ARTIFICIAL REPLAY: A META-ALGORITHM FOR HARNESSING HISTORICAL DATA IN BANDITS

Anonymous authors
Paper under double-blind review

ABSTRACT

While standard bandit algorithms sometimes incur high regret, their performance can be greatly improved by "warm starting" with historical data. Unfortunately, how best to incorporate historical data is unclear: naively initializing reward estimates using all historical samples can suffer from *spurious data* and *imbalanced data coverage*, leading to computational and storage issues—particularly in continuous action spaces. We address these two challenges by proposing ARTIFICIAL REPLAY, a meta-algorithm for incorporating historical data into *any arbitrary base bandit algorithm*. ARTIFICIAL REPLAY uses only a subset of the historical data *as needed* to reduce computation and storage. We provide guarantees that our method achieves equal regret as a full warm-start approach while potentially using only a fraction of the historical data for a broad class of base algorithms that satisfy *independence of irrelevant data* (IIData), a novel property that we introduce. We complement these theoretical results with a case study of K-armed and continuous combinatorial bandit algorithms, including on a green security domain using real poaching data, to show the practical benefits of ARTIFICIAL REPLAY in achieving optimal regret alongside low computational and storage costs. Across these experiments, we show that ARTIFICIAL REPLAY performs well for all settings that we consider, even for base algorithms that do not satisfy IIData.

1 INTRODUCTION

Multi-armed bandits and their variants are robust models for many real-world problems. Resulting algorithms have been applied to wireless networks (Zuo & Joe-Wong, 2021), COVID testing regulations (Bastani et al., 2021), and conservation efforts to protect wildlife from poaching (Xu et al., 2021). Typical bandit algorithms assume no prior knowledge of the expected rewards of each action, simply taking actions online to address the exploration–exploitation trade-off. However, many real-world applications offer access to *historical data*. For example, in the wildlife conservation setting, we may have access to years of historical patrol records that should be incorporated to learn poaching risk before deploying any bandit algorithm.

There is no consensus on how to optimally incorporate this historical data into online learning algorithms. The naive approach uses the full historical dataset to initialize reward estimates (Shivaswamy & Joachims, 2012), possibly incurring unnecessary and onerous computation and storage costs. These costs are particularly salient in continuous action settings with adaptive discretization, where the number of discretized regions is a direct function of the number of historical samples. If excessive data was collected on poor-performing actions, this *spurious data* with *imbalanced data coverage* would lead us to unnecessarily process and store an extremely large number of fine discretizations in low-performing areas of the action space, even when a significantly coarser discretization would be sufficient to inform us that region is not worth exploring. These two key challenges highlight that *the value of information of the historical dataset may not be a direct function of its size*. Real-world decision makers echo this sentiment: Martin et al. (2017) note that for conservation decisions, more information does not always translate into better actions; time is the resource which matters most.

A natural question one can ask is: *Is there an efficient way (in terms of space, computational, and sample complexity) to use historical data to achieve regret-optimal performance?* For example, many real-world applications of bandit algorithms, such as online recommender systems, may con-

tain historical datasets with millions of data points. Processing these millions of points would require an exceptional amount of upfront computation and storage cost, especially if many of those historical points are no longer relevant; many samples may encode out-of-date data such as old movies or discontinued products.

To this end, we propose ARTIFICIAL REPLAY, a meta-algorithm that modifies *any base bandit algorithm* to harness historical data. ARTIFICIAL REPLAY reduces computation and storage costs by only using historical data on an *as needed* basis. The key intuition is if we could *choose* which samples to include in the historical dataset, a natural approach would be to use a regret-optimal bandit algorithm to guide the sampling. ARTIFICIAL REPLAY builds on this intuition by using historical data as a replay buffer to artificially simulate online actions. Every time the base algorithm picks an action, we first check the historical data for any unused samples from the chosen action. If an unused sample exists, update the reward estimates and continue *without* advancing to the next timestep. Otherwise, sample from the environment, update the estimates using the observion, and continue to the next timestep. While this idea is easiest to understand in the context of the standard K-armed bandit problem, we discuss later how this framework naturally extends to other structure and information models, including continuous action spaces with semi-bandit feedback.

Although ARTIFICIAL REPLAY seems to be a natural heuristic to minimize use of historical data, it is not clear how to analyze its regret—specifically how much it loses compared to "full warm-start" (i.e., where the base algorithm is initialized with the full dataset). Surprisingly, however, we prove that under a widely applicable condition, the regret of ARTIFICIAL REPLAY (as a random variable) is distributionally identical to that of a full warm-start approach, while also guaranteeing significantly better time and storage complexity. Specifically, we show a *sample-path coupling*[1] between our ARTIFICIAL REPLAY approach and the full warm start approach with the same base algorithm, as long as the base algorithm satisfies a novel *independence of irrelevant data* (IIData) assumption. While our goal is not to show regret improvements, this result highlights how ARTIFICIAL REPLAY is a simple approach for incorporating historical data with identical regret to full warm start (approach done in practice) with significantly smaller computational overhead.

Finally, we show the practical benefits of our method by instantiating ARTIFICIAL REPLAY for several broad classes of bandits and evaluating on real-world data. To highlight the breadth of algorithms that satisfy the IIData property, we provide examples of regret-optimal IIData polices for K-armed and continuous combinatorial bandits. We use these examples to prove that ARTIFICIAL REPLAY can lead to arbitrary better storage and computational complexity requirements. We close with a case study of combinatorial bandit algorithms for continuous resource allocation in the context of green security domains, using a novel adaptive discretization technique. Across the experiments, we observe concrete gains in storage and runtime using real-world poaching data from the ARTIFICIAL REPLAY framework over a range of base algorithms, including algorithms that do not satisfy IIData such as Thompson sampling and Information Directed Sampling (IDS).

1.1 RELATED WORK

Multi-armed bandit problems have a long history in the online learning literature. We highlight the most closely related works below; for more extensive references please see our detailed discussion in Appendix B and Bubeck et al. (2012); Slivkins (2019); Lattimore & Szepesvári (2020).

Multi-Armed Bandit Algorithms. The design and analysis of bandit algorithms have been considered under a wide range of models. These algorithms were first studied in the K-armed bandit model in Lai & Robbins (1985), where the decision maker has access to a finite set of K possible actions at each timestep. However, numerous follow-up works have considered similar approaches when designing algorithms in continuous action spaces (Kleinberg et al., 2019) and with combinatorial constraints (Chen et al., 2013; Xu et al., 2021; Zuo & Joe-Wong, 2021). Our work provides a framework to modify existing algorithms to harness historical data. Moreover, we also propose a novel algorithm to incorporate adaptive discretization for combinatorial multi-armed bandits for continuous resource allocation, extending the discrete model from Zuo & Joe-Wong (2021).

[1]That is, we construct the regret process for both algorithms simultaneously on a joint probability space such that each individual process has the correct marginal, but both processes are equal over each sample path.

Incorporating Historical Data. Several papers have started to investigate how to incorporate historical data into bandit algorithms, starting with Shivaswamy & Joachims (2012) who consider a K-armed bandit model where each arm has a dataset of historical pulls. The authors develop a "warm start" UCB algorithm to initialize the confidence bound of each arm based on the full historical data—prior to learning. Bouneffouf et al. (2019) extended similar techniques to models with pre-clustered arms. These techniques were later extended to Bayesian and frequentist linear contextual bandits, where the linear feature vector is initialized based on standard regression over the historical data (Oetomo et al., 2021; Wang et al., 2017). Our work provides a contrasting approach to harnessing historical data in algorithm design: our meta-algorithm can be applied to *any* standard bandit framework and uses the historical data only *as needed*, leading to improved computation and storage gains.

2 Preliminaries

We now define the general bandit model and specify the finite-armed and online combinatorial allocation settings that we study in our experiments. See Appendix C for details.

2.1 General Stochastic Bandit Model

We consider a stochastic bandit problem with a fixed action set $\mathcal{A}$. Let $\Re : \mathcal{A} \to \Delta([0,1])$ be a collection of independent and *unknown* reward distributions over $\mathcal{A}$. Our goal is to pick an action $a \in \mathcal{A}$ to maximize $\mathbb{E}[\Re(a)]$, the expected reward, which we denote $\mu(a)$. The optimal reward is:

$$\textsc{Opt} = \max_{a \in \mathcal{A}} \mu(a) \,. \tag{1}$$

For now, we do not impose any additional structure on $\mathcal{A}$, which could potentially be discrete, continuous, or encode combinatorial constraints.

Historical Data. We assume that the algorithm designer has access to a historical dataset $\mathcal{H}^{hist} = \{a_j^{\mathcal{H}}, R_j^{\mathcal{H}}\}_{j \in [H]}$ containing H historical points with actions $\{a_j^{\mathcal{H}}\}_{j \in [H]}$ and rewards $R_j^{\mathcal{H}}$ sampled according to $\Re(a_j^{\mathcal{H}})$. We do not make any assumptions on how the historical actions $a_j^{\mathcal{H}}$ are chosen and view them as deterministic and fixed upfront. Our goal is to efficiently incorporate this historical data to improve the performance of a bandit algorithm.

Online Structure. Since the mean reward function $\mu(a)$ is initially unknown, we consider settings where the algorithm interacts with the environment sequentially over T timesteps. At timestep $t \in [T]$, the decision maker picks an action $A_t \in \mathcal{A}$ according to their policy π. The environment then reveals a reward R_t sampled from the distribution $\Re(A_t)$. The optimal reward $\textsc{Opt}$ would be achieved using a policy with full knowledge of the true distribution. We thus define *regret* as:

$$\textsc{Regret}(T, \pi, \mathcal{H}^{hist}) = T \cdot \textsc{Opt} - \textstyle\sum_{t=1}^{T} \mu(A_t) \,. \tag{2}$$

where the dependence on $\mathcal{H}^{hist}$ highlights that A_t can additionally depend on the historical dataset.

2.2 Finite, Continuous, and Combinatorial Action Spaces

Finite-Armed Bandit. The finite-armed bandit model can be viewed in this framework by considering K discrete actions $\mathcal{A} = [K] = \{1, \ldots, K\}$.

Combinatorial Multi-Armed Bandit for Continuous Resource Allocation (CMAB-CRA). A central planner has access to a metric space $\mathcal{S}$ of resources with metric $d_{\mathcal{S}}$. They are tasked with splitting a total amount of B divisible budget across N different resources within $\mathcal{S}$. An action consists of choosing N resources, i.e., N points in $\mathcal{S}$, and allocating the budget among that chosen subset. The feasible space of allocations is $\mathcal{B} = [0,1]$ and the feasible action space is:

$$\mathcal{A} = \left\{ (\vec{\mathbf{p}}, \vec{\beta}) \in \mathcal{S}^N \times \mathcal{B}^N \;\middle|\; \textstyle\sum_{i=1}^{N} \beta^{(i)} \le B, \quad d_{\mathcal{S}}(\mathbf{p}^{(i)}, \mathbf{p}^{(j)}) \ge \epsilon \quad \forall i \ne j \right\} . \tag{3}$$

The chosen action must satisfy the budgetary constraint (i.e., $\sum_i \beta^{(i)} \le B$), and the resources must be distinct (aka ϵ-away from each other according to $d_{\mathcal{S}}$ for some $\epsilon > 0$) to ensure the "same" resource is not chosen at multiple allocations. We additionally assume that $\Re$ decomposes independently over the (resource, allocation) pairs, in that $\mu(a) = \sum_{i=1}^{N} \mu(\mathbf{p}^{(i)}, \beta^{(i)})$. Lastly, we assume

Algorithm 1 ARTIFICIAL REPLAY

Require: Historical dataset $\mathcal{H}^{hist} = \{(a_j^{\mathcal{H}}, R_j^{\mathcal{H}})\}_{j\in[H]}$, base algorithm Π

Initialize set of used historical data points $\mathcal{H}_1^{on} = \emptyset$, and set of online data $\mathcal{H}_1 = \emptyset$
for $t = \{1, 2, \ldots\}$ **do**
  Initialize `flag` to be `True`
  while `flag` is `True` **do**
    Pick action $\tilde{A}_t \sim \Pi(\mathcal{H}_t^{on} \cup \mathcal{H}_t)$
    if $\tilde{A}_t$ is not contained in $\mathcal{H}^{hist} \setminus \mathcal{H}_t^{on}$ **then**
      Update `flag` to be `False` ▷ Finish a full timestep
      Set online action $A_t = \tilde{A}_t$
      Execute action A_t and observe reward $R_t \sim \Re(A_t)$ ▷ Take online sample
      Update $\mathcal{H}_{t+1} = \mathcal{H}_t \cup \{(A_t, R_t)\}$ and $\mathcal{H}_{t+1}^{on} = \mathcal{H}_t^{on}$
    else
      Update $\mathcal{H}_t^{on}$ to include one sample for $\tilde{A}_t$ from historical dataset $\mathcal{H}^{hist}$

the algorithm observes semi-bandit feedback of the form $(\mathbf{p}_t^{(i)}, \beta_t^{(i)}, R_t^{(i)})_{i\in[N]}$ for each resource and allocation pair sampled according to $\Re(\mathbf{p}_t^{(i)}, \beta_t^{(i)})$. Zuo & Joe-Wong (2021) proposed a discrete model of this problem as a generalization of the works in Dagan & Koby (2018); Lattimore et al. (2014; 2015) specialize it to consider scheduling a finite set of resources to maximize the expected number of jobs finished.

