fischer 2002). The formal pseudocode of both UCB and H-UCB are deferred to the appendix. While we provide the analysis with specific exploration parameter of $\ln t/n_a$, we note that the result easily carries over to arbitrary exploration parameters $f(t, n_a)$, once f is increasing over t and decreasing over n_a. We omit further details as it is beyond our interest.

Mainly, H-UCB consists of two phases: (i) in the first phase, it runs UCB1 by considering each agent as a single arm and selects one agent at each round, and (ii) in the second phase, it runs UCB within the selected agent. Precisely, it maintains two types of statistics: (i) agent-wise empirical average reward $\hat{\mu}_i$, number of pulls per agent n_i, and (ii) arm-wise empirical average reward $\hat{\mu}_{i,a}$ and number of pulls per arm $n_{i,a}$.

Theorem 1. *There exists a problem instance such that UCB1 is not replication-proof in the single-agent setting, and such that H-UCB is not replication-proof for any number of agents.*

The proof for the single-agent setting relies on a construction of bad problem instance which exploits the deterministic and adaptively exploring nature UCB. To briefly explain the proof, we consider an agent with two Bernoulli arms whose prior distribution is supported on $\{0, 1\}$, where it samples mean 0 and 1 equally likely. Let's focus on a realization in which one arm has mean 0, and the other has 1. Observe that replicating one arm essentially yields one of the following two bandit instances: (i) arms with mean rewards $(1, 1, 0)$ or (ii) arms with mean rewards $(1, 0, 0)$. Here, we write (x, y, z) to denote bandit instance with three arms mean rewards $1, 1, 0$. Note that, intuitively, (i) yields a lower expected regret, but higher in (ii), compared to the original instance with $(0, 1)$. Thus, our analysis mainly reduces to

show that the loss from (ii) is larger than the gains from (i). This can be made indeed true since after observing the dynamics of instance (ii), we can essentially set the time horizon to be exactly right *before* the two arms with mean rewards zero are *consecutively* chosen. In this way, the loss from (ii) can be made smaller while the gains from (i) is moderately large, which implies the dominance of replicating strategy. The proof for the multi-agent setting essentially is based on the fact that its Phase 2 does not guarantee replication-proofness within the chosen agent due to the single-agent argument, along with carefully chosen problem parameters to propagate this phenomenon to Phase 1 so as to increase the agent's expected utility under replication.

4 Warm-up: Single-agent Setting

As noted thus far, UCB is not replication-proof even in the single agent setting. Indeed, as per Theorem 1, it is not obvious to verify which algorithm would satisfy replication-proofness even in the single-agent setting. In this section, we step towards in constructing replication-proof algorithm, by carefully investigating the single-agent setting first. We present a set of conditions which is sufficient for an algorithm to be replication-proof in the single-agent setting, and then, as an example, will prove that exploration-then-commit (ETC) will satisfy these properties. As we restrict our attention to single-agent setting in this section, we often omit the agent index i in the notations.

To this end, we first introduce several notations. For any natural number l, let $\mathcal{P}_l$ be a set of all possible permutations $\sigma : [l] \mapsto [l]$. Now we introduce a dictionary form of bandit instance $\mathcal{I}$. Given the realization of the arms' mean rewards, suppose that the set of mean rewards constitute a sorted sequence $\{1 \geq \mu_1 > \mu_2 > \ldots > \mu_l > 0\}$, where some arms may share the same mean reward. For each μ_a for $a \in [l]$, we count the number of arms which have mean reward μ_a, and denote the count by c_a. Then, our bandit instance $\mathcal{I}$ can essentially presented as tuples of the mappings $\bigtimes_{a\in[l]}(\mu_a : c_a)$, which we say dictionary form of the bandit instance $\mathcal{I}$.

