Dual-Mandate Patrols: Multi-Armed Bandits for Green Security

Lily Xu,[1] **Elizabeth Bondi,**[1] **Fei Fang,**[2] **Andrew Perrault,**[1] **Kai Wang,**[1] **Milind Tambe**[1]
[1] Harvard University
[2] Carnegie Mellon University
{lily_xu, ebondi, aperrault, kaiwang}@g.harvard.edu, feif@cs.cmu.edu, milind_tambe@harvard.edu

Abstract

Conservation efforts in green security domains to protect wildlife and forests are constrained by the limited availability of defenders (i.e., patrollers), who must patrol vast areas to protect from attackers (e.g., poachers or illegal loggers). Defenders must choose how much time to spend in each region of the protected area, balancing exploration of infrequently visited regions and exploitation of known hotspots. We formulate the problem as a stochastic multi-armed bandit, where each action represents a patrol strategy, enabling us to guarantee the rate of convergence of the patrolling policy. However, a naive bandit approach would compromise short-term performance for long-term optimality, resulting in animals poached and forests destroyed. To speed up performance, we leverage smoothness in the reward function and decomposability of actions. We show a synergy between Lipschitz-continuity and decomposition as each aids the convergence of the other. In doing so, we bridge the gap between combinatorial and Lipschitz bandits, presenting a no-regret approach that tightens existing guarantees while optimizing for short-term performance. We demonstrate that our algorithm, LIZARD, improves performance on real-world poaching data from Cambodia.

1 Introduction

Green security efforts to protect wildlife, forests, and fisheries require defenders (patrollers) to conduct patrols across protected areas to guard against attacks (illegal activities) (Lober 1992). For example, to prevent poaching, rangers conduct patrols to remove snares laid out to trap animals (Fig. 1). Green security games have been proposed to apply Stackelberg security games, a game-theoretic model of the repeated interaction between a defender and an attacker, to domains such as the prevention of illegal logging, poaching, or overfishing (Fang, Stone, and Tambe 2015). A growing body of work develops scalable algorithms for Stackelberg games (Basilico, Gatti, and Amigoni 2009; Blum, Haghtalab, and Procaccia 2014). Subsequent work has focused more on applicability to the real-world, employing machine learning to predict attack hotspots then using game-theoretic planning to design patrols (Nguyen et al. 2016; Gholami et al. 2018).

Figure 1: Rangers searching for snares (right) near a waterhole (left) in Srepok Wildlife Sanctuary in Cambodia. The waterhole is frequented by deer, pig, and bison, which are targeted by poachers.

However, many protected areas lack adequate and unbiased past patrol data, disabling us from learning a reasonable adversary model in the first place (Moreto and Lemieux 2015). As one of many examples, Bajo Madidi in the Bolivian Amazon was newly designated as a national park in 2019 (Berton 2020). The park is plagued with illegal logging, and patrollers do not have historical data from which to make predictions. The defenders do not want to spend patrol effort solely on information gathering; they must simultaneously maximize detection of attacks. As green security efforts get deployed on an ever-larger scale in hundreds of protected areas around the world (Xu et al. 2020), addressing this information-gathering challenge becomes crucial.

Motivated by these practical needs, we focus on conducting *dual-mandate patrols*[1], with the goal of simultaneously detecting illegal activities and collecting valuable data to improve our predictive model, achieving higher long-term reward. The key challenge with dual-mandate patrols is the exploration–exploitation tradeoff: whether to follow the best patrol strategy indicated by historical data or conduct new patrols to get a better understanding of the attackers. Some recent work proposes using multi-armed bandits to formulate the problem (Xu, Tran-Thanh, and Jennings 2016; Gholami et al. 2019). Despite their advances, we show that these approaches require unrealistically long time horizons to achieve good performance. In the real world, these initial losses are less tolerable and can lead to stakeholders aban-

[1]Code available at https://github.com/lily-x/dual-mandate.

doning such patrol-assistance systems. As we are designing this system for future deployment, it is critical to account for these practical constraints.

In this paper, we address real-world characteristics of green security domains to design dual-mandate patrols, prioritizing strong performance in the short term as well as long term. Concretely, we introduce LIZARD, a bandit algorithm that accounts for (i) decomposability of the reward function, (ii) smoothness of the decomposed reward function across features, (iii) monotonicity of rewards as patrollers exert more effort, and (iv) availability of historical data. LIZARD leverages both decomposability and Lipschitz continuity simultaneously, *bridging the gap between combinatorial and Lipschitz bandits*. We prove that LIZARD achieves no-regret when adaptively discretizing the metric space, generalizing results from both Chen et al. (2016) and Kleinberg, Slivkins, and Upfal (2019). Specifically, we improve upon the regret bound of Lipschitz bandits for non-trivial cases with more than one dimension and extend combinatorial bandits to continuous spaces. Additionally, we show that LIZARD dramatically outperforms existing algorithms on real poaching data from Cambodia.

2 Background

Green security domains have been extensively modeled as green security games (Haskell et al. 2014; Fang et al. 2016; Mc Carthy et al. 2016; Kamra et al. 2018). In these games, resource-constrained defenders protect large areas (e.g., forests, savannas, wetlands) from adversaries who repeatedly attack them. Most work has focused on learning attacker behavior models from historical data (Nguyen et al. 2016; Gholami et al. 2018) and using these models to plan patrols with limited lookahead to prevent future attacks (Fang, Stone, and Tambe 2015; Xu et al. 2017).

Despite successful deployment in some conservation areas (Xu et al. 2020), researchers have recognized that it is not always possible to have abundant historical data when first deploying a green security algorithm. Fig. 2 demonstrates the shortcomings of a pure exploitation approach. Using a simulator built from real-world poaching data, we observe a large shortfall in reward unless an unrealistically large amount of historical data is collected. Naively exploiting historical data compromises reward unless a very large amount of data is collected. The gap between the orange and dashed green lines reveals the opportunity cost (regret) of acting solely based off historical data.