Extension to Green Security. The CMAB-CRA model can be used to specify green security domains from Xu et al. (2021) by letting the space $\mathcal{S}$ represent a protected area and letting $\mathcal{B}$ represent the discrete set of patrol resources to allocate, such as number of ranger hours per week, with the total budget B being 40 hours. This formulation generalizes to a more realistic continuous space model of the landscape, instead of the artificial fixed discretization that was considered in prior work consisting of 1×1 sq. km regions of the park. This also highlights the practical necessity that the chosen resources (here, the patrol locations) are ϵ-far away to ensure sufficient spread. In Section 5, we show that enabling patrol planning at a continuous level can help park rangers more precisely identify poaching hotspots.

3 ARTIFICIAL REPLAY FOR HARNESSING HISTORICAL DATA

We propose ARTIFICIAL REPLAY, a meta-algorithm that can be integrated with any base algorithm to incorporate historical data. We later prove that for any base algorithm satisfying *independence of irrelevant data* (IIData), a novel property we introduce, ARTIFICIAL REPLAY has identical regret to an approach which uses the full historical data upfront—showing that our approach reduces computation costs without trading off performance. Additionally, in Appendix E we discuss empirical improvements of ARTIFICIAL REPLAY applied to Thompson Sampling and Information Directed Sampling, two algorithms which do not satisfy IIData.

Algorithm Formulation. Any algorithm for online stochastic bandits can be thought of as a function mapping arbitrary ordered histories (i.e., collections of observed (a, R) pairs) to a distribution over actions in $\mathcal{A}$. More specifically, let $\Pi : \mathcal{D} \to \Delta(\mathcal{A})$ be an arbitrary base algorithm where $\mathcal{D}$ denotes the collection of possible histories (i.e., $\mathcal{D} = \cup_{i\geq 0}(\mathcal{A} \times \mathbb{R}_+)$). The policy obtained by a base algorithm Π *without incorporating historical data* simply takes the action sampled according to the policy $\pi_t^{\text{IGNORANT}(\Pi)} = \Pi(\mathcal{H}_t)$ where $\mathcal{H}_t$ is the data observed by timestep t. In comparison, consider an algorithm $\pi_t^{\text{FULL START}(\Pi)}$ which follows the same policy but uses the *full historical data upfront*, so takes the action sampled according to $\Pi(\mathcal{H}^{hist} \cup \mathcal{H}_t)$.

3.1 ARTIFICIAL REPLAY

The ARTIFICIAL REPLAY meta-algorithm incorporates the historical data $\mathcal{H}^{hist}$ into an arbitrary base algorithm Π, resulting in a policy we denote by $\pi^{\text{ARTIFICIAL REPLAY}(\Pi)}$. See Algorithm 1 for the pseudocode. We let $\mathcal{H}_t^{on}$ be the set of historical datapoints used by the start of time t. Initially, $\mathcal{H}_1^{on} = \emptyset$. For an arbitrary timestep t, the ARTIFICIAL REPLAY approach works as follows:

Let $\tilde{A}_t \sim \Pi(\mathcal{H}_t^{on} \cup \mathcal{H}_t)$ be the *proposed action* at the start of time t. Since we are focused on *simulating* the algorithm with historical data, we break into cases whether or not the current set of unused historical datapoints (i.e., $\mathcal{H}^{hist} \setminus \mathcal{H}_t^{on}$) contains any additional information about $\tilde{A}_t$.

- **No historical data available**: If $\tilde{A}_t$ is not contained in $\mathcal{H}^{hist} \setminus \mathcal{H}_t^{on}$, then the *selected action* is $A_t = \tilde{A}_t$, and we advance to timestep $t+1$. We additionally set $\mathcal{H}_{t+1}^{on} = \mathcal{H}_t^{on}$.
- **Historical data available**: If $\tilde{A}_t$ is contained in $\mathcal{H}^{hist} \setminus \mathcal{H}_t^{on}$, add that data point to $\mathcal{H}_t^{on}$ and repeat by picking another *proposed action*. We remain at time t.

Strikingly, our framework imposes minimal computational and storage overhead on top of existing algorithms, simply requiring a data structure to verify whether $\tilde{A} \in \mathcal{H}^{hist} \setminus \mathcal{H}_t^{on}$, which can be performed with hashing techniques. It is clear that the runtime and storage complexity of ARTIFICIAL REPLAY is no worse than FULL START. We also note that most practical bandit applications incorporate historical data obtained from database systems (e.g. content recommendation systems, wildlife poaching model discussed). This historical data will be stored regardless of the algorithm being employed, and so the key consideration is around computational requirements and not storage.

Additionally, our approach extends naturally to the following different models:

Continuous Spaces. The ARTIFICIAL REPLAY framework can be applied in continuous action spaces with discretization-based algorithms. For example, suppose that Π wants to select an action $a \in \mathcal{A}$, but the historical data has a sample from $a + \epsilon$, a slightly perturbed point. Discretization-based algorithms avoid precision issues since they map the continuous space to a series of regions which together cover the action set, and run algorithms or subroutines over the discretization. Checking for historical data simply checks for data within the bounds of the chosen discretized action.

Semi-Bandit Feedback. ARTIFICIAL REPLAY also naturally extends to combinatorial action sets with semi-bandit feedback where actions are *decomposable*, that is, they can be written as $a = (a_1, \ldots, a_N)$ with independent rewards. Suppose that Π wants to select an action $a = (a_1, a_2, \ldots, a_N)$ but the historical data has a sample from $(a'_1, a_2, \ldots a'_N)$. Even if the combinatorial action a does not appear in its entirety in the historical data, as long as there exists some subcomponent $a_i^{\mathcal{H}}$ (sometimes referred to as "subarm" in combinatorial bandits) in the historical data (e.g., a_2) , we can add that subcomponent $a_i^{\mathcal{H}}$ to $\mathcal{H}_t^{on}$ to update the base algorithm.

3.2 INDEPENDENCE OF IRRELEVANT DATA AND REGRET COUPLING

It is not immediately clear how to analyze the regret of ARTIFICIAL REPLAY. To enable regret analysis, we introduce a new property for bandit algorithms, *independence of irrelevant data*, which essentially requires that when an algorithm is about to take an action, providing additional data about other actions (i.e., those not selected by the algorithm) will not influence the algorithm's decision.

Definition 3.1 (Independence of irrelevant data)**.** *A deterministic base algorithm Π satisfies the* **independence of irrelevant data** *(IIData) property if whenever $A = \Pi(\mathcal{H})$ then*

$$\Pi(\mathcal{H}) = \Pi(\mathcal{H} \cup \mathcal{H}') \tag{4}$$

for any $\mathcal{H}'$ containing data from any actions a' other than A (that is, $a' \neq A$).

IIData is a natural robustness property for an algorithm to satisfy, highlighting that the algorithm evaluates actions independently when making decisions. IIData is conceptually analogous to the independence of irrelevant alternatives (IIA) axiom in computational social choice as a desiderata used to evaluate voting rules (Arrow, 1951). In Theorem 3.2 we show that for any base algorithm satisfying IIData, the regret of $\pi^{\text{FULL START}(\Pi)}$ and $\pi^{\text{ARTIFICIAL REPLAY}(\Pi)}$ will be equal.

Theorem 3.2. *Suppose that algorithm Π satisfies the independence of irrelevant data property. Then for any problem instance, horizon T, and historical dataset $\mathcal{H}^{hist}$ we have the following:*

$$\pi_t^{\text{ARTIFICIAL REPLAY}(\Pi)} \overset{d}{=} \pi_t^{\text{FULL START}(\Pi)}$$

$$\text{REGRET}(T, \pi^{\text{ARTIFICIAL REPLAY}(\Pi)}, \mathcal{H}^{hist}) \overset{d}{=} \text{REGRET}(T, \pi^{\text{FULL START}(\Pi)}, \mathcal{H}^{hist}) \,.$$

Algorithm 2 Monotone UCB (MONUCB)

Initialize $n_1(a) = 0$, $\overline{\mu}_1(a) = 1$, and $\text{UCB}_1(a) = 1$ for each $a \in [K]$
for $t = \{1, 2, \ldots\}$ **do**
  Let $A_t = \arg\max_{a \in [K]} \text{UCB}_t(a)$
  Receive reward R_t sampled from $\Re(A_t)$
  Update $n_{t+1}(A_t) = n_t(A_t) + 1$, $n_{t+1}(a) = n_t(a)$ for $a \neq A_t$
  Update $\overline{\mu}_{t+1}(A_t) = (n_t(A_t)\overline{\mu}_t(A_t) + R_t)/n_{t+1}(A_t)$, $\overline{\mu}_{t+1}(a) = \overline{\mu}_t(a)$ for $a \neq A_t$
  Update $\text{UCB}_{t+1}(a) = \text{UCB}_t(a)$ for $a \neq A_t$ and
    $\text{UCB}_{t+1}(A_t) = \min\{\text{UCB}_t(A_t), \overline{\mu}_{t+1}(A_t) + \sqrt{2\log(T)/n_{t+1}(A_t)}\}$

This theorem shows that ARTIFICIAL REPLAY allows us to achieve identical regret guarantees as FULL START while simultaneously using data more efficiently. In the subsequent section, we show three example regret-optimal algorithms which satisfy this property, even in the complex CMAB-CRA setting. The algorithms we modify are all UCB-based algorithms. In fact, it is easy to modify most UCB-based algorithms to satisfy IIData by simply imposing monotonicity of the confidence bound estimates for an action rewards. This is easily implementable and preserves all regret guarantees. While for brevity we only discuss IIData algorithms in the K-Armed and CMAB-CRA set-up, it is easy to see how to modify other UCB-based algoriths (e.g. LinUCB for linear bandits (Abbasi-Yadkori et al., 2011)) to satisfy IIData. In the existing bandit literature, there has been a narrow focus on only finding regret-optimal algorithms. We propose that IIData is another desirable property that implies ease and robustness for optimally and efficiently incorporating historical data.

4 IIDATA ALGORITHMS

In this section, we provide IIData algorithms with optimal regret guarantees for two settings: the K-armed and CMAB-CRA models. We show that IIData is easy to guarantee for UCB algorithms requiring only a minor modification to existing algorithms while not impacting confidence bounds guarantees.

We defer algorithm details to Appendix D and proofs to Appendix F. We'll show in Appendix E that in practice, ARTIFICIAL REPLAY still performs nearly optimally even with algorithms that do not satisfy IIData.

4.1 K-ARMED BANDITS

The first algorithm we propose, named Monotone UCB (denoted as MONUCB), is derived from the UCB1 algorithm introduced in Auer et al. (2002). At every timestep t, the algorithm tracks the following: *(i)* $\overline{\mu}_t(a)$ for the estimated mean reward of action $a \in [K]$, *(ii)* $n_t(a)$ for the number of times the action a has been selected by the algorithm prior to timestep t, and *(iii)* $\text{UCB}_t(a)$ for an upper confidence bound estimate for the reward of action a. At every timestep t, the algorithm picks the action A_t which maximizes $\text{UCB}_t(a)$ (breaking ties deterministically). After observing R_t, we increment $n_{t+1}(A_t) = n_t(A_t) + 1$, update $\overline{\mu}_{t+1}(A_t)$, and set:

$$\text{UCB}_{t+1}(A_t) = \min\left\{\text{UCB}_t(A_t),\ \overline{\mu}_{t+1}(A_t) + \sqrt{\frac{2\log(T)}{n_{t+1}(A_t)}}\right\}. \tag{5}$$

The only modification of Monotone UCB from standard UCB is the additional step forcing the UCB estimates to be monotone decreasing over t. It is clear that this modification has no affect on the regret guarantees. Under the "good event" analysis, if $\text{UCB}_t(a) \geq \mu(a)$ with high probability, then the condition still holds at time $t+1$, even after observing a new data point. In the following theorem, we show that MONUCB satisfies IIData and is regret-optimal, achieving the same instance-dependent regret bound as the standard UCB algorithm.

Theorem 4.1. *Monotone UCB satisfies the IIData property and has for* $\Delta(a) = \max_{a'} \mu(a') - \mu(a)$*:*

$$\text{REGRET}(T, \pi^{\text{IGNORANT(MONUCB)}}, \mathcal{H}^{hist}) = O(\sum_a \log(T)/\Delta(a)). \tag{6}$$

This guarantee allows us to use Theorem 3.2 to establish that ARTIFICIAL REPLAY and FULL START have identical regret with MONUCB as a base algorithm. In the next theorem, we

show that ARTIFICIAL REPLAY is robust to *spurious data*, where the historical data has excess samples $a_j^{\mathcal{H}}$ coming from poor performing actions. Spurious data imposes computational challenges, since the FULL START approach will pre-process the full historical dataset regardless of the observed rewards or the inherent value of the historical data. In contrast, ARTIFICIAL REPLAY will only use the amount of data useful for learning.

Theorem 4.2. *For every $H \in \mathbb{N}$ there exists a historical dataset $\mathcal{H}^{hist}$ with $|\mathcal{H}^{hist}| = H$ where the runtime of $\pi^{\text{FULL START(MONUCB)}} = \Omega(H + T)$ whereas the runtime of $\pi^{\text{ARTIFICIAL REPLAY(MONUCB)}} = O(T + \min\{\sqrt{T}, \log(T)/\min_a \Delta(a)^2\})$.*

This highlights that the computational overhead of ARTIFICIAL REPLAY in comparison to FULL START can be arbitrarily better. For storage requirements, the FULL START algorithm requires $O(K)$ storage for maintaining estimates for each arm. In contrast, a naive implementation of ARTIFICIAL REPLAY requires $O(K + H)$ storage since the entire historical dataset needs to be stored. However, using hashing techniques can address the extra H factor. In Section 4.2 we will see an example where IIData additionally has strong storage benefits over FULL START.

Lastly, to complement the computational improvements of ARTIFICIAL REPLAY applied to MONUCB, we can also show an improvement of regret. This analysis crucially uses the regret coupling, since FULL START(MONUCB) is much easier to reason about than its ARTIFICIAL REPLAY counterpart.