Suppose that we have a standard multi-armed bandit instance $\mathcal{I}$ with dictionary form $(\mu_a : c_a)_{a\in[l]}$. Then, we define a *permuted bandit instance* $\mathcal{I}_\sigma$ to be a bandit instance with dictionary form of $\bigtimes_{a\in[l]}(\mu_a : c_{\sigma(a)})$, *i.e.,* arms with mean reward μ_i appears $c_{\sigma(a)}$ times. For instance, given the bandit instance $\mathcal{I} = (0.5 : 1, 0.7 : 2)$, consider a permutation σ such that $\sigma(1) = 2$ and $\sigma(2) = 1$ correspondingly. Then, one can easily verify that $\mathcal{I}_\sigma = (0.5 : 2, 0.7 : 1)$.

Now, we introduce the random permutation regret.

Definition 5 (Random permutation regret)**.** Given a single-agent bandit instance $\mathcal{I}$ with l arms, we define an algorithm $\mathfrak{A}$'s *random permutation regret* (RP-Regret) as follows.

$$\begin{aligned}\text{RP-}\textsc{Reg}(\mathfrak{A}, T) &= \mathbb{E}_{\sigma\in\mathcal{P}_l}\left[\textsc{Reg}_{I_\sigma}(\mathfrak{A}, T)\right] \\ &= \frac{1}{|\mathcal{P}_l|}\cdot\left(\sum_{\sigma\in\mathcal{P}_l}\textsc{Reg}_{I_\sigma}(\mathfrak{A}, T)\right). \qquad (1)\end{aligned}$$

Given the random permutation regret, we define the following property of algorithm, which will play an essential role in obtaining truthful algorithm in the single-agent setting.

Definition 6 (Truthful under random permutation)**.** Given an arbitrary single-agent bandit instance $\mathcal{I}$ with l arms, consider a truthful strategy $\mathcal{O}$ and arbitrary strategy $\mathcal{S}$. An algorithm $\mathfrak{A}$ is *truthful under random permutation* (TRP) if

$$\text{RP-}\textsc{Reg}(\mathfrak{A}, T; \mathcal{O}) \leq \text{RP-}\textsc{Reg}(\mathfrak{A}, T; \mathcal{S}).$$

Note that TRP requires the inequality to holds for arbitrary bandit instance $\mathcal{I}$. The foundation of these notions essentially builds upon the observation from our counterexample provided in Theorem 1. To elaborate more, consider a strategy $\mathcal{S}$ that only replicates the first arm given a bandit instance $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_l$. Then, we can observe that any permutation $\sigma \in \mathcal{P}_l$ results in a bandit instance that belongs to the following family of bandit instances: define $\mathcal{I}^* = \cup_{i\in[l]}\mathcal{I}_i$ such that

$$\mathcal{I}_1 = \{\underbrace{\mu_1, \mu_1}_{repl}, \mu_2, \ldots, \mu_l\}, \ldots, \mathcal{I}_l = \{\mu_1, \mu_2, \ldots, \underbrace{\mu_l, \mu_l}_{repl}\}.$$

In words, $\mathcal{I}_i$ denotes the case that the strategy $\mathcal{S}$ replicates the arm with i-th highest reward. For example, suppose that $\sigma \in \mathcal{P}_l$ satisfies $\sigma(1) = i$ for some $i \in [l]$. Since $\mathcal{S}$ replicates the first arm, under the permutation σ, the arm with parameter μ_i is replicated, and thus it corresponds to $\mathcal{I}_i$. Then, our notion of TRP essentialy asks what is an algorithm that makes the expected regret of $\mathcal{I}$ smaller than (or equal to) the average of expected regret of $\mathcal{I}_1, \ldots, \mathcal{I}_l$. Put it differently, TRP essentially asks the following fundamental question, as we pointed out in the introduction.

Given a bandit instance, does "uniform randomly" replicate one arm increase the expected regret?

This foundation of TRP will play an essential role in constructing replication-proof algorithm in both the single-agent and the multi-agent setting. Indeed, from our proof of Theorem 1, one can observe that UCB does not satisfy the above question, and thus fails to be replication-proof.