Some recent work has begun to consider the need to also explore. Xu, Tran-Thanh, and Jennings (2016) follow an online learning paradigm, modeling the defender actions as pulling a set of arms (targets to patrol) at each timestep against an attacker who adversarially sets the rewards at each target. The model is then solved using FPL-UE, a variant of Follow the Perturbed Leader (FPL) algorithm (Kalai and Vempala 2005). Gholami et al. (2019) propose a hybrid model, MINION, that employs the FPL algorithm to choose between two meta arms representing FPL-UE and a supervised learning model based on historical data. However, both papers ignore domain structure such as feature similarity between targets and do not incorporate existing historical data into the online learners.

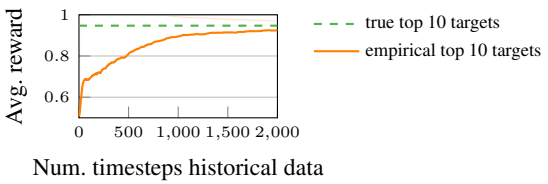

Figure 2: Naively protecting 10 out of 100 potential poaching targets based on our predictions would require 6 years of data to accurately protect the most important 10 targets.

There is a plethora of work on multi-armed bandits (MABs) (Bubeck and Cesa-Bianchi 2012). For brevity, we focus on confidence bound–based algorithms; other common methods include epsilon-greedy and Thompson sampling (Lattimore and Szepesvári 2018). For stochastic MAB with finite arms, the seminal UCB1 algorithm (Auer, Cesa-Bianchi, and Fischer 2002) always chooses the arm with the highest upper confidence bound and achieves no-regret, i.e., the expected reward is sublinearly close to always pulling the best arm in hindsight. For continuous arms, we must impose assumptions on the payoff function for tractability. Lipschitz continuity bounds the rate of change of a function μ, requiring that $|\mu(x) - \mu(y)| \leq L \cdot \mathcal{D}(x, y)$ for two points x, y and distance function $\mathcal{D}$. We refer to L as the Lipschitz constant. Kleinberg, Slivkins, and Upfal (2019) propose the *zooming algorithm* for Lipschitz MAB, which extends UCB1 with an adaptive refinement step to place more arms in high-reward areas of the continuous space. A parallel line of work applies Lipschitz-continuity assumptions to bandits to generate tighter confidence bounds (Bubeck et al. 2011; Magureanu, Combes, and Proutiere 2014). Chen et al. (2016) extend UCB1 to the combinatorial case with the *CUCB algorithm*, which tracks individual payoffs separately and uses an oracle to select the best combination. However, CUCB does not extend to continuous arms. Our proposed algorithm, LIZARD, outperforms both the zooming and CUCB algorithms by explicitly handling arms that are both Lipschitz-continuous and combinatorial.

Wildlife decline due to poaching demands fast action, so we cannot afford the infinite-time horizon usually considered by bandit algorithms. The case of short-horizon MABs is not well-studied. Some papers of this type (Gray, Zhu, and Ontañón 2020) focus on adjusting exploration patterns rather than not address the types of challenges induced by smoothness and combinatorial rewards, as are present in the domain we consider.

3 Problem Description

The protected area for which we must plan patrols is discretized into N targets, each associated with a feature vector $\vec{y}_i \in \mathbb{R}^K$ which is static across the time horizon T. In the poaching prevention domain, for example, the K features include geospatial characteristics such as distance to river, forest cover, animal density, and slope. We consider each

timestep to be a short period of patrolling, e.g., a day, and a time horizon that reflects our foreseeable horizon for patrol planning, e.g., 1.5 years, corresponding to 500 timesteps.

In each round, the defender determines an *effort vector* $\vec{\beta} = (\beta_1, \ldots, \beta_N)$ which specifies the amount of effort to spend on each target. The defender has a budget B for total effort, i.e., $\sum_i \beta_i \leq B$. In practice, β_i may represent the number of hours spent on foot patrolling in target i. If the defender has two teams of patrollers, each of which can patrol for 5 hours a day, then $B = 10$. The planned patrols have to be conveyed clearly to the *human* patrollers on the ground to then be executed (Plumptre et al. 2014). Thus, an arbitrary value of β_i may be impractical. For example, we may ask the patrollers to patrol in an area for 30 minutes, but not 21.3634 minutes. Therefore, we enforce discretized patrol effort levels, requiring $\beta_i \in \Psi = \{\psi_1, \ldots, \psi_J\}$ for J levels of effort.

Some targets may be attacked. The reward of a patrol corresponds to the total number of targets where attacks were detected (Critchlow et al. 2015). Let the expected reward of a patrol vector $\vec{\beta}$ be $\mu(\vec{\beta})$. Our objective is to specify an effort vector $\vec{\beta}^{(t)}$ for each timestep t in an online fashion to minimize regret with respect to the optimal effort vector $\vec{\beta}^\star$ against a stochastic adversary, where regret is defined as $T\mu(\vec{\beta}^\star) - \sum_{t=1}^{T} \mu(\vec{\beta}^{(t)})$.

In practice, the likelihood of a defender detecting an attack is dependent on the amount of patrol effort exerted. In previous work on poaching prevention, a target can represent a large region, e.g., 1×1 km area, where snares are well-hidden. Thus, spending more time means the human patrollers can check the whole region more thoroughly and are more likely to detect snares. We represent the defender's expected reward at target i as a function $\mu_i(\beta_i) \in [0, 1]$. We define random variables $X_i^{(t)}$ as the observed reward (attack or no attack) from target i at time t. Then $X_i^{(t)}$ follows a Bernoulli distribution with mean $\mu_i(\beta_i^{(t)})$ with effort $\beta_i^{(t)}$.

3.1 Domain Characteristics

We leverage the following four characteristics pervasive throughout green security domains to direct our approach.

Decomposability The overall expected reward for the defender is decomposable across targets and additive. For executing a patrol with effort $\vec{\beta}$ across all targets, the expected composite reward is a function $\mu(\vec{\beta}) = \sum_{i=1}^{N} \mu_i(\beta_i)$.