Theorem 4.3. *Let H_a be the number of datapoints in $\mathcal{H}^{hist}$ for each action $a \in [K]$. Then the regret of Monotone UCB with historical dataset $\mathcal{H}^{hist}$ is:*

$$\text{REGRET}(T, \pi^{\text{ARTIFICIAL REPLAY(MONUCB)}}, \mathcal{H}^{hist}) \leq O\Big(\sum_{a \in [K]: \Delta_a \neq 0} \max\big\{0, \tfrac{\log(T)}{\Delta(a)} - H_a \Delta(a)\big\}\Big).$$

Theorem 4.2 together with Theorem 4.3 helps highlight the advantage of using ARTIFICIAL REPLAY over FULL START in terms of improving computational complexity while maintaining an equally improved regret guarantee. This reduces to the standard UCB guarantee when $\mathcal{H}^{hist} = \emptyset$. Moreover, it highlights the impact historical data can have on the regret. If $|H_a| \geq \log(T)/\Delta(a)^2$ for each a then the regret of the algorithm will be constant not scaling with T. We note that there are no existing regret lower bounds for incorporating historical data in bandit algorithms. Our main goal is not to improve regret guarantees (although Theorem 4.3 highlights the advantage of historical data), but instead highlight a simple, intuitive, and implementable approach through ARTIFICIAL REPLAY which matches the performance of FULL START while simultaneously having smaller compute.

We close with an example of a K-armed bandit algorithm which does not satisfy the IIData assumption. Thompson Sampling (Russo et al., 2018), which samples arms according to the posterior probability that they are optimal, does not satisfy IIData. Data from other actions other than the one chosen will adjust the posterior distribution, and hence will adjust the selection probabilities as well. While we do not obtain a regret coupling, in Fig. 8 (appendix) we show that there are still empirical gains for using ARTIFICIAL REPLAY over FULL START across a variety of base algorithms.

4.2 CMAB-CRA

Incorporating historical data optimally and efficiently is difficult in continuous action settings. Two natural approaches are to *(i)* discretize the action space $\mathcal{A}$ based on the data using nearest neighbor estimates, or *(ii)* learn a regression of the mean reward using available data. Consider a setting where excessive data is collected from poor-performing actions. Discretization-based algorithms will unnecessarily process and store a large number of discretizations in low-performing regions of the space. Regression-based methods will use compute resources to learn an accurate predictor of the mean reward in irrelevant regions. The key issues are that the computational and storage cost grows with the size of the historical dataset, and the estimation and discretization is done independent of the quality of the reward.

To contrast this approach, we present two discretization-based algorithms that satisfy IIData with strong performance guarantees. In particular, we detail fixed and adaptive discretization (ADAMONUCB in Algorithm 3) algorithms that only use the historical dataset to update estimates

of the reward. Due to space, we describe the algorithms only at a high level and defer details to Appendix D.

Our algorithms are Upper Confidence Bound (UCB) style as the selection rule maximizes Eq. (1) over the combinatorial action set (Eq. (3)) through a discretization of $\mathcal{S}$. For each allocation $\beta \in \mathcal{B}$, the algorithm maintains a collection of regions $\mathcal{P}_t^\beta$ of $\mathcal{S}$ which covers $\mathcal{S}$. For the fixed discretization variant, $\mathcal{P}_t^\beta$ is fixed at the start of learning, and in the adaptive discretization version it is refined over the course of learning based on observed data. At every timestep t and region $\mathcal{R} \in \mathcal{P}_t^\beta$, the algorithm tracks the following: *(i)* $\overline{\mu}_t(\mathcal{R}, \beta)$ for the estimated mean reward of region $\mathcal{R}$ at allocation β, *(ii)* $n_t(\mathcal{R}, \beta)$ for the number of times $\mathcal{R}$ has been selected at allocation β prior to timestep t, and *(iii)* $\text{UCB}_t(\mathcal{R}, \beta)$ for an upper confidence bound estimate. At a high level, our algorithm performs three steps in each iteration t:

1. **Action selection**: Greedily select at most N regions in $\mathcal{P}_t^\beta$ to maximize $\text{UCB}_t(\mathcal{R}, \beta)$ subject to the budget constraints (see Eq. (10) in the appendix). Note that we must additionally ensure that each region is selected at only a *single* allocation value.
2. **Update parameters**: For each of the selected regions, increment $n_t(\mathcal{R}, \beta)$ by one, update $\overline{\mu}_t(\mathcal{R}, \beta)$ based on observed data, and set $\text{UCB}_{t+1}(\mathcal{R}, \beta) = \min\{\text{UCB}_t(\mathcal{R}, \beta), \overline{\mu}_t(\mathcal{R}, \beta) + b(n_t(\mathcal{R}, \beta))\}$ for some appropriate bonus term $b(\cdot)$. This enforces monotonicity in the UCB estimates similar to MonUCB and is required for the IIData property.
3. **Re-partition**: This step differentiates the *adaptive discretization* algorithm from fixed discretization, which maintains the same partition across all timesteps. We split a region when the confidence in its estimate (i.e., $b(n_t(\mathcal{R}, \beta))$) is smaller than the diameter of the region. This condition may seem independent of the *quality* of a region, but since it is incorporated into a learning algorithm, the number of samples in a region is correlated with its reward. In Fig. 4 (appendix) we highlight how the adaptive discretization algorithm hones in on regions with large reward without knowing the reward function before learning.

These algorithms modify existing approaches applied to CMAB-CRA in the bandit and reinforcement learning literature, which have been shown to be regret-optimal (Xu et al., 2021; Sinclair et al., 2021). We additionally note that these approaches are IIData.

Theorem 4.4. *The fixed and adaptive discretization algorithms when using a "greedy" solution to solve Eq.* (1) *have property IIData.*

Here we require the algorithm to use the standard "greedy approximation" to Eq. (1), which is a knapsack problem in the CMAB-CRA set-up (Williamson & Shmoys, 2011). This introduces additional approximation ratio limitations in general. However, under additional assumptions on the mean reward function $\mu(\mathbf{p}, \beta)$, the greedy solution is provably optimal. For example, optimality of the greedy approximation holds when $\mu(\mathbf{p}, \beta)$ is piecewise linear and monotone, or more broadly when $\mu(a)$ is submodular. See Appendix D for more discussion.

Finally, we comment that the Full Start implementation of these adaptive discretization algorithms will have storage and computational costs proportional to the size of the historical dataset (since the algorithms ensure that the discretization scales with respect to the number of samples). In contrast, Artificial Replay uses only a fraction of the historical dataset and so again has improved computation and storage complexity. This is validated in the experimental results in Appendix E.

5 Experiments

We show the benefits of Artificial Replay by showing that our meta-algorithm achieves identical performance to Full Start while offering significant practical advantages in reducing runtime and storage. We consider two classes of bandit domains: K-armed and CMAB-CRA. As part of our evaluation on combinatorial bandits, we introduce a new model for green security games with continuous actions by adaptively discretizing the landscape of a large protected area of Murchison Falls National Park in Uganda.

All of the code to reproduce the experiments is available at `https://github.com/lily-x/artificial-replay`. Results are averaged over 60 iterations with random seeds, with standard error plotted; experiment details and additional results are available in Appendix E.

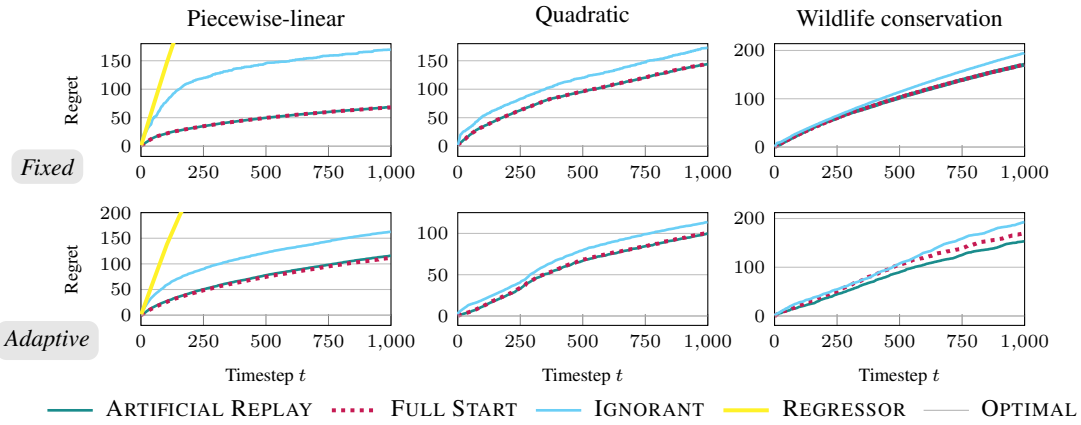

Figure 1: *(CMAB-CRA)* Cumulative regret (y-axis; lower is better) across time $t \in [T]$. ARTIFICIAL REPLAY performs equally as well as FULL START across all domain settings, including both fixed discretization (top row) and adaptive discretization (bottom). REGRESSOR performs quite poorly.

Domains. We conduct experiments on the two bandit models described in Section 2.2: finite K-armed bandits and CMAB-CRA, using both fixed and adaptive discretization. For the continuous combinatorial setting, we provide two stylized domains: a piecewise-linear and a quadratic reward function. To emphasize the practical benefit of ARTIFICIAL REPLAY, we evaluate on a real-world resource allocation setting for biodiversity conservation. We study real ranger patrol data from Murchison Falls National Park, shared as part of a collaboration with the Uganda Wildlife Authority and the Wildlife Conservation Society. We use historical patrol observations to build the history $\mathcal{H}^{hist}$; we analyze these historical observations in detail in Appendix E to show that this dataset exhibits both spurious data and imbalanced coverage as discussed in Section 4.

Baselines. We compare ARTIFICIAL REPLAY against IGNORANT and FULL START approaches for each setting. In the K-armed model, we use MONUCB as the base algorithm. In CMAB-CRA we use fixed and adaptive discretization as well as REGRESSOR, a neural network learner that is a regression-based approach analogue to FULL START. REGRESSOR is initially trained on the entire historical dataset, then iteratively retrained after 128 new samples are collected. We also compute for each setting the performance of an OPTIMAL action based on the true rewards and a RANDOM baseline that acts randomly while satisfying the budget constraint.

Results. The results in Fig. 1 empirically validate our theoretical result from Theorem 3.2: the performance of ARTIFICIAL REPLAY is identical to that of FULL START, and reduces regret considerably compared to the naive IGNORANT approach. We evaluate the regret (compared to OPTIMAL) of each approach across time $t \in [T]$. Concretely, we consider the three domains of piecewise-linear reward, quadratic reward, and green security with continuous space $\mathcal{S} = [0, 1]^2$, $N = 5$ possible action components, a budget $B = 2$, and 3 levels of effort. We include $H = 300$ historical data points. See Fig. 9 (appendix) for regret and analysis of historical data use on the K-armed bandit.

Not only does ARTIFICIAL REPLAY achieve equal performance, but its computational benefits over FULL START are clear even on practical problem sizes. As we increase historical data from $H = \{10; 100; 1{,}000; 10{,}000\}$ in Fig. 2, the proportion of irrelevant data increases. Our method achieves equal performance, overcoming the previously unresolved challenge of *spurious data*, while FULL START suffers from arbitrarily worse storage complexity (Theorem 4.2). With 10,000 historical samples and a time horizon of 1,000, we see that 58.2% of historical samples are irrelevant to producing the most effective policy.

When faced with *imbalanced data coverage*, the benefits of ARTIFICIAL REPLAY become clear—most notably in the continuous action setting with adaptive discretization. In Fig. 3, as we increase the number of historical samples on bad regions (bottom 20th percentile of reward), the additional data require finer discretization, leading to arbitrarily worse storage and computational complexity for FULL START with equal regret. In Fig. 3(c), we see that with 10% of data on bad arms, AR-

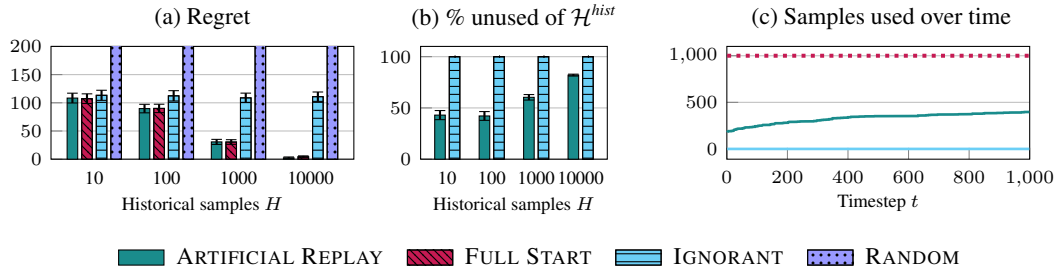

Figure 2: *(K-Armed)* Increasing the number of historical samples H leads FULL START to use unnecessary data, particularly as H gets very large. ARTIFICIAL REPLAY achieves equal performance in terms of regret (plot a) while using less than half the historical data (plot b). In plot c we see that with $H = 1{,}000$ historical samples, ARTIFICIAL REPLAY uses (on average) 117 historical samples before taking its first online action. The number of historical samples used increases at a decreasing rate, using only 396 of 1,000 total samples by the horizon T. Results are shown on the K-armed bandit setting with $K = 10$ and horizon $T = 1{,}000$.