Definition 7 (Permutation invariance)**.** An algorithm is *permutation invariant* (PI) if given the random bits of the algorithm and the arms, the choice of which arms to pull remains the same (up to tie-breaking) when the index of the arms are permuted.

Note that the permutation-invariance essentially requires the algorithm to be agnostic to the index of the arms, *i.e.,* does not use the arm index to choose the arm, possibly except the tie-breaking case. One can easily verify that it holds for a broad class canonical algorithms such as UCB and ε-greedy.

Theorem 2. *In the single-agent setting, if an algorithm is TRP and PI, then it is replication-proof.*

The proof essentially follows from the observation that the algorithm's ex-ante regret can be partitioned into a disjoint set of random permutation regret, with respect to the ex-post realization of the mean rewards. Based on this disjoint partition, we can effectively marginalize over the priors while maintaining the partition, and conclude that the dominance over the partition yields the overall dominance.

Furthermore, we observe that TRP is indeed a necessary condition for an algorithm to be replication-proof in the single-agent setting.

Proposition 1. *If an algorithm is not TRP, then it is not replication-proof.*

Discussion 1. One may wonder if PI can be violated for any replication-proof algorithm. For example, one can consider a black-box algorithm that first randomly selects an agent without using any statistical information but only with the agent indices, and runs a bandit algorithm within the agent. Further, suppose that it asymmetrically favors some agents' indices with higher probability in the first phase. Such an algorithm indeed is not permutation invariant, since the permutation of agent index would lead to a different outcome given the same reward tapes. From agents' perspective, however, each agent only needs to care about the secondary arm selection phase after the agent selection since they cannot change the probability to be selected in the agent selection phase. Using these observations, one can observe that such an algorithm is indeed replication-proof, but not permutation invariant. One may refine the notion of PI to be restricted within each agent, but we do not argue more details as it is beyond of our interest.

Constructing TRP algorithm. We now present an algorithm that satisfies TRP. In what follows, we mainly prove that exploration-then-commit (ETC) satisfies TRP once equipped with a proper parameter. Its pseudocode is presented in Algorithm 3 in the appendix. We suppose that the algorithm breaks tie in a uniform random manner.[8] ETC essentially decouples the exploration phase and the exploitation phase very explicitly, and thus brings a significant advantage in comparing its expected regret under several problem instances.

For analytical tractability, we pose some *assumptions* on the support of the prior distributions. Namely, for any agent i, F_i has a discrete support over $[0, 1]$ for $i \in \mathcal{N}$. Further, we define Δ_i be the minimum gap between any two possible outcomes from F_i, and let $\Delta = \min_{i\in\mathcal{N}} \Delta$. Furthermore, the algorithm knows this gap Δ.

Discussion 2. We discuss how one can weaken these assumptions at the end of this section, possibly at the cost of polylogarithmic blowup in the regret rate or another assumptions. We also remark that such assumptions, especially that the algorithm knows the minimum gap, often appear in the literature, *c.f.,* (Auer, Cesa-Bianchi, and Fischer 2002; Audibert, Bubeck, and Munos 2010; Garivier, Lattimore, and Kaufmann 2016).

Discussion 3. In practice, such scenario is fairly plausible in real-world applications. For example in content platforms, suppose that a content creator comes up with several contents and is about to register. Assume the reward of each content is just the number of views. Typically, the content creator does not exactly know the reward of each content but has historical data on its previous contents. The historical data can be used to structure a prior distribution of the quality of his/her contents. Since each content creator usually has a very small number of original contents compared to the recommendation algorithm's time scale (one for each traffic), this prior distribution may consist of a fairly limited number of samples, thereby inducing a discrete support with a few number of points.

Then, our main result in the single-agent setting can be written as follows.