Lipschitz-continuity As discussed, the expected reward to the defender at target i is given by the function $\mu_i(\beta_i)$, which is dependent on effort β_i. Furthermore, the expected reward depends on the features $\vec{y}_i$ of that target, that is, $\mu_i(\beta_i) = \widetilde{\mu}(\vec{y}_i, \beta_i)$ for all i. Past work to predict poaching patterns using machine learning models, which rely on assumptions of Lipschitz continuity, have been shown to perform well in real-world field experiments (Xu et al. 2020). Thus, we assume that the reward function $\widetilde{\mu}(\cdot, \cdot)$ is Lipschitz-continuous in feature space as well as across effort levels. That is, two distinct targets in the protected area with identical features will have identical reward functions, and two targets a and b with features $\vec{y}_a$, $\vec{y}_b$ and effort β_a, β_b have rewards that differ by no more than

$$|\widetilde{\mu}(\vec{y}_a, \beta_a) - \widetilde{\mu}(\vec{y}_b, \beta_b)| \leq L \cdot \mathcal{D}((\vec{y}_a, \beta_a), (\vec{y}_b, \beta_b)) \quad (1)$$

for some Lipschitz constant L and distance function $\mathcal{D}$, such as Euclidean distance. Hence, the composite reward $\mu(\vec{\beta})$ is also Lipschitz-continuous.

For simplicity of notation, we assume that the same Lipschitz constant applies over the entire space. In practice, we could achieve tighter bounds by separately estimating the Lipschitz constant for each dimension.

Monotonicity The more effort spent on a target, the higher the expected reward as our likelihood of finding a snare increases. That is, we assume $\mu(\beta_i)$ is monotonically non-decreasing in β_i. Additionally, we assume that zero effort corresponds with zero reward ($\mu_i(0) = 0$), as defenders cannot prevent attacks on targets they do not visit.

Historical data Finally, many conservation areas have data from past patrols, which we use to warm start the online learning algorithm. To improve our short-term performance, we are careful in how we integrate historical data, as described in Sec. 4.3.

3.2 Domain Challenges

There are additional challenges in green security domains that prevent us from directly applying existing algorithms. No-regret guarantees describe the performance at infinite time horizons, but, practically, we require strong performance within short time frames. Losses in initial rounds are less tolerable, as the consequences of ineffective patrols are deforestation and loss of wildlife. In addition, conservation managers may lose faith in using an online learning approach. In response, we provide an algorithm with infinite horizon guarantees and also empirically show strong performance in the short term.

4 LIZARD Online Learning Algorithm

We present our algorithm, LIZARD (LIpschitZ Arms with Reward Decomposability), for online learning in green security domains.

Standard bandit algorithms suffer from the curse of dimensionality: the set of arms would be Ψ^N, which has size J^N. Thus, we cast the problem as a combinatorial bandit (Chen et al. 2016). At each iteration, we choose a patrol strategy $\vec{\beta}$ that satisfies the budget constraint and observe the patrol outcome of each target i under the chosen effort β_i. An *arm* is one effort level β_i on a specific target i; a *super arm* is $\vec{\beta}$, a collection of N arms. By tracking decomposed rewards, we only need to track observations from NJ arms.

We now maintain exponentially fewer samples, but the number of arms is still prohibitively large. With, say, $N =$

Algorithm 1: LIZARD

Inputs: Number of targets N, time horizon T, budget B, discretization levels Ψ, target features $\vec{y}_i$
$n(i, \psi_j) = 0$, $\texttt{reward}(i, \psi_j) = 0 \quad \forall i \in [N], j \in [J]$
for $t = 1, 2, \ldots, T$ **do**
  Compute UCB_t using Eq. 4
  Solve $\mathcal{P}(\text{UCB}_t, B, N, T)$ to select super arm $\vec{\beta}$
  Observe rewards $X_1^{(t)}, X_2^{(t)}, \ldots, X_n^{(t)}$
  for $i = 1, 2, \ldots, N$ **do**
    $\texttt{reward}(i, \beta_i) = \texttt{reward}(i, \beta_i) + X_i^{(t)}$
    $n(i, \beta_i) = n(i, \beta_i) + 1$

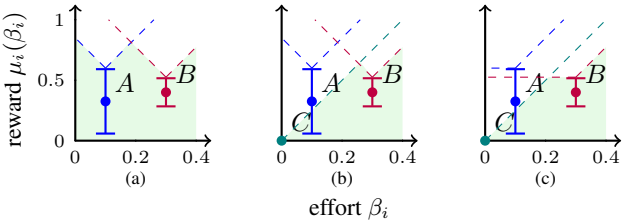

Figure 3: The Lipschitz assumption enables us to prune confidence bounds. We show the impact of each SELFUCBs on the UCBs of other arms in effort space of target i. The solid brackets represent the SELFUCBs. The dashed lines represent the bounds imposed by each arm on the rest of the space. The shaded green region covers the potential value of the reward function at different levels of effort. We visualize the additive effect of (a) Lipschitz-continuity, (b) zero effort yields zero reward, and (c) monotonicity. Note that these plots demonstrate UCBs for one target and that Lipschitz continuity also applies *across* targets based on feature similarity.

100 targets and $J = 10$ effort levels, we would need to develop estimates of reward for 1,000 arms. To address this challenge, we leverage feature similarity between arms to speed up learning, demonstrating the synergy between decomposability and Lipschitz-continuity. In this section, we show how we transfer knowledge between arms with similar effort levels and features.

4.1 Upper Confidence Bounds with Similarity

We take an upper confidence bound (UCB) approach where the rewards are tracked separately for different arms. We show that we can incorporate Lipschitz-continuity of the reward functions into the UCB of each arm to achieve tighter confidence bounds.

Let $\bar{\mu}_t(i, j) = \texttt{reward}_t(i, \psi_j)/n_t(i, \psi_j)$ be the average reward of target i at effort ψ_j given cumulative empirical reward $\texttt{reward}_t(i, \psi_j)$ over $n_t(i, \psi_j)$ arm pulls by timestep t.