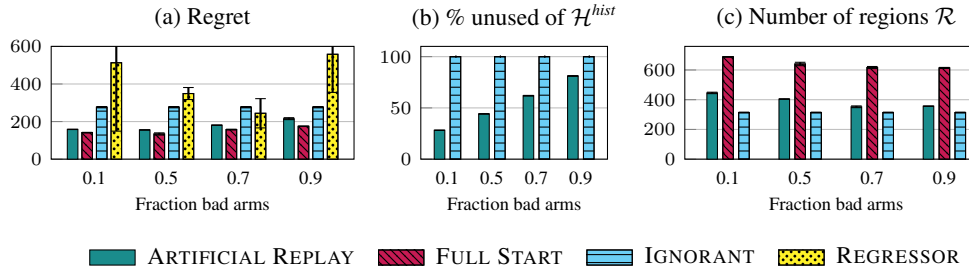

Figure 3: *(CMAB-CRA)* Holding $H = 10{,}000$ constant, we increase the fraction of historical data samples on bad arms (bottom 20% of rewards). The plots show (a) regret, (b) % of unused historical data and (c) number of discretized regions in partition $\mathcal{P}$. ARTIFICIAL REPLAY enables significantly improved runtime and reduced storage while matching the performance of FULL START. Results on the CMAB-CRA setting with adaptive discretization on the quadratic domain.

TIFICIAL REPLAY requires only 446 regions $\mathcal{R}$ compared to 688 used by FULL START; as we get more spurious data and that fraction increases to 90%, then ARTIFICIAL REPLAY requires only 356 regions while FULL START still stores 614 regions.

6 CONCLUSION

We present ARTIFICIAL REPLAY, a meta-algorithm that modifies *any base bandit algorithm* to efficiently harness historical data. We show that under a widely applicable IIData condition, the regret of ARTIFICIAL REPLAY (as a random variable) is distributionally identical to that of a full warm-start approach, while also guaranteeing significantly better time complexity. We additionally give examples of regret-optimal IIData algorithms in the K-armed and CMAB-CRA settings. Our experimental results highlight the advantage of using ARTIFICIAL REPLAY over FULL START via a variety of base algorithms, applied to K-armed and continuous combinatorial bandit models, including for algorithms such as Thompson sampling and Information Directed Sampling (IDS) that do not exhibit IIData. Directions for future work include *(i)* find IIData algorithms in other bandit domains such as linear contextual bandits, *(ii)* incorporate the ARTIFICIAL REPLAY approach into reinforcement learning, and *(iii)* provide theoretical bounds showing that ARTIFICIAL REPLAY has *optimal* data usage when incorporating historical data.

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Kenneth J Arrow. *Social choice and individual values*. John Wiley & Sons, 1951.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Hamsa Bastani, Kimon Drakopoulos, Vishal Gupta, Ioannis Vlachogiannis, Christos Hadjicristodoulou, Pagona Lagiou, Gkikas Magiorkinis, Dimitrios Paraskevis, and Sotirios Tsiodras. Efficient and targeted COVID-19 border testing via reinforcement learning. *Nature*, 599(7883): 108–113, 2021.

Djallel Bouneffouf, Srinivasan Parthasarathy, Horst Samulowitz, and Martin Wistuba. Optimal exploitation of clustering and history information in multi-armed bandit. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, (IJCAI)*, pp. 2016–2022, 2019.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

Stefano Canessa, Gurutzeta Guillera-Arroita, José J Lahoz-Monfort, Darren M Southwell, Doug P Armstrong, Iadine Chadès, Robert C Lacy, and Sarah J Converse. When do we need more data? a primer on calculating the value of information for applied ecologists. *Methods in Ecology and Evolution*, 6(10):1219–1228, 2015.

Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the Thirtieth International Conference on Machine Learning (ICML)*, volume 28, pp. 151–159. PMLR, 2013.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 208–214, 2011.

Yuval Dagan and Crammer Koby. A better resource allocation algorithm with semi-bandit feedback. In *Algorithmic Learning Theory*, pp. 268–320. PMLR, 2018.

Adam N Elmachtoub, Ryan McNellis, Sechan Oh, and Marek Petrik. A practical method for solving contextual bandit problems using decision trees. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

Fei Fang, Peter Stone, and Milind Tambe. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *Proceedings of the Twenty-fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

Shahrzad Gholami, Amulya Yadav, Long Tran-Thanh, Bistra Dilkina, and Milind Tambe. Don't put all your strategies in one basket: Playing green security games with imperfect prior knowledge. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 395–403, 2019.

Debarun Kar, Benjamin Ford, Shahrzad Gholami, Fei Fang, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Mustapha Nsubaga, et al. Cloudy with a chance of poaching: Adversary behavior modeling and forecasting with real-world poaching data. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2017.

Robert Kleinberg and Nicole Immorlica. Recharging bandits. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 309–319. IEEE, 2018.

Robert Kleinberg, Alexandru Niculescu-Mizil, and Yogeshwer Sharma. Regret bounds for sleeping experts and bandits. *Machine Learning*, 80(2):245–272, 2010.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Bandits and experts in metric spaces. *Journal of the ACM (JACM)*, 66(4):1–77, 2019.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

Tor Lattimore, Koby Crammer, and Csaba Szepesvári. Optimal resource allocation with semi-bandit feedback. *UAI*, 2014.

Tor Lattimore, Koby Crammer, and Csaba Szepesvári. Linear multi-resource allocation with semi-bandit feedback. *Advances in Neural Information Processing Systems*, 28, 2015.

Tara G Martin, Abbey E Camaclang, Hugh P Possingham, Lynn A Maguire, and Iadine Chadès. Timing of protection of critical habitat matters. *Conservation Letters*, 10(3):308–316, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with deep reinforcement learning. In *Proceedings of the Twenty-Seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2013.

Thanh H Nguyen, Arunesh Sinha, Shahrzad Gholami, Andrew Plumptre, Lucas Joppa, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, and Rob Critchlow. CAPTURE: A new predictive anti-poaching tool for wildlife protection. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, 2016.

Bastian Oetomo, R Malinga Perera, Renata Borovica-Gajic, and Benjamin IP Rubinstein. Cutting to the chase with warm-start contextual bandits. In *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 459–468. IEEE, 2021.

Andrew J Plumptre, Richard A Fuller, Aggrey Rwetsiba, Fredrick Wanyama, Deo Kujirakwinja, Margaret Driciru, Grace Nangendo, James EM Watson, and Hugh P Possingham. Efficiently targeting resources to deter illegal activities in protected areas. *Journal of Applied Ecology*, 51 (3):714–725, 2014.

Larry D Pyeatt and Adele E Howe. Decision tree function approximation in reinforcement learning. In *Proceedings of the Third International Symposium on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Models*, volume 2, pp. 70–77, 2001.

Yundi Qian, Chao Zhang, Bhaskar Krishnamachari, and Milind Tambe. Restless poachers: Handling exploration-exploitation tradeoffs in security domains. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, pp. 123–131, 2016.

Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. *Operations Research*, 66(1):230–252, 2018.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on Thompson sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96, 2018.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *Proceedings of the Fourth International Conference on Learning Representations (ICLR)*, 2017.

Pannagadatta Shivaswamy and Thorsten Joachims. Multi-armed bandit problems with history. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1046–1054. PMLR, 2012.

Sean R Sinclair, Siddhartha Banerjee, and Christina Lee Yu. Adaptive discretization in online reinforcement learning. *arXiv preprint arXiv:2110.15843*, 2021.

Aleksandrs Slivkins. Contextual bandits with similarity information. In *Proceedings of the Twenty-Fourth Annual Conference On Learning Theory (COLT)*, pp. 679–702, 2011.

Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning*, 12(1-2):1–286, 2019. ISSN 1935-8237.

Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research*, 16(52):1731–1755, 2015.

William TB Uther and Manuela M Veloso. Tree based discretization for continuous state space reinforcement learning. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI)*, 1998.

Lu Wang, Chengyu Wang, Keqiang Wang, and Xiaofeng He. BiUCB: A contextual bandit algorithm for cold-start and diversified recommendation. In *2017 IEEE International Conference on Big Knowledge (ICBK)*, pp. 248–253, 2017.

David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

Lily Xu, Shahrzad Gholami, Sara Mc Carthy, Bistra Dilkina, Andrew Plumptre, Milind Tambe, Rohit Singh, Mustapha Nsubuga, Joshua Mabonga, Margaret Driciru, et al. Stay ahead of poachers: Illegal wildlife poaching prediction and patrol planning under uncertainty with field test evaluations. In *Proceedings of the IEEE 36th International Conference on Data Engineering (ICDE)*, 2020.

Lily Xu, Elizabeth Bondi, Fei Fang, Andrew Perrault, Kai Wang, and Milind Tambe. Dual-mandate patrols: Multi-armed bandits for green security. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Rong Yang, Benjamin J Ford, Milind Tambe, and Andrew Lemieux. Adaptive resource allocation for wildlife protection against illegal poachers. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 453–460, 2014.

Chicheng Zhang, Alekh Agarwal, Hal Daumé Iii, John Langford, and Sahand Negahban. Warm-starting contextual bandits: Robustly combining supervised and bandit feedback. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pp. 7335–7344, 2019.

Jinhang Zuo and Carlee Joe-Wong. Combinatorial multi-armed bandits for resource allocation. In *55th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–4. IEEE, 2021.

Jinhang Zuo, Xiaoxi Zhang, and Carlee Joe-Wong. Observe before play: Multi-armed bandit with pre-observations. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pp. 7023–7030, 2020.

A TABLE OF NOTATION

Symbol	Definition
	Problem setting specifications
$\mathcal{A}$	Feasible action space (see Eq. (3))
$\Re$	Reward distribution, i.e., $\Re : \mathcal{A} \to \Delta([0,1])$
OPT	Optimal objective value of Eq. (1)
T	Time horizon
$\text{REGRET}(\pi, T, \mathcal{H}^{hist})$	Cumulative regret for an algorithm π on T timesteps with historical data $\mathcal{H}^{hist}$
$\mathcal{H}^{hist}, H$	Historical data available to algorithm and number of historical datapoints
a	Generic action $a \in \mathcal{A}$
$a_j^{\mathcal{H}}$	Historical action at index j in history $\mathcal{H}^{hist}$
A_t	Selected action chosen at timestep t
R_t	Reward observed at timestep t from action A_t
Π	Base algorithm that maps ordered (a, R) pairs to a distribution over $\mathcal{A}$
$\pi^{\text{ARTIFICIAL REPLAY}}$	ARTIFICIAL REPLAY framework and its resulting policy
$\pi^{\text{FULL START}}$	FULL START: Full warm starting operation and its resulting policy
π^{IGNORANT}	IGNORANT: Original policy ignoring historical data
	CMAB-CRA specification
CMAB-CRA	Combinatorial Multi-Armed Bandit for Continuous Resource Allocation
$\mathcal{S}, \mathcal{B}$	The continuous resource space, and (discrete) space of allocation values
$d_{\mathcal{S}}, d_{\text{max}}$	Metric over $\mathcal{S}$ and the diameter of $\mathcal{S}$
N, ϵ	Maximum number of regions which can be selected, and the minimum distance
$\Re(\mathbf{p}, \vec{\beta})$	Reward distribution for a particular allocation $(\mathbf{p}, \vec{\beta})$
$\mu(\mathbf{p}, \vec{\beta})$	Mean reward for a particular allocation $(\mathbf{p}, \vec{\beta})$
	Monotone UCB
$\overline{\mu}_t(a), n_t(a)$	Mean reward estimates and number of samples for action $a \in [K]$
$\text{UCB}_t(a)$	Upper confidence bound estimate of action a
	Fixed Discretization
$\mathcal{P}$	Fixed ϵ covering of $\mathcal{S}$
$\overline{\mu}_t(\mathcal{R}, \beta), n_t(\mathcal{R}, \beta)$	Mean reward estimates and number of samples for region $\mathcal{R} \in \mathcal{P}$
$\text{UCB}_t(\mathcal{R}, \beta)$	Upper confidence bound estimate of $\mu(\mathcal{R}, \beta)$
$b(t)$	Bonus term (confidence radius) for a region which has been selected t times
	Adaptive Discretization
$\mathcal{P}_t^{\beta}$	Partition of space $\mathcal{S}$ at timestep t for allocation $\beta \in \mathcal{B}$
$\overline{\mu}_t(\mathcal{R}, \beta), n_t(\mathcal{R}, \beta)$	Mean reward estimates and number of samples for region $\mathcal{R} \in \mathcal{P}_t^{\beta}$
$\text{UCB}_t(\mathcal{R}, \beta)$	Upper confidence bound estimate of $\mu(\mathcal{R}, \beta)$
$r(\mathcal{R})$	Diameter of a region $\mathcal{R}$
$b(t)$	Bonus term for a region which has been selected t times

Table 1: List of common notations

B DETAILED RELATED WORK

Multi-armed bandit problems and its sub-variants (including the finite-armed and CMAB-CRA model discussed here) have a long history in the online learning and optimization literature. We highlight the most closely related works below, but for more extensive references see Bubeck et al. (2012); Slivkins (2019); Lattimore & Szepesvári (2020).

Multi-Armed Bandit Algorithms. The design and analysis of bandit algorithms have been considered under a wide range of models. These were first studied in the so-called K-Armed Bandit model in Lai & Robbins (1985); Auer et al. (2002), where the algorithm has access to a finite set of K possible actions at each timestep. The algorithm is characterized by its *Optimistic Upper Confidence Bound* approach, where exploration is garnered by acting greedily with respect to optimistic estimates of the mean reward of each action. Numerous follow-up works have considered similar approaches when designing algorithms in continuous action spaces (Kleinberg et al., 2019), linear reward models (Chu et al., 2011), and with combinatorial constraints (Xu et al., 2021; Zuo & Joe-Wong, 2021). Our work provides a framework for taking existing algorithms to additionally harness historical data. Moreover, in Appendix D we also propose a novel algorithm incorporating data-driven adaptive discretization for combinatorial multi-armed bandits for continuous resource allocation.