Theorem 3. *ETC with exploration length $m \geq \frac{2}{\Delta^2} l \ln(2T)$ is TRP in the single-agent setting. Further, it has regret upper bound of*

$$\sum_{a\in[l]} \Big(\frac{2\delta_a l \ln(2T)}{\Delta^2} + 1 \Big),$$

where l denotes the number of arms and δ_a denotes the gap of the mean rewards between optimal arm and arm a for the single agent.

The proof essentially follows from the standard regret analysis of ETC along with carefully chosen m. By setting m sufficiently large, we can essentially decrease the probability that a suboptimal arm is chosen to be $o(1/T)$. This will guarantee that the expected number of rounds the suboptimal arm is chosen is smaller than 1, whereas any replicating strategy incurs a simple regret lower bound larger than this quantity. Together with Theorem 2 and the fact that ETC is PI (which is straightforward to verify), it follows that ETC is replication-proof in the single-agent setting.

Note that the proof of Theorem 3 heavily relies on the fact that the algorithm can observe Δ in advance and set the exploration length correspondingly. Thus, despite we aim to obtain a *prior-independent* algorithm that does not require any knowledge in the agent's prior distributions, it is not *truly* prior-independent in an algorithmic manner since it requires some problem parameters to operate the algorithm. We discuss several ways to refine such restrictions, thereby suggesting a road for truly prior-independent algorithm. First, assume Δ is constant but not known to the algorithm. In this case, one may replace Δ in the algorithm to be some increasing functions $f(T) = \omega(1)$, and it is straightforward to verify that Theorem 3 still holds, but in an asymptotical regime. Similarly, one can effectively wipe out the dependency on the total number of original arms l by assuming that $l = O(1)$ and replacing it with some increasing functions. Furthermore, the knowledge on T can be wiped out by the standard doubling trick, *c.f.,* see Chapter 1 in (Slivkins et al. 2019), without sacrificing the truthfulness. Remark that all these refinements only introduce polylogarithmic blowup in the regret upper bound, where we do not provide a formal proof as it is a cumbersome application of standard techniques. The following theorem spells out these arguments.

Proposition 2. *Assume $l = O(1), \Delta = \Omega(1)$. Then, there is a prior-independent algorithm with polylogarithmic regret satisfying TRP and PI in the single-agent setting for sufficiently large T, that does not require any information on problem parameters.*

5 Multi-agent Replication-proof Algorithm

Now we turn our attention to the more general setting with multiple agents. We start by using the machinery of *hierarchical structure* in H-UCB, but in a different manner. A direct implication of Theorem 2 is that if we run a variant

[8] Any tie-breaking rule does not hurt our analysis.

of H-UCB such that the first phase runs a simple uniform-random selection algorithm, and the second phase runs some arbitrary TRP and PI algorithms, then the resulting algorithm is replication-proof. To formally see this why, let $\mathcal{S}_i$ be the strategy of agent i. Define $\Gamma_i(\mathcal{S}_i)$ be the agent i's utility under the single-agent setting with agent i, *i.e.,* when there are no other agents other than i. Due to the uniformly random selection nature in the first phase, the expected utility of any agent i can be simply written $u_i(\mathcal{S}_i, \mathcal{S}_{-i}) = \Gamma_i(\mathcal{S}_i)/n$. Note that the dynamic of the algorithm purely reduces to the dynamic of the second phase algorithm if there's only a single agent. Hence, by Theorem 2 and due to the TRP and PI of the second phase algorithm, we conclude that truthful registration is a dominant strategy for any agent i, and thus the result follows. Note, however, that if we deploy uniform random selection algorithm in ALG_1, it essentially suffers *linear regret* since it always choose any suboptimal agent with constant probability at every rounds.