Let $r_t(i, j)$ be the *confidence radius* at time t defined as

$$r_t(i, j) = \sqrt{\frac{3 \log(t)}{2 n_t(i, \psi_j)}} \,. \tag{2}$$

We distinguish between UCB and a term we call SELFUCB. The SELFUCB of an arm (i, j) representing target i with effort level j at time t is the UCB of an arm based only on its own observations, given by

$$\text{SELFUCB}_t(i, j) = \bar{\mu}_t(i, j) + r_t(i, j) \,. \tag{3}$$

This definition of SELFUCB corresponds with the standard interpretation of confidence bounds from UCB1 (Auer, Cesa-Bianchi, and Fischer 2002). The UCB of an arm is then computed by taking the minimum of the bounds of all SELFUCBs as applied to the arm. These bounds are determined by adding the distance between arm (i, j) and all other arms (u, v) to the SELFUCB:

$$\text{UCB}_t(i, j) = \min_{\substack{u \in [N] \\ v \in [J]}} \{\text{SELFUCB}_t(u, v) + L \cdot \mathit{dist}\} \tag{4}$$
$$\mathit{dist} = \max\{0, \psi_v - \psi_j\} + \mathcal{D}(\vec{y}_i, \vec{y}_u)$$

which exploits Lipschitz continuity between the arms. See Fig. 3 for a visualization. The distance between two arms depends on the similarity of their features and effort. The first term of *dist* considers similarity of effort level (Fig. 3a); the second considers feature similarity between targets according to distance function $\mathcal{D}$. We define $\text{UCB}_t(i, 0) = 0$ for all $i \in [N]$ due to the assumption that zero effort yields zero reward (Fig. 3b). To address the monotonically non-decreasing reward across effort space, we constrain the first term of *dist* to be nonnegative (Fig. 3c).

4.2 Super Arm Selection

With the computed UCBs, the selection of super arms (patrol strategies) can be framed as a knapsack optimization problem. We aim to maximize the value of our sack (sum of the UCBs) subject to a budget constraint (total effort). To solve this problem, we use the following integer linear program $\mathcal{P}$.

$$\begin{aligned} \max_{z} \quad & \textstyle\sum_{i \in [N]} \sum_{j \in [J]} z_{i,j} \cdot \text{UCB}_t(i, j) && (\mathcal{P}) \\ \text{s.t.} \quad & z_{i,j} \in \{0, 1\} && \forall i \in [N], j \in [J] \\ & \textstyle\sum_{j \in [J]} z_{i,j} = 1 && \forall i \in [N] \\ & \textstyle\sum_{i \in [N]} \sum_{j \in [J]} z_{i,j} \psi_j \leq B \end{aligned}$$

There is one auxiliary variable $z_{i,j}$, constrained to be binary, for each level of patrol effort for each target. Setting $z_{i,j} = 1$ means we exert ψ_j effort on target i. The second constraint sets β_i by requiring that we pull one arm per target (which may be the zero effort arm $\beta_i = 0$). The third constraint mandates that we stay within budget. This integer program has NJ variables and $N + 1$ constraints. Pseudocode for the LIZARD algorithm is given in Algorithm 1.

4.3 Incorporating Historical Data

We incorporate historical data to further improve bounds without compromising the regret guarantee (see Sec. 5.3). Shivaswamy and Joachims (2012) show that, in the infinite horizon case, a logarithmic amount of historical data can reduce regret from logarithmic to constant. However, care must be taken to consider short-term effects. Standard methods that include historical arm pulls as if they occurred online can result in poor short-term performance, as in the following example:

Example 1. *Let A and B be two arms with noise-free rewards $\mu(A) = 1$ and $\mu(B) = 0$. Suppose that in the historical data, A is pulled $H \gg 1$ times and B has never been pulled. If those historical arm pulls are counted in the calculation of confidence radii, then we must pull the bad arm B first $s \geq O(H/\log H)$ times until the UCB of arm B is smaller than the UCB of arm A:*

$$\sqrt{\frac{3\log s}{2s}} = 0 + r_s(B) \leq 1 + r_s(A) = 1 + \sqrt{\frac{3\log s}{2H}}$$

yielding linear regret $O(s) = O\left(\frac{H}{\log H}\right)$ in the first s rounds.

Intuitively, the more times the optimal arm is pulled in the historical data, the greater the short-term regret. Because we expect the historical data to oversample good arms (as patrollers prefer visiting areas that are frequently attacked), we design LIZARD to avoid this problem by ignoring the number of historical arm pulls when computing the confidence radii. Specifically, LIZARD initializes the observed rewards $\texttt{reward}_t(\cdot,\cdot)$ and the number of pulls $n_t(\cdot,\cdot)$ with the historical observations, but, when computing the confidence radii $r_t(i,j)$, LIZARD only considers the number of pulls from online play.

5 Regret Analysis

We provide a regret bound for Algorithm 1 with fixed discretization (Sec. 5.1), which is useful in practice but cannot achieve theoretical no-regret due to the discretization factor. Thus, we then offer Algorithm 2 with adaptive discretization to achieve no-regret (Sec. 5.2), showing that there is no barrier to achieving no regret in practice other than the need for fixed discretization in operationalizing our algorithm in the field. Our regret bound improves upon that of the zooming algorithm of Kleinberg, Slivkins, and Upfal (2019) for all dimensions $d > 1$. In our problem formulation, each dimension of the continuous action space represents a target, so $d = N$. In fact, the regret bound for the zooming algorithm is a provable lower bound; we are able to improve this lower bound through decomposition (Sec. 5.3). Furthermore, we extend the line of research on combinatorial bandits, generalizing the CUCB algorithm from Chen et al. (2016) to continuous spaces.

Full proofs are included in the appendix.