Incorporating Historical Data. Several papers have started to investigate techniques for incorporating historical data into bandit algorithms. Shivaswamy & Joachims (2012) started by considering a K-armed bandit model where each arm has a dataset of historical pulls. The authors develop a *Warm Start UCB* algorithm where the confidence term of each arm is initialized based on the full historical data, prior to learning. Similar techniques were extended to models where there are pre-clustered arms, where the authors provide regret guarantees depending on the cluster quality of the fixed clusters (Bouneffouf et al., 2019). These techniques were extended to Bayesian and frequentist linear contextual bandits where the linear feature vector is updated by standard regression over the historical data (Oetomo et al., 2021; Wang et al., 2017). The authors show empirically that these approaches perform better in early rounds, and applied the set-up to recommendation systems. A second line of work has considered warm-starting contextual bandit models with fully supervised historical data and online bandit interaction (Swaminathan & Joachims, 2015; Zhang et al., 2019). Lastly, Zuo et al. (2020) consider augmented data collection schemes where the decision maker can "pre sample" some arms before decisions in the typical bandit set-up.

All of the prior work considers a FULL START approach in specific bandit models to incorporate historical data. In contrast, we provide an *efficient* meta-algorithm for harnessing historical data in *arbitrary stochastic bandit models*. We show our approach has improved runtime and storage over a naive full-start approach. Additionally, we provide, to the best of our knowledge, the first application of incorporating historical data to combinatorial bandit models.

Bandit Algorithms for Green Security Domains. Green security focuses on allocating defender resources to conduct patrols across protected areas to prevent illegal logging, poaching, or overfishing (Fang et al., 2015; Plumptre et al., 2014). These green security challenges have been addressed with game theoretic models (Yang et al., 2014; Nguyen et al., 2016); supervised machine learning (Kar et al., 2017; Xu et al., 2020); and multi-armed bandits, including restless (Qian et al., 2016), recharging (Kleinberg & Immorlica, 2018), adversarial (Gholami et al., 2019), and combinatorial bandits (Xu et al., 2021). Related notions of value of information have been studied in the context of ecological decision making to quantify how obtaining more information can help to better manage ecosystems (Canessa et al., 2015).

Combinatorial Bandits for Resource Allocation. The CMAB-CRA model is a continuous extension of the combinatorial multi-armed bandit for discrete resource allocation (CMAB-DRA) problem studied in Zuo & Joe-Wong (2021). They propose two algorithms which both achieve logarithmic regret when the allocation space is finite or one-dimensional. We extend their upper confidence bound algorithmic approach to consider both fixed and adaptive data-driven discretization of the continuous resource space, and additionally consider the impact of historical data in learning.

Adaptive Discretization Algorithms. Discretization-based approaches to standard multi-armed bandits and reinforcement learning have been explored both heuristically and theoretically in different settings. Adaptive discretization was first analyzed theoretically for the standard stochastic

continuous multi-armed bandit model, where Kleinberg et al. (2019) developed an algorithm which achieves instance-dependent regret scaling with respect to the so-called "zooming dimension" of the action space. This approach was later extended to contextual models in Slivkins (2011). In Elmachtoub et al. (2017) the authors improve on the practical performance and scalability by considering decision-tree instead of dyadic partitions of the action space. Similar techniques have been applied to reinforcement learning, where again existing works have studied the theoretical challenges of designing discretization-based approaches with instance-specific regret guarantees (Sinclair et al., 2021), and heuristic performance under different tree structures (Uther & Veloso, 1998; Pyeatt & Howe, 2001). However, none of these algorithms have been extensively studied within the concept of including historical data, or applied to the combinatorial bandit model, with the exception of Xu et al. (2021). Our work builds upon theirs through a novel method of incorporating historical data into an algorithm, and by additionally considering adaptive instead of fixed discretization.

Experience Replay in Reinforcement Learning. Lastly, we note that the ARTIFICIAL REPLAY approach has relations to experience replay in the reinforcement learning literature (Schaul et al., 2017; Mnih et al., 2013). In contrast to ARTIFICIAL REPLAY, which is designed to use *historical data* collected independently of the algorithm, experience replay uses online observations (i.e., datapoints in $\mathcal{H}_t$) and requires using *off-policy* estimation procedures to incorporate the information in learning.

C FULL PRELIMINARY DETAILS

In this section we restate and give further assumptions for the general stochastic bandit model described in Section 2.

C.1 K-ARMED BANDIT

The finite-armed bandit model can be viewed in this framework by considering $\mathcal{A} = [K] = \{1, \ldots, K\}$. This recovers the classical model from Lai & Robbins (1985); Auer et al. (2002).

C.2 COMBINATORIAL MULTI-ARMED BANDIT FOR CONTINUOUS RESOURCE ALLOCATION (CMAB-CRA)

A central planner has access to a metric space $\mathcal{S}$ of resources with metric $d_{\mathcal{S}}$. They are tasked with splitting a total amount of B divisible budget across N different resources within $\mathcal{S}$. For example, in a wildlife conservation domain, the space $\mathcal{S}$ can be considered as the protected area of a park, and the allocation budget corresponds to divisible effort, or proportion of rangers allocated to patrol in the chosen area. We denote the feasible space of allocations as $\mathcal{B}$ and define the feasible action space as follows:

$$\mathcal{A} = \left\{ (\vec{\mathbf{p}}, \vec{\beta}) \in \mathcal{S}^N \times \mathcal{B}^N \mid \sum_{i=1}^{N} \beta^{(i)} \leq B, \quad d_{\mathcal{S}}(\mathbf{p}^{(i)}, \mathbf{p}^{(j)}) \geq \epsilon \quad \forall i \neq j \right\}. \tag{7}$$

Note that we require the chosen action to satisfy the budgetary constraint (i.e. $\sum_i \beta^{(i)} \leq B$), and that the chosen resources are distinct (aka ϵ-away from each other according to $d_{\mathcal{S}}$).

Further, let $\Re : \mathcal{S} \times \mathcal{B} \to \Delta([0,1])$ be the *unknown* reward distribution over the resource and allocation space. The goal of the algorithm is to pick an action $A = (\vec{\mathbf{p}}, \vec{\beta}) \in \mathcal{A}$ in a way that maximizes $\sum_{i=1}^{N} \mathbb{E}\big[\Re(\mathbf{p}^{(i)}, \beta^{(i)})\big]$, the expected total mean reward accumulated from the resources subject to the budget constraints. Denoting $\mathbb{E}[\Re(\mathbf{p}, \beta)]$ as $\mu(\mathbf{p}, \beta)$, the optimization problem is formulated below:

$$\begin{aligned} \max_{\vec{\mathbf{p}}, \vec{\beta}} \ & \sum_{i=1}^{N} \mu(\mathbf{p}^{(i)}, \beta^{(i)}) \\ \text{s.t.} \ & \sum_{i} \beta^{(i)} \leq B \\ & d_{\mathcal{S}}(\mathbf{p}^{(i)}, \mathbf{p}^{(j)}) \geq \epsilon \quad \forall i \neq j \,. \end{aligned} \tag{8}$$

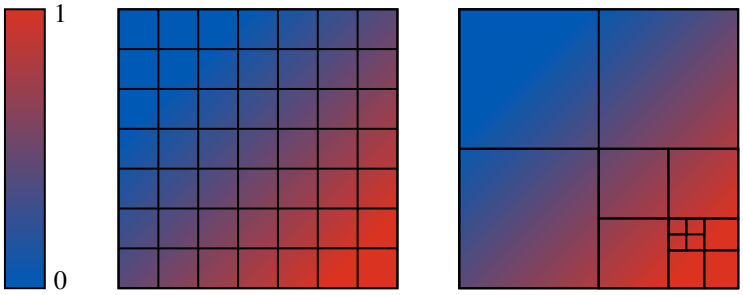

Figure 4: Comparison of a fixed (middle) and adaptive (right) discretization on a two-dimensional resource set $\mathcal{S}$ for a fixed allocation level β. The underlying color gradient corresponds to the mean reward $\mu(\mathbf{p}, \beta)$ with red corresponding to higher value and blue to lower value (see figure on left for legend). The fixed discretization algorithm is forced to *explore uniformly* across the entire resource space. In contrast, the adaptive discretization algorithm is able to maintain a *data efficient* representation, even without knowing the underlying mean reward function a priori.

Lastly, we consider the historical data $\mathcal{H}^{hist}$ to also be decomposed, in that each element is a particular $(\mathbf{p}, \beta, R)$ pair with $R \sim \Re(\mathbf{p}, \beta)$.

C.2.1 Assumptions

We make two assumptions on the underlying problem. The first is a standard nonparametric assumption, highlighting that the resource space $\mathcal{S}$ is a metric space and that the true underlying reward function μ is Lipschitz with respect to $d_{\mathcal{S}}$. This assumption is common in the continuous bandit literature; see Kleinberg et al. (2019) for more discussion.

Assumption 1. *$\mathcal{S}$ is a compact metric space endowed with a metric $d_{\mathcal{S}}$ with diameter d_{max}, and $\mathcal{B}$ is a discrete space, both known to the decision maker. We assume that the mean reward function $\mu(\mathbf{p}, \beta)$ is Lipschitz with respect to the metric $d_{\mathcal{S}}$ over $\mathbf{p}$ with known Lipschitz constant L.*

The next assumption assumes access to an oracle for solving optimization problems of the form of Eq. (8) for arbitrary choice of reward functions $r(\mathbf{p}, \beta)$. We can relax this assumption to instead assume that there exists a randomized approximation oracle by appropriately shifting the regret benchmark. However, in Section 5 we run experiments with exact solution oracles and omit this discussion from this work.

Assumption 2. *The optimization problem formulated in Eq.* (8) *can be solved for arbitrary reward functions $\mu(\mathbf{p}, \beta)$.*

C.2.2 Mapping to Green Security Domains

The CMAB-CRA model can be used to specify green security domains from Xu et al. (2021). $\mathcal{S}$ is used to represent the "protected region", or geographic region of the park, and $\mathcal{B}$ is the discrete set of potential patrol efforts to allocate, such as the number of ranger hours per week, with the total budget B being 40 hours. This formulation generalizes Xu et al. (2021) to a more realistic continuous space model of the landscape, instead of the artificial fixed discretization that was considered in prior work consisting of 1×1 sq. km regions on park).

D Discretization-Based Algorithms for CMAB-CRA

In this section we detail a fixed and adaptive discretization algorithm for CMAB-CRA which satisfy the IIData property. We start off by summarizing the main algorithm sketch, before highlighting the key details and differences between the two. For pseudocode of the adaptive discretization algorithm see **??**. We describe the algorithm without incorporating historical data (where its counterpart involving historical data can be used by treating this as the base algorithm Π and appealing to Artificial Replay or Full Start described in Section 3).

Our algorithms are Upper Confidence Bound (UCB) style as the selection rule maximizes Eq. (8) approximately over a discretization of $\mathcal{S}$. Both algorithms are parameterized by the time horizon T and a value $\delta \in (0, 1)$.

For each allocation $\beta \in \mathcal{B}$ the algorithm maintains a collection of regions $\mathcal{P}_t^\beta$ of $\mathcal{S}$. Each element $\mathcal{R} \in \mathcal{P}_t^\beta$ is a region with diameter $r(\mathcal{R})$. For the fixed discretization variant, $\mathcal{P}_t^\beta$ is fixed at the start of learning. In the adaptive discretization algorithm, this partitioning is refined over the course of learning in an *data-driven manner*.

For each time period t, the algorithm maintains three tables linear with respect to the number of regions in the partitions $\mathcal{P}_t^\beta$. For every region $\mathcal{R} \in \mathcal{P}_t^\beta$ we maintain an *upper confidence value* $\text{UCB}_t(\mathcal{R}, \beta)$ for the true $\mu(\mathcal{R}, \beta)$ value for points in $\mathcal{R}$ (which is initialized to be one), determined based on an estimated mean $\overline{\mu}_t(\mathcal{R}, \beta)$ and $n_t(\mathcal{R}, \beta)$ for the number of times $\mathcal{R}$ has been selected by the algorithm in timesteps up to t. The latter is incremented every time $\mathcal{R}$ is played, and is used to construct the bonus term. At a high level, our algorithms perform two steps in each iteration t: select an action via the *selection rule* and then *update parameters*. In addition, the adaptive discretization algorithm will *re-partition* the space. In order to define the steps, we first introduce some definitions and notation.

Let $t_{\mathcal{R}} = n_t(\mathcal{R}, \beta)$ be the number of times the algorithm has selected region $\mathcal{R} \in \mathcal{P}$ at allocation β by time t. The *confidence radius* (or bonus) of region $\mathcal{R}$ is defined:

$$b(t_{\mathcal{R}}) = 2\sqrt{\frac{2\log(T/\delta)}{t_{\mathcal{R}}}} \tag{9}$$

corresponding to the uncertainty in estimates due to stochastic nature of the rewards. Lastly, the UCB value for a region $\mathcal{R}$ is computed as $\text{UCB}_t(\mathcal{R}, \beta) = \min\{\text{UCB}_{t-1}(\mathcal{R}, \beta), \overline{\mu}_t(\mathcal{R}, \beta) + b(n_t(\mathcal{R}, \beta)\}$. This enforces monotonicity in the UCB estimates, similar to MonUCB, and is required for the IIData property to hold.