To capture both of sublinear regret and replication-proofness, we need more sophisticated algorithm with sublinear regret in the first phase. To this end, we present H-ETC-R presented in Algorithm 5 in the appendix, which adopts ETC in both phases along with additional device of *restarting round*. Similar to H-UCB, this consists of two phases each running ETC agent-wise and arm-wise manner, respectively. For analytical tractability, we say that the set of arms belong to *stochastically ordered family* if two arms a and b satisfy $\mu_a \geq \mu_b$, then the reward distribution of arm a is (first-order) stochastic dominant over that of b, *i.e.,* $\mathbb{P}[r_a \geq x] \geq \mathbb{P}[r_b \geq x]$ for any $x \geq 0$, where r_i denotes arm i's reward random variable.[9] Further, define $L = \max_{i \in [n]} l_i$. Then, our main result is as follows.

Theorem 4. *Consider a stochastically ordered family of arms, and discrete support of prior distributions. Consider* H-ETC-R *with* $M \geq mL$ *and* $m = \frac{2}{\Delta^2} L \ln(2T)$*, and* $\tau = Mn$ *Then,* H-ETC-R *is replication-proof.*[10]

Proof sketch. We provide a brief proof sketch. The proof mainly relies on the single-agent result in Theorem 2 along with a number of algebraic manipulations which heavily relies on the nature of ETC and the restarting round. Let us focus on agent i. We can decompose agent i's total expected utility to be (i) exploitation phase utility and (ii) exploration phase utility. Given a realization of the reward tapes, the utility from (i) and (ii) is independent due to the nature of ETC. Since (i) simply reduces to running single-agent ETC for each agent, we can compare (i) of two strategies using the single-agent result, thus truthful strategy is at least better for (i) thanks to the single-agent theorem.

For (ii), if agent i is not selected in the exploitation phase, it is simply zero, so we can only focus on the case when agent i is selected in the exploitation phase. Thus, its expected utility from (ii) can be written as multiplication of (a) probability that agent i has the largest empirical average reward and (b) the expected utility by being pulled over the exploitation rounds. For (b), since our choice of τ *restarts* Phase 2 for any agent exactly at the beginning of exploitation phase of Phase 1, the expected utility therein simply reduces to the expected utility of single-agent setting with agent i given the time horizon $T - Mn$. Thus, again by our single-agent result of Theorem 2, the truthful strategy yields larger (or equal) quantity for (b) as well. Finally, by carefully using a series of concentration inequalities, we bound the difference from (a) between the truthful strategy and any other strategy is relatively smaller and by doing so, loss from (a) cannot make up the gains from (b) and (i), which eventually implies the dominance of truthful strategy. □

We finalize our results by presenting regret upper bound of the H-ETC-R with ETC and ETC, which concludes that it achieves sublinear regret as well as replication-proofness.

Theorem 5. *Set* ALG_1 *be ETC with* $M = \max(mL, \sqrt{T \ln T})$*, and* ALG_2 *be ETC with* $m = 2/\Delta^2 \cdot L \ln(2T)$*. Then,* H-ETC-R *has expected regret of* $O(\frac{nL^3\sqrt{T \ln T}}{\Delta^3})$.

The regret analysis is based on a construction of clean event on each agent under which the agent's empirical average reward becomes close enough to his own optimal arm's mean reward. The deviation probability of this clean event can be obtained using tail bounds along with the regret analysis of ETC, which essentially implies the necessity of having ALG_1's exploration length polynomial. The regret bound then easily follows from obtaining the probability that all the clean event holds. The formal proof is deferred to appendix.[11]

6 Conclusion

We study bandit mechanism design problem to disincentivize replication of arms, which is the first to study Bayesian extension of (Shin, Lee, and Ok 2022). Our extension brings significant challenges in analyzing equilibrium, even in the single-agent setting. We first prove that H-UCB of (Shin, Lee, and Ok 2022) does not work. We then characterize sufficient conditions for an algorithm to be replication-proof in the single-agent setting. Based on the single-agent result, we obtain the existence of replication-proof algorithm with sublinear regret, by introducing a restarting round and exploiting the structural property of exploration-then-commit algorithm. We further provide a regret analysis of our replication-proof algorithm, which matches the regret of H-UCB.