5.1 Fixed Discretization

Theorem 2. *Given the minimum discretization gap Δ, number of arms N, Lipschitz constant L, and time horizon T, the regret bound of Algorithm 1 with* SELFUCB *is*

$$\textit{Reg}_\Delta(T) \leq O\left(NL\Delta T + \sqrt{N^3\Delta^{-1}T\log T} + N^2L\Delta^{-1}\right). \quad (5)$$

Proof sketch. The regret in Theorem 2 comes from (i) discretization regret in the first term of Eq. 5 and (ii) suboptimal arm selections in the last two terms. The discretization regret is due to inaccurate approximation caused by discretization, where the error can be bounded by rounding the optimal arm selection and fractional effort levels to their closest discretized levels. The suboptimal arm selections are due to insufficient samples across all discretized subarms and have sublinear regret in terms of T. □

Under a finite horizon, it is sufficient to pick a discretization level based on the time horizon. For example, given a finite horizon T, we can pick a discretization that is optimal relative to T. We want to use a finer discretization to minimize the discretization regret, while at the same time minimizing the number of effort levels to explore. Thus, we trade off between the selection regret and discretization regret: we want them to be of the same order as the regret bound in Theorem 2, i.e., $O(\sqrt{N^3\Delta^{-1}T\log T}) = O(NL\Delta T)$, or equivalently $\Delta = (\log T/T)^{\frac{1}{3}}N^{\frac{1}{3}}L^{-\frac{2}{3}}$. This result matches the intuition that we should use a finer discretization when either (i) the total timesteps is larger, (ii) the number of targets is smaller, or (iii) the Lipschitz constant is larger (i.e., function values change more drastically).

With Theorem 2 we observe that the barrier to achieving no-regret is the discretization error which is linear in all terms, which brings us to adaptive discretization.

5.2 Adaptive Discretization

To achieve no-regret with an infinite time horizon, we need adaptive discretization. Adaptive discretization is less practical on the ground, but would be useful in other bandit settings where we could more precisely spend our budget such as influence maximization and facility location. As shown in Algorithm 2, we begin with a coarse patrol strategy, beginning with binary decisions on whether or not to visit each target, then gradually progress to a finer discretization.

Theorem 3. *Given the number of arms N, Lipschitz constant L, and time horizon T, the regret bound of Algorithm 2 with* SELFUCB *is be given by*

$$\textit{Reg}(T) \leq O\left(L^{\frac{4}{3}}NT^{\frac{2}{3}}(\log T)^{\frac{1}{3}}\right). \quad (6)$$

Proof sketch. The regret bound in Theorem 2 is not sublinear due to the additional discretization error. An intuitive way to alleviate this error is to adaptively reduce the discretization gap. We run each discretization gap Δ for $T_\Delta = \frac{N}{L^2\Delta^3}\log\frac{N}{L^2\Delta^3}$ time steps to make the discretization error and the selection error of the same order. We then start over with a finer discretization $\Delta/2$ to make the discretization error smaller. After summing the regret from all different phrases of discretization, we achieve sublinear regret as shown in Eq. 6. □

Algorithm 2: Adaptively Discretized LIZARD

Inputs: Number of targets N, time horizon T, budget B, target features $\vec{y}_i$, Lipschitz constant L
Discretization levels $\Psi = \{0, 1\}$, gap $\Delta = 1$
$T_k = \frac{N2^{3k}}{L^2} \log \frac{N2^{3k}}{L^2}$ $\forall k \in \mathbb{N} \cup \{0\}$
$n(i, \psi_j) = 0$, $\texttt{reward}(i, \psi_j) = 0$ $\forall i \in [N], j \in [J]$
for $t = 1, 2, \ldots, T$ **do**
  if $t > \sum_{j=0}^{k-1} T_j$ **then**
    Set $\Delta = 2^{-k}$ and $\Psi = \{0, \Delta, ..., 1\}$
  Compute UCB_t using Eq. 4
  Solve $\mathcal{P}(\text{UCB}_t, B, N, T)$ to select super arm $\vec{\beta}$
  Observe rewards $X_1^{(t)}, X_2^{(t)}, \ldots, X_n^{(t)}$
  for $i = 1, 2, \ldots, N$ **do**
    $\texttt{reward}(i, \beta_i) = \texttt{reward}(i, \beta_i) + X_i^{(t)}$
    $n(i, \beta_i) = n(i, \beta_i) + 1$

Under reasonable problem settings, T dominates all other variables, so the regret in Theorem 3 is effectively of order $O(T^{\frac{2}{3}}(\log T)^{\frac{1}{3}})$. Our regret bound matches the bound of the zooming algorithm with covering dimension $d = 1$. Our setting is instead N-dimensional, falling into a space with covering dimension $d = N$. The regret bound for a metric space with covering dimension d is $O(T^{\frac{d+1}{d+2}}(\log T)^{\frac{1}{d+2}})$, which approaches $\tilde{O}(T)$ as d approaches infinity (Kleinberg, Slivkins, and Upfal 2008). More generally, our regret bound improves upon that of the zooming algorithm for any $d > 1$.

Theorem 3 signifies that LIZARD can successfully decouple the N-dimensional metric space into individual sub-dimensions while maintaining the smaller regret order, showcasing the power of decomposibility.

5.3 Tightening Confidence Bounds

We have so far offered regret bounds to account for decomposability and Lipschitz-continuity across effort space. We now guarantee that the regret bounds continue to hold with Lipschitz-continuity in feature space, monotonicity, and historical information. We first look at how prior knowledge affects the regret bound in the combinatorial bandit setting:

Theorem 4. *Consider a combinatorial multi-arm bandit problem. If the bounded smoothness function given is $f(x) = \gamma x^\omega$ for some $\gamma > 0, \omega \in (0, 1]$ and the Lipschitz upper confidence bound is applied to all m base arms, the cumulative regret at time T is bounded by*

$$Reg(T) \leq \frac{2\gamma}{2-\omega}(6m \log T)^{\frac{\omega}{2}} \cdot T^{1-\frac{\omega}{2}} + \left(\frac{\pi^2}{3} + 1\right) mR_{max}$$

where R_{max} is the maximum regret achievable.

Theorem 4 matches the regret of the CUCB algorithm (Chen et al. 2016), generalizing combinatorial bandits to continuous spaces. Theorem 4 also allows us to generalize Theorems 2 and 3 to our setting with a tighter UCB from Lipschitz-continuity, which yields Theorem 5:

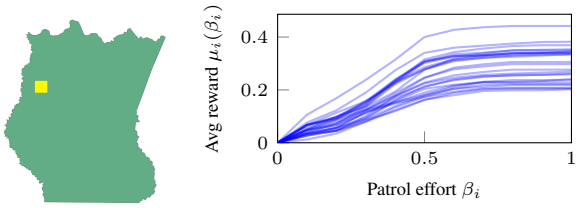

Figure 4: Map of Srepok with a 5×5 km region highlighted and the real-world reward functions of the corresponding 25 targets.