At each timestep t, the algorithm *selects regions* according to the following optimization procedure:

$$\begin{aligned}
\max_{z(\mathcal{R},\beta)\in\{0,1\}} & \sum_{\beta\in\mathcal{B}} \sum_{\mathcal{R}\in\mathcal{P}_t^\beta} \text{UCB}_t(\mathcal{R}, \beta) \cdot z(\mathcal{R}, \beta) \\
\text{s.t.} & \sum_{\beta\in\mathcal{B}} \sum_{\mathcal{R}\in\mathcal{P}_t^\beta} \beta \cdot z(\mathcal{R}, \beta) \leq B \\
& \sum_{\beta\in\mathcal{B}} \sum_{\mathcal{R}\in\mathcal{P}_t^\beta} z(\mathcal{R}, \beta) \leq N \\
& z(\mathcal{R}, \beta) + \sum_{\beta'\neq\beta} \sum_{\tilde{\mathcal{R}}\in\mathcal{P}_t^{\beta'}, \mathcal{R}\subset\tilde{\mathcal{R}}} z(\tilde{\mathcal{R}}, \beta') \leq 1 \quad \forall \beta, \mathcal{R} \in \mathcal{P}_t^\beta
\end{aligned} \tag{10}$$

The objective encodes the goal of maximizing the upper confidence bound terms. The first constraint encodes the budget limitation, and the second that at most N regions can be selected. The final constraint is a technical one, essentially requiring that for each region $\mathcal{R} \in \mathcal{S}$, the same region is not selected at different allocation amounts. In the supplementary code base we provide an efficient implementation which avoids this step by "merging" the trees appropriately so that each region contains a vector of estimates of $\overline{\mu}_t(\mathcal{R}, \beta)$ for each $\beta \in \mathcal{B}$. Based on the optimal solution, the final action A is taken by picking $(\mathbf{p}, \beta)$ for each $\mathcal{R}$ such that $\mathbf{p} \in \mathcal{R}$ and $z(\mathcal{R}, \beta) = 1$. We lastly note that this optimization problem is a well-known "knapsack" problem with efficient polynomial-time approximation guarantees. It also has a simple "greedy" solution scheme, which iteratively selects the regions with largest $\text{UCB}_t(\mathcal{R}, \beta)/\beta$ ratio (i.e. the so-called "bang-per-buck"). See Williamson & Shmoys (2011) for more discussion.

After subsequently observing the rewards for the selected regions, we increment $t_{\mathcal{R}} = n_t(\mathcal{R}, \beta)$ by one for each selected region, update $\overline{\mu}_t(\mathcal{R}, \beta)$ accordingly with the additional datapoint, and compute $\text{UCB}_t(\mathcal{R}, \beta)$. Then the two rules are defined as follows:

1. **Selection rule**: Greedily select at most N regions subject to the budgetary constraints which maximizes $\text{UCB}_t(\mathcal{R}, \beta)$ following Eq. (10).

Algorithm 3 Adaptive Discretization Algorithm (AdaMonUCB)

procedure Adaptive Discretization for CMAB-CRA($\mathcal{S}, \mathcal{B}, T, \delta$)
  Initiate $|\mathcal{B}|$ partitions $\mathcal{P}_1^\beta$ for each $\beta \in \mathcal{B}$, each containing a single region with radius $d_{\max}$ and $\overline{\mu}_1^\beta$ estimate equal to 1
  for each timestep $\{t \leftarrow 1, \ldots, T\}$ **do**
    Select the regions by the selection rule Eq. (10)
    For each selected region $\mathcal{R}$ (regions where $z(\mathcal{R}, \beta) = 1$), add (a, β) to A_t for any $a \in \mathcal{R}$
    Play action A_t in the environment
    Update parameters: $t = n_{t+1}(\mathcal{R}_{\text{sel}}, \beta) \leftarrow n_t(\mathcal{R}_{\text{sel}}, \beta) + 1$ for each selected region $\mathcal{R}_{\text{sel}}$ with $z(\mathcal{R}_{\text{sel}}, \beta) = 1$, and update $\overline{\mu}_t(\mathcal{R}_{\text{sel}}, \beta)$ accordingly with observed data

    if $n_{t+1}(\mathcal{R}, \beta) \geq \left(\frac{d_{\max}}{r(\mathcal{R})}\right)^2$ and $r(\mathcal{R}) \geq 2\epsilon$ **then** Split Region($\mathcal{R}, \beta, t$)
procedure Split Region($\mathcal{R}$, β, t)
  Set $\mathcal{R}_1, \ldots, \mathcal{R}_n$ to be an $\frac{1}{2}r(\mathcal{R})$-packing of $\mathcal{R}$, and add each region to the partition $\mathcal{P}_{t+1}^\beta$
  Initialize parameters $\overline{\mu}_t(\mathcal{R}_i, \beta)$ and $n_t(\mathcal{R}_i, \beta)$ for each new region $\mathcal{R}_i$ to inherent values from the parent region $\mathcal{R}$

2. **Update parameters**: For each of the selected regions $\mathcal{R}$, increment $n_t(\mathcal{R}, \beta)$ by 1, update $\overline{\mu}_t(\mathcal{R}, \beta)$ based on observed data, and update $\text{UCB}_t(\mathcal{R}, \beta)$ while ensuring monotonicity.

D.1 Fixed Discretization

This algorithm is additionally parameterized by a discretization level γ. The algorithm starts by maintaining a γ-covering of $\mathcal{S}$, which we denote as $\mathcal{P}_t^\beta = \mathcal{P}$ for all $t \in [T]$ and $\beta \in \mathcal{B}$.

D.2 Adaptive Discretization

For each effort level $\beta \in \mathcal{B}$ the algorithm maintains a collection of regions $\mathcal{P}_t^\beta$ of $\mathcal{S}$ which is refined over the course of learning for each timestep t. Initially, when $t = 1$, there is only one region in each partition $\mathcal{P}_1^\beta$ which has radius $d_{\max}$ containing $\mathcal{S}$.

The key differences from the fixed discretization are two-fold. First, the *confidence radius* or bonus of region $\mathcal{R}$ is defined via:

$$b(t) = 2\sqrt{\frac{2\log(T/\delta)}{t}} + \frac{2Ld_{\max}}{\sqrt{t}} \,.$$

The first term corresponds to uncertainty in estimates due to stochastic nature of the rewards, and the second is the discretization error by expanding estimates to all points in the region.

Second, after selecting an action and updating the estimates for the selected regions, the algorithm additionally decides whether to update the partition. This is done via:

3 **Re-partition the space**: Let $\mathcal{R}$ denote any selected ball and $r(\mathcal{R})$ denote its radius. We split when $r(\mathcal{R}) \geq 2\epsilon$ and $n_t(\mathcal{R}, \beta) \geq (d_{\max}/r(\mathcal{R}))^2$. We then cover $\mathcal{R}$ with new regions $\mathcal{R}_1, \ldots, \mathcal{R}_n$ which form an $\frac{1}{2}r(\mathcal{R})$-Net of $\mathcal{R}$. We call $\mathcal{R}$ the *parent* of these new balls and each child ball inherits all values from its parent. We then add the new balls $\mathcal{R}_1, \ldots, \mathcal{R}_n$ to $\mathcal{P}_t^\beta$ to form the partition for the next timestep $\mathcal{P}_{t+1}^\beta$.

Benefits of Adaptive Discretization. The fixed discretization algorithm cannot adapt to the underlying structure in the problem since the discretization is fixed prior to learning. This causes increased computational, storage, and sample complexity since each individual region must be explored in order to obtain optimal regret guarantees. Instead, our adaptive discretization algorithm adapts the discretization in a data-driven manner to reduce unnecessary exploration. The algorithms keep a fine discretization across important parts of the space, and a coarser discretization across unimportant regions. See Fig. 4 for a sample adaptive discretization observed, and Kleinberg et al. (2010); Sinclair et al. (2021) for more discussion on the benefits of adaptive discretization over fixed.

Implementation of Adaptive Discretization. In the attached code base we provide an efficient implementation of ADAMONUCB, the adaptive discretization algorithm for the CMAB-CRA domain. Pseudocode for ADAMONUCB is in Algorithm 3.

We represent the partition $\mathcal{P}_t^\beta$ as a tree with leaf nodes corresponding to *active balls* (i.e. ones which have not yet been split). Each node in the tree keeps track of $n_t(\mathcal{R}, \beta)$, $\overline{\mu}_t(\mathcal{R}, \beta)$, and $\text{UCB}_t(\mathcal{R}, \beta)$. While the partitioning works for any compact metric space, we implement it in $\mathcal{S} = [0, 1]^2$ under the infinity norm metric. With this metric, the high level implementation of the three steps is as follows:

- **Selection rule**: In this step we start by "merging" all of the trees for each allocation level β (i.e. $\mathcal{P}_t^\beta$ for $\beta \in \mathcal{B}$) into a single tree, with estimates $\text{UCB}_t(\mathcal{R}, \cdot)$ represented as a vector for each $\beta \in \mathcal{B}$ rather than a scalar. On this merged partition of $\mathcal{S}$ we solve Eq. (10) *over the leaves* to avoid modelling the constraint which ensures that each region is selected at a single allocation amount.
- **Update estimates**: Updating the estimates simply updates the stored $n_t(\mathcal{R}, \beta), \overline{\mu}_t(\mathcal{R}, \beta)$, and $\text{UCB}_t(\mathcal{R}, \beta)$ for each of the selected regions based on observed data.
- **Re-partition the space**: In order to split a region $\mathcal{R}$, we create four new subregions corresponding to splitting the two dimensions in half. For example, the region $[0, 1]^2$ will be decomposed to

  $$[0, \tfrac{1}{2}] \times [0, \tfrac{1}{2}] \quad [0, \tfrac{1}{2}] \times [\tfrac{1}{2}, 1] \quad [\tfrac{1}{2}, \tfrac{1}{2}] \times [0, \tfrac{1}{2}] \quad [\tfrac{1}{2}, \tfrac{1}{2}] \times [\tfrac{1}{2}, \tfrac{1}{2}] \,.$$

  We add on the children to the tree with links to its parent node, and initialize all estimates to that of its parent.

Lastly, we comment that in order to implement the FULL START and ARTIFICIAL REPLAY versions of the adaptive discretization algorithm we pre-processed the entire historical data into a tree structure. This allowed us to check whether the given action has historical data available by simply checking the corresponding node in the pre-processed historical data tree.

E EXPERIMENT DETAILS

We provide additional details about the experimental domains, algorithm implementation, and additional results. The additional results include experiments to evaluate the performance of ARTIFICIAL REPLAY on algorithms for combinatorial finite-armed bandit (Chen et al., 2013) as well as the standard K-armed bandit. For the later we include simulations with Thompson sampling (Russo et al., 2018), and information-directed sampling (Russo & Van Roy, 2018), which do not satisfy the IIData property, but still experience empirical improvements when using ARTIFICIAL REPLAY against FULL START.

E.1 DOMAIN DETAILS

E.1.1 FINITE K-ARMED BANDIT

For the K-armed bandit, we generate mean rewards for each arm $a \in [K]$ uniformly at random.

E.1.2 CMAB-CRA

Piecewise-linear. This synthetic domain has a piecewise-linear reward function to ensure that the greedy approximation solution is optimal, as discussed in Section 4.2. As a stylized setting, this reward function uses a very simple construction:

$$\mu(\mathbf{p}, \beta) = \beta \cdot \left(\frac{p_1}{2} + \frac{p_2}{2} \right). \tag{11}$$

We visualize this reward in Fig. 6. The optimal reward is at $(1, 1)$.

Quadratic. The quadratic environment is a synthetic domain with well-behaved polynomial reward function of the form:

$$\mu(\mathbf{p}, \beta) = \beta\big(1 - (p_1 - 0.5)^2 + (p_2 - 0.5)^2\big) \tag{12}$$

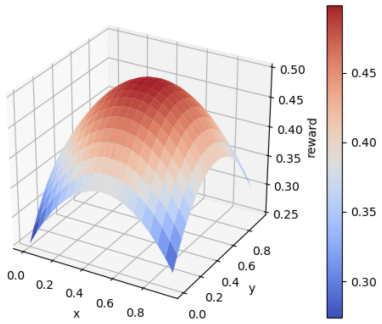

Figure 5: Reward function for the quadratic environment.

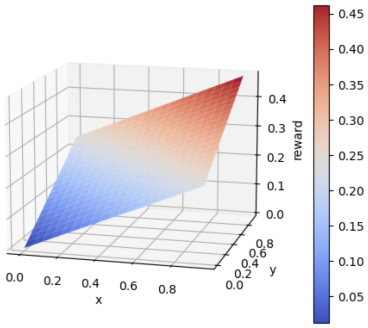

Figure 6: Reward function for the piecewise-linear environment.

which we visualize in Fig. 5). The optimal reward at is achieved at $(0.5, 0.5)$.

Green Security Domain. For the green security domain, we wish to predict the probability that poachers place snares throughout a large protected area using ranger patrol observations. We use real-world historical patrol data from Murchison Falls National Park. The historical data are continuous-valued GPS coordinates (longitude, latitude) marking trajectories with locations automatically recorded every 30 minutes. Between the years 2015 and 2017, we have 180,677 unique GPS waypoints.

We normalize the space of the park boundary to the range $[0, 1]$ for both dimensions. For each point $\mathbf{p} \in [0, 1]^2$ in this historical data, we compute "effort" or allocation by calculating straight-line *trajectories* between the individual waypoints to compute the distance patrolled, allocating to each waypoint one half the sum of the line segments to which it is connected. We then associate with each point a binary label $\{0, 1\}$ representing the observation. Direct observations of poaching are rather rare, so to overcome strong class imbalance for the purposes of these experiments, we augment the set of instances we consider a positive label to include any human-related or wildlife observation. Everything else (e.g., position waypoint) gets a negative label.

To generate a continuous-action reward function, we build a neural network to learn the reward function across the park and use that as a simulator for reward. The neural network takes three inputs, $a = (point_1, point_2, \beta)$, and outputs a value $[0, 1]$ to indicate probability of an observation. This probability represents $\mu(a)$ for the given point and allocation.