Acknowledgement

This work is partially supported by DARPA QuICC, ONR MURI 2024 award on Algorithms, Learning, and Game Theory, Army-Research Laboratory (ARL) grant W911NF2410052, NSF AF:Small grants 2218678, 2114269, 2347322.

[9] For example, Bernoulli arms and Gaussian arms parameterized by mean rewards, belongs to this family. We also refer to (Yu 2009) for more examples in exponential family.

[10] Similar to the discussion in Proposition 2, we can refine this results and make the algorithm truly prior-independent. We omit these details of beyond the interest.

[11] We remark that the algorithm's dependency on the problem parameters can be weakened similar to Proposition 2, but we omit the details as it goes beyond our interest.

References

Agrawal, S.; and Goyal, N. 2012. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, 39–1. JMLR Workshop and Conference Proceedings.

Audibert, J.-Y.; Bubeck, S.; and Munos, R. 2010. Best arm identification in multi-armed bandits. In *COLT*, 41–53.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47: 235–256.

Badanidiyuru, A.; Kleinberg, R.; and Slivkins, A. 2018. Bandits with knapsacks. *Journal of the ACM (JACM)*, 65(3): 1–55.

Badanidiyuru, A.; Langford, J.; and Slivkins, A. 2014. Resourceful contextual bandits. In *Conference on Learning Theory*, 1109–1134. PMLR.

Barraza-Urbina, A.; and Glowacka, D. 2020. Introduction to bandits in recommender systems. In *Proceedings of the 14th ACM Conference on Recommender Systems*, 748–750.

Braverman, M.; Mao, J.; Schneider, J.; and Weinberg, S. M. 2019. Multi-armed bandit problems with strategic arms. In *Conference on Learning Theory*, 383–416. PMLR.

Djallel, B.; and Irina, R. 2019. A survey on practical applications of multi-armed and contextual bandits. *arXiv preprint arXiv:1904.10040*.

Esmaeili, S.; Hajiaghayi, M.; and Shin, S. 2023. Replication-proof Bandit Mechanism Design. *arXiv preprint arXiv:2312.16896*.

Feng, Z.; Parkes, D.; and Xu, H. 2020. The intrinsic robustness of stochastic bandits to strategic manipulation. In *International Conference on Machine Learning*, 3092–3101. PMLR.

Garivier, A.; Lattimore, T.; and Kaufmann, E. 2016. On explore-then-commit strategies. *Advances in Neural Information Processing Systems*, 29.

Hartline, J. D.; et al. 2013. Bayesian mechanism design. *Foundations and Trends® in Theoretical Computer Science*, 8(3): 143–263.

Liu, L. T.; Mania, H.; and Jordan, M. 2020. Competing bandits in matching markets. In *International Conference on Artificial Intelligence and Statistics*, 1618–1628. PMLR.

Liu, L. T.; Ruan, F.; Mania, H.; and Jordan, M. I. 2021. Bandit learning in decentralized matching markets. *The Journal of Machine Learning Research*, 22(1): 9612–9645.

Nisan, N.; Roughgarden, T.; Tardos, E.; and Vazirani, V. V. 2007. *Algorithmic game theory*. Cambridge university press.

Shin, S.; Lee, S.; and Ok, J. 2022. Multi-armed Bandit Algorithm against Strategic Replication. In *International Conference on Artificial Intelligence and Statistics*, 403–431. PMLR.

Slivkins, A.; et al. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2): 1–286.

Tang, L.; Jiang, Y.; Li, L.; and Li, T. 2014. Ensemble contextual bandits for personalized recommendation. In *Proceedings of the 8th ACM Conference on Recommender Systems*, 73–80.

Yu, Y. 2009. Stochastic ordering of exponential family distributions and their mixturesxk. *Journal of Applied Probability*, 46(1): 244–254.