Theorem 5. *Given the minimum discretization gap Δ, number of targets N, Lipschitz constant L, and time horizon T, the regret bound of Algorithm 1 with* UCB *is*

$$Reg_\Delta(T) \leq O\left(NL\Delta T + \sqrt{N^3\Delta^{-1}T \log T} + N^2L\Delta^{-1}\right) \quad (7)$$

and the regret bound of Algorithm 2 with UCB *is*

$$Reg(T) \leq O\left(L^{\frac{4}{3}}NT^{\frac{2}{3}}(\log T)^{\frac{1}{3}}\right) . \quad (8)$$

Finally, when historical data is used, we can treat all the historical arm pulls as previous arm pulls with regret bounded by the maximum regret $R_{\max}$.[2] This yields a regret bound with a time-independent constant and is thus still sublinear, achieving no-regret. Taken together, these capture all the properties of the LIZARD algorithm, so we can state:

Corollary 6. *The regret bounds of Algorithm 1 and Algorithm 2 still hold with the inclusion of decomposability, Lipschitz-continuity, monotonicity, and historical data.*

Theorem 5 highlights the interplay between Lipschitz-continuity and decomposition. The zooming algorithm achieves the provable lower bound on regret $\tilde{O}(T^{\frac{d+1}{d+2}})$ (Kleinberg 2004). *The regret that we achieve improves upon that lower bound for all $d > 1$, which is only possible due to the addition of decomposition.*

6 Experiments

We conduct experiments using both synthetic data and real poaching data. Our results validate that the addition of decomposition and Lipschitz-continuity not only improves our theoretical guarantees but also leads to stronger empirical performance. We show that LIZARD (Algorithm 1) learns effectively within practical time horizons.

We consider a patrol planning problem with $N = 25$ or 100 targets (each a 1 sq. km grid cell), representing the region reachable from a single patrol post, and time horizon $T = 500$ representing a year and a half of patrols. The budget is the number of teams, e.g., $B = 5$ corresponds to 5 teams of rangers. We use 50 timesteps of historical data, approximately two months of patrol, as we focus on achieving strong performance in parks with limited historical patrols.

[2]Note that LIZARD treats all historical pulls as a single arm pull with return equal to the average return to avoid short-term losses (Sec. 4.3). However, the regret bound covers the broader case of using *any* finite number of historical arm pulls.

Table 1: Performance across multiple problem settings, throughout which LIZARD achieves closest-to-optimal performance.

	SYNTHETIC DATA								REAL-WORLD DATA							
$(N, B) =$	(25, 1)		(25, 5)		(100, 5)		(100, 10)		(25, 1)		(25, 5)		(100, 5)		(100, 10)	
$T =$	200	500	200	500	200	500	200	500	200	500	200	500	200	500	200	500
LIZARD	**40.9**	**55.0**	**20.3**	**37.8**	**26.5**	**44.9**	**15.8**	**54.6**	**79.5**	**93.1**	**61.4**	**70.6**	**74.7**	**82.7**	**56.8**	**71.9**
CUCB	5.7	0.8	15.7	29.4	−3.4	−2.3	−0.4	12.1	45.0	57.1	49.5	68.3	25.5	50.9	35.5	49.4
MINION	−42.7	−46.4	−60.0	−35.9	−1.1	−53.3	−9.1	−21.3	−13.1	−39.3	−28.7	−10.1	−18.4	−14.4	−18.6	−13.7
Zooming	−20.6	−18.8	−0.4	−6.8	−25.1	−24.5	−22.8	−21.5	−14.4	−5.5	40.5	40.8	−38.5	−36.0	−21.5	−22.7

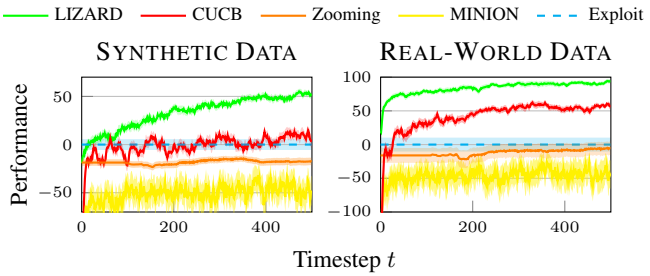

Figure 5: Performance, measured in terms of percentage of reward achieved between OPTIMAL − EXPLOIT, over time. Shaded region shows standard error. Setting shown is $N = 25$, $B = 1$. LIZARD (green) performs best.

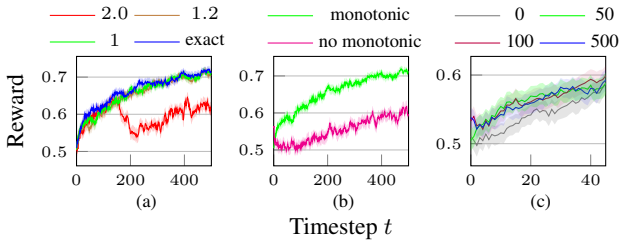

Figure 6: Impact of (a) different Lipschitz constants L_i, (b) monotonicity assumption, (c) varying amount of historical data. The green line in each plot depicts the setting used by LIZARD. Run on synthetic data with $N = 25$, $B = 1$.

Real-world data We leverage patrol data from Srepok Wildlife Sanctuary in Cambodia to learn the reward function, as dependent on features and effort (Xu et al. 2020). We train a machine learning classifier to predict whether rangers will detect poaching activity at each target for a given effort level. We then generate piecewise-linear approximations of the reward functions, shown in Fig. 4, which naturally intersects the origin and is nondecreasing with increased effort.

Synthetic data To produce synthetic data, we generate piecewise-linear functions that mimic the behavior of real-world reward functions. We define feature similarity as the maximum difference between the reward functions across all effort levels. We generate historical data with a biased distribution over the targets, corresponding to the realistic scenario where rangers spend more effort on targets that are easily accessible, such as those closest to a patrol post.