E.2 Adaptive Discretization

We offer a visual demonstration of the adaptive discretization process in Fig. 7, using 10,000 real samples of historical patrol observations from Murchison Falls National Park. This discretization is used to iteratively build the dataset tree used to initialize the Full Start algorithm with adaptive discretization.

E.3 Additional Experimental Results

In Fig. 8 we evaluate the performance of Artificial Replay compared to Full Start and Ignorant using two multi-armed bandit algorithms that do not have the IIData property, Thompson Sampling (TS) and Information-Directed Sampling (IDS). Although our theoretical regret guarantees do not apply to Thompson sampling or IDS as base algorithms, these results demonstrate that empirically Artificial Replay still performs remarkably well, matching the performance of Full Start with Thompson sampling and avoiding the exploding regret that Full Start suffers with IDS.

We note that with imbalanced data, Full Start is converging on a suboptimal action with more historical data. This is because the IDS algorithm, when warm-started with historical data, maintains a near-'zero entropy' posterior distribution over a sub-optimal action which is over-represented in the historical dataset. Since the selection procedure takes the action that maximizes expected re-

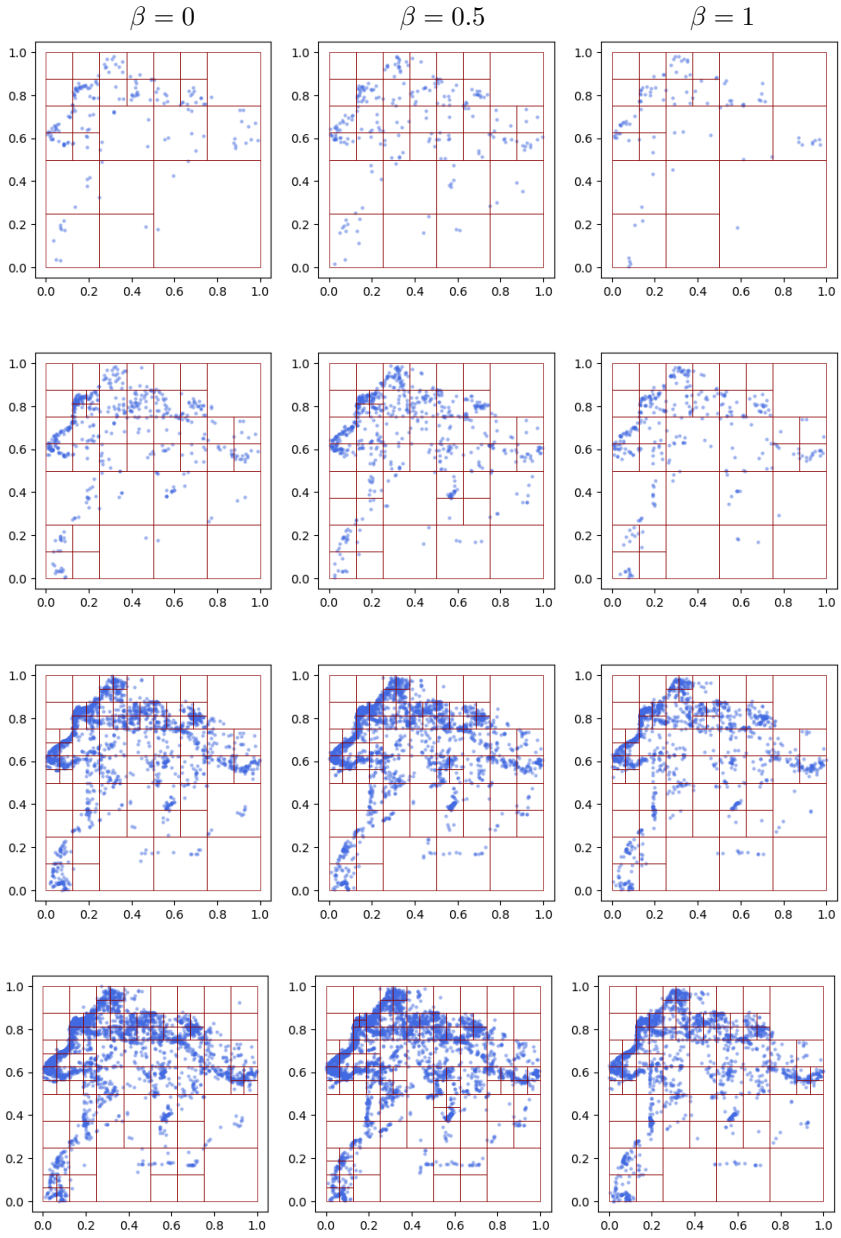

Figure 7: Adaptive discretization in the $\mathcal{S} = [0, 1]^2$ space using 10,000 samples of real historical patrol observations from Murchison Falls National Park. Each row depicts the distribution of historical samples and the space partition after 500, 1,500, 6,000, and 10,000 samples are added to the dataset tree. Each column visualizes the dataset tree for each of three levels of effort $\beta \in \mathcal{B} = \{0, 0.5, 1\}$. As shown, these real-world historical samples exhibit strong *imbalanced data coverage*, leading to significantly fine discretizations in areas with many samples and very coarse discretization in other regions.

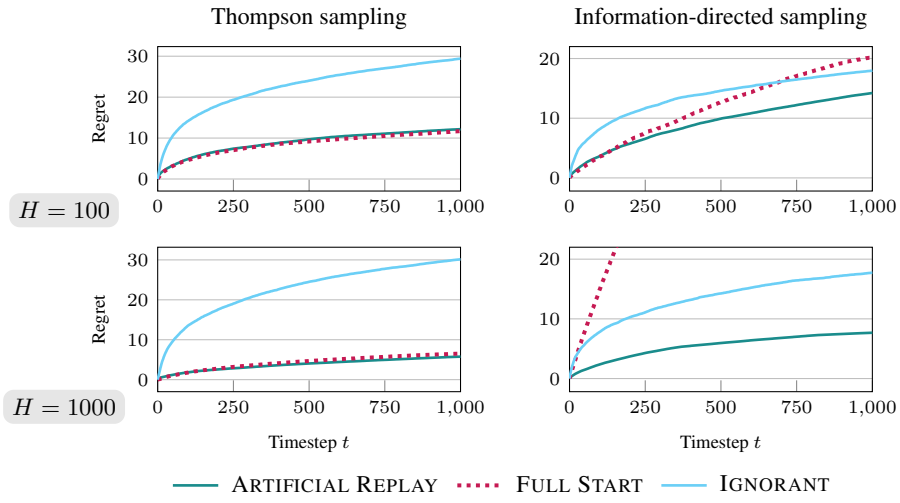

Figure 8: Cumulative regret (y-axis; lower is better) across time $t \in [T]$. ARTIFICIAL REPLAY performs competitively across all domain settings, with both Thompson sampling (Russo et al., 2018) (left) and information-directed sampling (Russo & Van Roy, 2018) (right). In FULL START applied to information-directed sampling with $H = 1000$ the algorithm converges on a sub-optimal arm since its posterior variance is low (due to more data), resulting in poor regret performance due to *spurious data*.

turn divided by posterior variance, the algorithm continuously picks this sub-optimal action at each timestep.

In Fig. 9 we consider the combinatorial bandit setting with a set of $K = 10$ discrete arms and a budget $B = 3$ over $T = 1{,}000$ timesteps. This setting is similar to Fig. 2 but instead here we consider a combinatorial setting (where multiple arms can be pulled at each timestep) rather than a standard stochastic K-armed bandit. Across different values of H, ARTIFICIAL REPLAY matches the performance of FULL START (Fig. 9(a)) despite using an increasing smaller fraction of the historical dataset $\mathcal{H}^{hist}$ (Fig. 9(b)). The regret plot in Fig. 9(c) shows that the regret of our method is coupled with that of FULL START across time. Fig. 9(d) tracks the number of samples from history $\mathcal{H}^{hist}$, with $H = 1{,}000$, used over time: ARTIFICIAL REPLAY uses 471 historical samples before taking its first online action. The number of historical samples used increases at a decreasing rate and ends with using 740 samples.

E.4 EXPERIMENT EXECUTION

Each experiment was run with 60 iterations where the relevant plots are taking the mean of the related quantities. All randomness is dictated by a seed set at the start of each simulation for verifying results. The experiments were conducted on a personal laptop with a 2.4 GHz Quad-Core Intel Core i5 processor and 16 GB of RAM.

F OMITTED PROOFS

F.1 SECTION 3 PROOFS

Proof of Theorem 3.2. We start off by showing that $\pi_t^{\text{ARTIFICIAL REPLAY}(\Pi)} \overset{d}{=} \pi^{\text{FULL START}(\Pi)}$ using the reward stack model for a stochastic bandit instance introduced in Lattimore & Szepesvári (2020). Due to the fact that the observed rewards are independent (both across actions but also across timesteps), consider a sample path where $(R_{a,t})_{a \in \mathcal{A}, t \in [T]}$ are pre-sampled according to $\Re(a)$. Upon pulling arm a in timestep t, the algorithm is given feedback $R_{a,t}$.

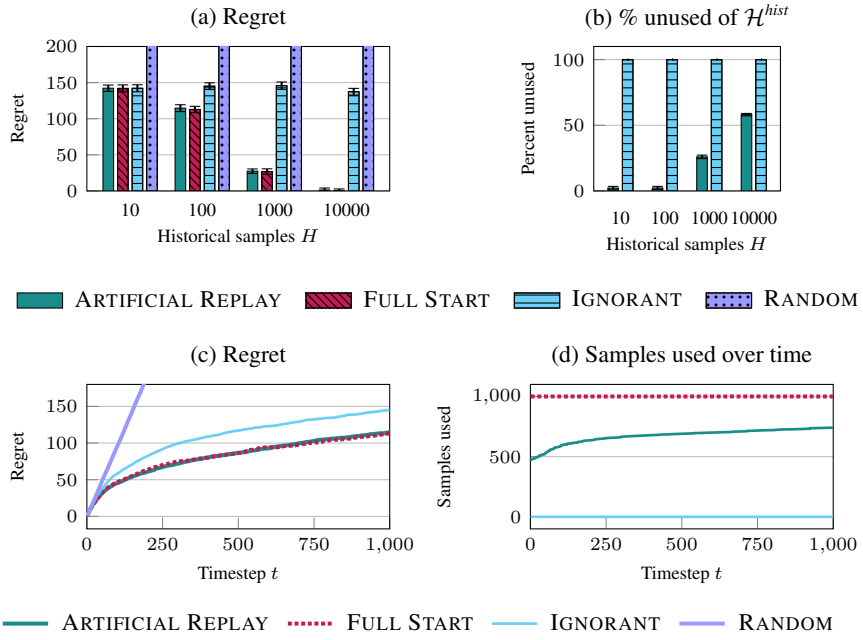

Figure 9: We consider a combinatorial bandit setting with finite actions: $K = 10$ arms, $B = 3$ budget, and horizon $T = 1{,}000$. Increasing the number of historical samples H leads FULL START to use unnecessary data, particularly as H gets very large. ARTIFICIAL REPLAY achieves equal performance in terms of regret (plot a) while using less than half the historical data (plot b). In (plot c) we see that with $H = 1{,}000$ historical samples, ARTIFICIAL REPLAY uses 471 historical samples before taking its first online action. The number of historical samples used increases at a decreasing rate, using 740 total samples by the horizon T.

It is important to note that the resulting probability space generated in the reward stack model is identical in distribution to any sequence of histories observed by running a particular algorithm. More specifically, it preserves the following two properties:

(a) The conditional distribution of the action A_t given the sequence $(A_1, R_{A_1,1}), \ldots, (A_{t-1}, R_{A_{t-1},t-1})$ is $\pi_t(\cdot \mid \mathcal{H}_t)$ almost surely.

(b) The conditional distribution of the reward R_t is $\Re(A_t)$ almost surely.

Based on this reward stack model, we show by induction on t that $\pi_t^{\text{ARTIFICIAL REPLAY}(\Pi)} = \pi_t^{\text{FULL START}(\Pi)}$. Since this is true on an independent sample path, it results in a probabilistic coupling between the two algorithms to have the same distribution.

Base Case: $t = 1$.

By definition of $\pi^{\text{FULL START}(\Pi)}$ we know that

$$\pi_1^{\text{FULL START}(\Pi)} = \Pi(\mathcal{H}^{hist}) \, .$$

However, consider $\pi^{\text{ARTIFICIAL REPLAY}(\Pi)}$. The ARTIFICIAL REPLAY meta-algorithm will keep selecting actions until it creates a dataset $\mathcal{H}_1^{on} \subset \mathcal{H}^{hist}$ such that $\Pi(\mathcal{H}_1^{on})$ has no more unused samples in $\mathcal{H}^{hist}$. Denoting $A_1 = \Pi(\mathcal{H}_1^{on})$, the unused samples $\mathcal{H}^{hist} \setminus \mathcal{H}_1^{on}$ contains no data on A_1. As a result, by the independence of irrelevant data property for Π we have that $\Pi(\mathcal{H}_1^{on}) = \Pi(\mathcal{H}^{hist})$ and so $\pi_1^{\text{FULL START}(\Pi)} = \pi_1^{\text{ARTIFICIAL REPLAY}(\Pi)}$. Note that this shows that the observed data for the algorithms $\mathcal{H}_2$ are also identical (due to the reward stack model) via $\mathcal{H}_2 = \{A_1, R_{A_1,1}\}$.

Step Case: $t - 1 \to t$.