Algorithms We compare to three baselines: *CUCB* (Chen et al. 2016), *zooming* (Kleinberg, Slivkins, and Upfal 2019), and *MINION* (Gholami et al. 2018). Zooming is an online learning algorithm that ignores decomposability, whereas CUCB uses decomposition but ignores similarity between arms. We use *exploit history* as a naive baseline, which greedily exploits historical data with a static strategy. We compute the *optimal* strategy exactly by solving a mixed-integer program over the true piecewise-linear reward functions, subject to the budget constraint. Experiments were run on a macOS machine with 2.4 GHz quad-core i5 processors and 16 GB memory. We take the mean of 30 trials.

Fig. 5 shows performance on both real-world and synthetic data, evaluated as the reward achieved at timestep t, where the reward of historical exploit is 0 and of optimal is 1. The performance of LIZARD significantly surpasses that of the baselines throughout. On the real-world data, LIZARD provides a particular advantage over CUCB in the early rounds. Table 1 shows the performance of each algorithm at $T = 200$ and 500, with varying values of N and B. LIZARD beats the baselines in every scenario.

We return to the four characteristics of green security domains (Sec. 3.1) and show that integrating each feature improves LIZARD. See Fig. 6 for results from our ablation study. **Decomposability:** CUCB, a naive decomposed bandits algorithm, greatly exceeds the non-decomposed zooming algorithm throughout most of Table 1, demonstrating the value of decomposition. **Lipschitz-continuity:** Fig. 6a reveals the value of information gained from knowing the exact value of the Lipschitz constant in each dimension (L_i exact). We do not assume perfect knowledge of the Lipschitz constants L_i, instead selecting an approximate value $L_i = 1$ and using that same estimate across all dimensions. As shown, significantly overestimating L_i with $L_i = 2$ hinders performance; this setting is closer to that of CUCB, with no Lipschitz continuity. **Monotonicity:** Fig. 6b shows that the monotonicity assumption adds a significant boost to performance. **Historical data:** Fig. 6c demonstrates the value of adding historical information in early rounds. By timestep 40, the benefit of historical data is negligible.

7 Conclusion

We present LIZARD, an integrated algorithm for online learning in green security domains that leverages the advantages of decomposition and Lipschitz-continuity. With this approach, we transcend the proven lower regret bound of the zooming algorithm for Lipschitz bandits and extend combinatorial bandits to continuous spaces, showcasing their combined benefit. These results validate our approach of treating real-world conditions not as constraints but rather as useful features that lead to faster convergence. On top of achieving theoretical no-regret, we also demonstrate improved short-term performance empirically, increasing the usefulness of this approach in practice—particularly in high-stakes environments where we cannot compromise short-term reward.

The next step is to move toward deployment. Researchers have demonstrated success deploying patrol strategies in the field to prevent poaching (Xu et al. 2020). Our LIZARD algorithm could be directly implemented as-is by integrating historical data, estimating the Lipschitz constant, and setting the budget as the number of hours available per day.

Ethics and Broader Impact

This research has been conducted in close collaboration with domain experts to design our approach. Their insights have shaped the trajectory of our project. For example, our initial idea was to plan information-gathering patrols, taking an active learning approach to gather data where the predictive model was most uncertain. However, conservation experts pointed out in subsequent discussions that patrollers could not afford to spend time purely gathering data; they must prioritize preventing illegal poaching, logging, and fishing in the short-term. These priorities guided our project, particularly our focus on minimizing short-term regret as well as long-term regret.

Our work is intended to serve as an assistive technology, helping rangers identify potentially snare-laden regions they otherwise might have missed, given that these parks can be up to thousands of square kilometers and rangers can only patrol a small number of regions at each timestep. We do not intend this work to replace the critical role that rangers play in conservation management; no AI tool could substitute for the complex skills and domain insights that rangers and park managers have to plan and conduct patrols.

A concern that might be raised could be that poachers who get access to this algorithm could predict where rangers might go and therefore place their snares more strategically. However, this algorithm would only be of use with access to the patrol data, which poachers would not have.

References

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47(2-3): 235–256.

Basilico, N.; Gatti, N.; and Amigoni, F. 2009. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proceedings of the Eighth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-09)*, 57–64.

Berton, E. F. 2020. Rare trees are disappearing as 'wood pirates' log Bolivian national parks. https://news.mongabay.com/2020/01/rare-trees-are-disappearing-as-wood-pirates-log-bolivian-national-parks/.

Blum, A.; Haghtalab, N.; and Procaccia, A. D. 2014. Learning optimal commitment to overcome insecurity. In *Advances in Neural Information Processing Systems 27 (NeurIPS-14)*, 1826–1834.

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

Bubeck, S.; Munos, R.; Stoltz, G.; and Szepesvári, C. 2011. X-armed bandits. *Journal of Machine Learning Research* 12(May).

Chen, W.; Wang, Y.; Yuan, Y.; and Wang, Q. 2016. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research* 17(1): 1746–1778.

Critchlow, R.; Plumptre, A. J.; Driciru, M.; Rwetsiba, A.; Stokes, E. J.; Tumwesigye, C.; Wanyama, F.; and Beale, C. 2015. Spatiotemporal trends of illegal activities from ranger-collected data in a Ugandan national park. *Conservation Biology* 29(5): 1458–1470.

Fang, F.; Nguyen, T. H.; Pickles, R.; Lam, W. Y.; Clements, G. R.; An, B.; Singh, A.; Tambe, M.; and Lemieux, A. 2016. Deploying PAWS: Field Optimization of the Protection Assistant for Wildlife Security. In *Proceedings of the Twenty-eighth Conference on Innovative Applications of Artificial Intelligence (IAAI-16)*.

Fang, F.; Stone, P.; and Tambe, M. 2015. 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-15)*.

Gholami, S.; Mc Carthy, S.; Dilkina, B.; Plumptre, A.; Tambe, M.; Driciru, M.; Wanyama, F.; Rwetsiba, A.; Nsubaga, M.; Mabonga, J.; et al. 2018. Adversary models account for imperfect crime data: Forecasting and planning against real-world poachers. In *Proceedings of the Seventeenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-18)*, 823–831.