Since we know that $\pi_\tau^{\text{FULL START}(\Pi)} = \pi_\tau^{\text{ARTIFICIAL REPLAY}(\Pi)}$ for $\tau < t$, both algorithms have access to the same set of observed online data $\mathcal{H}_t$. By definition of $\pi^{\text{FULL START}(\Pi)}$:

$$\pi_t^{\text{FULL START}(\Pi)} = \Pi(\mathcal{H}^{hist} \cup \mathcal{H}_t) \, .$$

However, the ARTIFICIAL REPLAY algorithm continues to use offline samples until it generates a subset $\mathcal{H}_t^{on} \subset \mathcal{H}^{hist}$ such that $\Pi(\mathcal{H}_t^{on} \cup \mathcal{H}_t)$ has no further samples in $\mathcal{H}^{hist}$. Hence, by the independence of irrelevant data property again:

$$\Pi(\mathcal{H}^{hist} \cup \mathcal{H}_t) = \Pi(\mathcal{H}_t^{on} \cup \mathcal{H}_t) \, ,$$

and so $\pi_t^{\text{FULL START}(\Pi)} = \pi_t^{\text{ARTIFICIAL REPLAY}(\Pi)}$. Again we additionally have that $\mathcal{H}_{t+1} = \mathcal{H}_t \cup \{(A_t, R_{A_t,t})\}$ are identical for both algorithms.

Together this shows that $\pi^{\text{FULL START}(\Pi)} \stackrel{d}{=} \pi^{\text{ARTIFICIAL REPLAY}(\Pi)}$. Lastly we note that the definition of regret is $\text{REGRET}(T, \pi, \mathcal{H}^{hist}) = T \cdot \text{OPT} - \sum_{t=1}^{T} \mu(A_t)$ where A_t is sampled from π. Hence the previous policy-based coupling implies that $\text{REGRET}(T, \pi^{\text{ARTIFICIAL REPLAY}(\Pi)}, \mathcal{H}^{hist}) \stackrel{d}{=} \text{REGRET}(T, \pi^{\text{FULL START}(\Pi)}, \mathcal{H}^{hist})$ as well. □

F.2 SECTION 4 PROOFS

Proof of Theorem 4.1. Suppose that the base algorithm Π is MONUCB, and let $\mathcal{H}$ be an arbitrary dataset. Using the dataset, MONUCB will construct upper confidence bound values $\text{UCB}(a)$ for each action $a \in [K]$. The resulting policy is to pick the action $\Pi(\mathcal{H}) = \arg\max_{a \in [K]} \text{UCB}(a)$. Let $A_{\mathcal{H}}$ be the action which maximizes the $\text{UCB}(a)$ value.

Additionally, let $\mathcal{H}'$ be an arbitrary dataset containing observations from actions other than $A_{\mathcal{H}}$. Based on the enforced monotonicty of the indices from MONUCB, any $a \in [K]$ with $a \neq A_{\mathcal{H}}$ will have its constructed $\text{UCB}(a)$ no larger than its original one constructed with only using the dataset $\mathcal{H}$. Moreover, $\text{UCB}(A_{\mathcal{H}})$ will be unchanged since the additional data $\mathcal{H}'$ does not contain any information on $A_{\mathcal{H}}$. As a result, the policy will take $\Pi(\mathcal{H} \cup \mathcal{H}') = A_{\mathcal{H}}$ since it will still maximize the $\text{UCB}(a)$ index.

Next we provide a regret analysis for IGNORANT(MONUCB). We assume without loss of generality that there is a unique action a^* which maximizes $\mu(a)$. We let $\Delta(a) = \mu(a^*) - \mu(a)$ be the sub-optimality gap for any other action a. To show the regret bound we follow the standard regret decomposition, which notes that $\mathbb{E}\left[\text{REGRET}(T, \pi, \mathcal{H}^{hist})\right] = \sum_a \Delta(a)\mathbb{E}[n_T(a)]$. To this end, we start off with the following Lemma, giving a bound on the expected number of pulls of MONUCB in the "ignorant" setting (i.e., without incorporating any historical data).

Lemma F.1. *The expected number of pulls for any sub-optimal action a of* MONUCB *satisfies*

$$\mathbb{E}[n_T(a)] \leq \frac{2K}{T} + \frac{8\log(T)}{\Delta(a)^2}$$

Proof of Lemma F.1. Denote by $S_{a,\tau}$ to be the empirical sum of τ samples from action a. Note that via an application of Hoeffding's inequality:

$$\mathbb{P}\left(\left|\mu(a) - \frac{S_{a,\tau}}{\tau}\right| \geq \sqrt{\frac{2\log(T)}{\tau}}\right) \leq \frac{2}{T^4}\,.$$

A straightforward union bound with this fact shows that the following event occurs with probability at least $1 - 2K/T^2$:

$$\mathcal{E} = \left\{\forall a \in [K], 1 \leq k \leq T, |\mu(a) - S_{a,k}/k| \leq \sqrt{\frac{2\log(T)}{k}}\right\}.$$

Now consider an arbitrary action $a \neq a^*$. We start by showing that on the event $\mathcal{E}$ that $n_t(a) \leq 4\log(T)/\Delta(a)^2$. If action a was taken over a^* at some timestep t then:

$$\text{UCB}_t(a) > \text{UCB}_t(a^*)\,.$$

However, using the reward stack model and the definition of $\text{UCB}_t(a)$ we know that

$$\begin{aligned}\text{UCB}_t(a) &= \min_{\tau \leq t} \frac{S_{a,n_\tau(a)}}{n_\tau(a)} + \sqrt{\frac{2\log(T)}{n_\tau(a)}} \quad \text{by definition of monotone UCB} \\ &\leq \frac{S_{a,n_t(a)}}{n_t(a)} + \sqrt{\frac{2\log(T)}{n_t(a)}}.\end{aligned}$$

Moreover, under the good event $\mathcal{E}$ we know that $\mu(a^*) \leq \text{UCB}_t(a^*)$ and that

$$\mu(a) \geq \frac{S_{a,n_t(a)}}{n_t(a)} - \sqrt{\frac{2\log(T)}{n_t(a)}}\,.$$

Combining this together gives

$$\mu(a^*) \leq \mu(a) + 2\sqrt{\frac{2\log(T)}{n_t(a)}}\,.$$

Rearranging this inequality gives that $n_t(a) \leq 8\log(T)/\Delta(a)^2$. Lastly, the final bound comes from the law of total probability and the bound on $\mathbb{P}(\mathcal{E})$. □

Lastly, Lemma F.1 can be used to obtain a bound on $\text{REGRET}(T, \pi^{\text{MONUCB}}, \mathcal{H}^{hist}) = O(\sum_a \log(T)/\Delta(a))$ using the previous regret decomposition, recovering the regret bound of standard UCB. □

In order to show Theorems 4.2 and 4.3 we start by showing a lemma similar to Lemma F.1 but for FULL START(MONUCB).

Lemma F.2. *Let H_a be the number of datapoints in $\mathcal{H}^{hist}$ for an action $a \in [K]$. The expected number of pulls for any sub-optimal action a of* FULL START *(*MONUCB*) satisfies*

$$\mathbb{E}[n_T(a)] \leq \frac{2K}{T} + \max\left\{0, \frac{8\log(T)}{\Delta(a)^2} - H_a\right\}.$$

Proof of Lemma F.2. We replicate the proof of Lemma F.1 but additionally add H_a samples to the estimates of each action $a \in [K]$. Denote by $S_{a,\tau}$ the empirical sum of $\tau + H_a$ samples from action a. Note that via an application of Hoeffding's inequality:

$$\mathbb{P}\left(\left|\mu(a) - \frac{S_{a,\tau}}{\tau + H_a}\right| \geq \sqrt{\frac{2\log(T)}{\tau + H_a}}\right) \leq \frac{2}{T^4} .$$

A straightforward union bound with this fact shows that the following event occurs with probability at least $1 - 2K/T^2$:

$$\mathcal{E} = \left\{\forall a \in [K], 1 \leq k \leq T, |\mu(a) - S_{a,k}/(k + H_a)| \leq \sqrt{\frac{2\log(T)}{k + H_a}}\right\} .$$

Now consider an arbitrary action $a \neq a^*$. If action a was taken over a^* at some timestep t then:

$$\text{UCB}_t(a) > \text{UCB}_t(a^*) .$$

By definition of the $\text{UCB}_t(a)$ term we know

$$\begin{aligned}
\text{UCB}_t(a) &= \min_{\tau \leq t} \frac{S_{a,n_\tau(a)}}{n_\tau(a) + H_a} + \sqrt{\frac{2\log(T)}{n_\tau(a) + H_a}} && \text{(by definition of monotone UCB)} \\
&\leq \frac{S_{a,n_t(a)}}{n_t(a) + H_a} + \sqrt{\frac{2\log(T)}{n_t(a) + H_a}} .
\end{aligned}$$

Moreover, under the good event $\mathcal{E}$ we know that $\mu(a^*) \leq \text{UCB}_t(a^*)$ and that

$$\mu(a) \geq \frac{S_{a,n_t(a)}}{n_t(a) + H_a} - \sqrt{\frac{2\log(T)}{n_t(a) + H_a}} .$$

Combining this together gives

$$\mu(a^*) \leq \mu(a) + 2\sqrt{\frac{2\log(T)}{n_t(a) + H_a}} .$$

Rearranging this inequality gives that $n_t(a) \leq 8\log(T)/\Delta(a)^2 - H_a$. Lastly, the final bound comes from the law of total probability and the bound on $\mathbb{P}(\mathcal{E})$. □

Using the previous two lemmas we are able to show Theorems 4.2 and 4.3.

Proof of Theorem 4.2. Without loss of generality we consider a setting with $K = 2$. Let a^* denote the optimal arm and a the other arm. Consider the historical dataset as follows: $\mathcal{H}^{hist} = (a, R_j^{\mathcal{H}})_{j \in [H]}$ where each $R_j^{\mathcal{H}} \sim \Re(a)$. By definition of FULL START(MONUCB) we know that the time complexity of the algorithm is at least $T + H$ since the algorithm will process the entire historical dataset.

In contrast, ARTIFICIAL REPLAY(MONUCB) will stop playing action a after $O(\log(T)/\Delta(a)^2)$ timesteps via Lemma F.1. Hence the time complexity of ARTIFICIAL REPLAY(MONUCB) can be upper bound by $O(T + \log(T))$. □

Proof of Theorem 4.3. First via Theorem 3.2 we note that in order to analyze $\pi^{\text{ARTIFICIAL REPLAY(MONUCB)}}$ it suffices to consider $\pi^{\text{FULL START(MONUCB)}}$. Using Lemma F.2 and a standard regret decomposition we get that:

$$\begin{aligned}
\mathbb{E}\Big[\text{REGRET}(T, \pi^{\text{FULL START(MONUCB)}}, \mathcal{H}^{hist})\Big] &= \sum_{a \neq a^*} \Delta(a)\mathbb{E}[n_T(a)] \quad \text{(by regret decomposition)} \\
&\leq \sum_{a \neq a^*} \Delta(a) O\left(\max\left\{0, \frac{\log(T)}{\Delta(a)^2} - H_a\right\}\right) \quad \text{(by Lemma F.2)} \\
&= \sum_{a \neq a^*} O\left(\max\left\{0, \frac{\log(T)}{\Delta(a)} - H_a\Delta(a)\right\}\right) . \qquad \square
\end{aligned}$$

Proof of Theorem 4.4. Suppose that Π is discretization based, uses monotone confidence estimates, and is using a greedy approximation solution to solving Eq. (8).

Using the dataset, the discretization based algorithm will construct upper confidence bound values $\mathrm{UCB}(\mathcal{R}, \beta)$ for each $\mathcal{R}$ in some discretization of the space. The resulting policy is to pick the action which solves an optimization problem of the form:

$$\begin{aligned} \max_{z} & \sum_{\beta \in \mathcal{B}} \sum_{\mathcal{R} \in \mathcal{P}_t^\beta} \mathrm{UCB}(\mathcal{R}, \beta) z(\mathcal{R}, \beta) \\ \text{s.t.} & \sum_{\beta \in \mathcal{B}} \sum_{\mathcal{R} \in \mathcal{P}_t^\beta} \beta z(\mathcal{R}, \beta) \leq B \\ & \sum_{\beta \in \mathcal{B}} \sum_{\mathcal{R} \in \mathcal{P}_t^\beta} z(\mathcal{R}, \beta) \leq N \\ & z(\mathcal{R}, \beta) + \sum_{\beta' \neq \beta} \sum_{\tilde{\mathcal{R}} \in \mathcal{P}_t^{\beta'}, \mathcal{R} \subset \tilde{\mathcal{R}}} z(\tilde{\mathcal{R}}, \beta') \leq 1 \quad \forall \beta, \mathcal{R} \in \mathcal{P}_t^\beta \end{aligned} \tag{13}$$

where $\mathcal{P}_t^\beta$ is a fixed discretization of the space $\mathcal{S}$, and $\mathrm{UCB}(\mathcal{R}, \beta)$ is constructed based on $\mathcal{H}$. Let $A_{\mathcal{H}}$ be the resulting combinatorial action from solving the optimization problem.

Additionally, let $\mathcal{H}'$ be an arbitrary dataset containing observations from actions other than the subarms in $A_{\mathcal{H}}$. Based on the enforced monotonicty of the indices, any subarm which is not selected will have its constructed $\mathrm{UCB}(\mathcal{R}, \beta)$ no larger than its original one constructed with only using the dataset $\mathcal{H}$. Moreover, the indexes of the chosen regions will be unchanged since $\mathcal{H}'$ contains no information on them. Hence, since Π is solving the expression using a greedy selection rule we know that the resulting action is the same, since the selected regions were unchanged and the UCB values are monotone decreasing with respect to the historical dataset. □