Gholami, S.; Yadav, A.; Tran-Thanh, L.; Dilkina, B.; and Tambe, M. 2019. Don't Put All Your Strategies in One Basket: Playing Green Security Games with Imperfect Prior Knowledge. In *Proceedings of the Eighteenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-19)*, 395–403.

Gray, R. C.; Zhu, J.; and Ontañón, S. 2020. Regression Oracles and Exploration Strategies for Short-Horizon Multi-Armed Bandits. In *2020 IEEE Conference on Games (CoG)*, 312–319. IEEE.

Haskell, W.; Kar, D.; Fang, F.; Tambe, M.; Cheung, S.; and Denicola, E. 2014. Robust Protection of Fisheries with COmPASS. In *Proceedings of the Twenty-sixth Conference*

on Innovative Applications of Artificial Intelligence (IAAI-14).

Kalai, A.; and Vempala, S. 2005. Efficient algorithms for online decision problems. *J. of Computer and System Sciences* 71(3): 291–307.

Kamra, N.; Gupta, U.; Fang, F.; Liu, Y.; and Tambe, M. 2018. Policy learning for continuous space security games using neural networks. In *Proceedings of the Thirty-second AAAI Conference on Artificial Intelligence (AAAI-18)*.

Kleinberg, R.; Slivkins, A.; and Upfal, E. 2008. Multi-armed bandits in metric spaces. In *Proceedings of the 40th ACM Symposium on Theory of Computing (STOC 2008)*, 681–690. ACM.

Kleinberg, R.; Slivkins, A.; and Upfal, E. 2019. Bandits and experts in metric spaces. *J. of the ACM* .

Kleinberg, R. D. 2004. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems 17 (NeurIPS-04)*, 697–704.

Lattimore, T.; and Szepesvári, C. 2018. Bandit algorithms. *preprint* 28.

Lober, D. J. 1992. Using forest guards to protect a biological reserve in Costa Rica: one step towards linking parks to people. *Journal of Environmental Planning and Management* 35(1): 17–41.

Magureanu, S.; Combes, R.; and Proutiere, A. 2014. Lipschitz bandits: Regret lower bounds and optimal algorithms. In *Proceedings of Conference on Learning Theory (COLT-14)*.

Mc Carthy, S. M.; Tambe, M.; Kiekintveld, C.; Gore, M. L.; and Killion, A. 2016. Preventing illegal logging: Simultaneous optimization of resource teams and tactics for security. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*.

Moreto, W. D.; and Lemieux, A. M. 2015. Poaching in Uganda: Perspectives of law enforcement rangers. *Deviant Behavior* 36(11): 853–873.

Nguyen, T. H.; Sinha, A.; Gholami, S.; Plumptre, A.; Joppa, L.; Tambe, M.; Driciru, M.; Wanyama, F.; Rwetsiba, A.; Critchlow, R.; et al. 2016. CAPTURE: A New Predictive Anti-Poaching Tool for Wildlife Protection. In *Proceedings of the Fifteenth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-16)*, 767–775.

Plumptre, A. J.; Fuller, R. A.; Rwetsiba, A.; Wanyama, F.; Kujirakwinja, D.; Driciru, M.; Nangendo, G.; Watson, J. E.; and Possingham, H. P. 2014. Efficiently targeting resources to deter illegal activities in protected areas. *Journal of Applied Ecology* 51(3): 714–725.

Shivaswamy, P.; and Joachims, T. 2012. Multi-armed bandit problems with history. In *Artificial Intelligence and Statistics*, 1046–1054.

Xu, H.; Ford, B.; Fang, F.; Dilkina, B.; Plumptre, A.; Tambe, M.; Driciru, M.; Wanyama, F.; Rwetsiba, A.; Nsubaga, M.; et al. 2017. Optimal patrol planning for green security games with black-box attackers. In *Proceedings of the 8th Conference on Decision and Game Theory for Security (GameSec-17)*, 458–477. Springer.

Xu, H.; Tran-Thanh, L.; and Jennings, N. R. 2016. Playing repeated security games with no prior knowledge. In *Proceedings of the Fifteenth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-16)*, 104–112.

Xu, L.; Gholami, S.; Carthy, S. M.; Dilkina, B.; Plumptre, A.; Tambe, M.; Singh, R.; Nsubuga, M.; Mabonga, J.; Driciru, M.; et al. 2020. Stay Ahead of Poachers: Illegal Wildlife Poaching Prediction and Patrol Planning Under Uncertainty with Field Test Evaluations. In *Proceedings of the 36th IEEE International Conference on Data Engineering (ICDE-20)*.

A Proof of Theorem 2

Theorem 2. *Given the minimum discretization gap Δ, number of arms N, Lipschitz constant L, and time horizon T, the regret bound of Algorithm 1 with* SELFUCB *is*

$$Reg_{\Delta}(T) \leq O\left(NL\Delta T + \sqrt{N^3\Delta^{-1}T\log T} + N^2L\Delta^{-1}\right). \quad (5)$$

Proof. We first analyze the discretization regret. We then bound the arm selection regret by treating the problem as a combinatorial multi-armed bandit problem and apply a theorem given by Chen et al. (2016). Combining these two regret terms gives us the overall regret bound for the fixed discretization algorithm.

Discretization Regret Using discretization levels Ψ with a minimum gap Δ, let OPT_Ψ be the true optimum value and $\vec{\beta}^\star_\Psi$ be the optimal solution when the rewards of all discretized points are fully known. Let the true optimum and the optimal solution without discretization be OPT and $\vec{\beta}^\star$. Then we have

$$\begin{aligned}
\text{OPT} = \mu(\vec{\beta}^\star) &= \sum_{i=1}^{N} \mu_i(\beta_i^\star) \\
&\leq \sum_{i=1}^{N} \left(\mu_i(\lfloor\beta_i^\star\rfloor_\Psi) + L|\beta_i^\star - \lfloor\beta_i^\star\rfloor_\Psi|\right) \\
&\leq \sum_{i=1}^{N} \left(\mu_i(\lfloor\beta_i^\star\rfloor_\Psi) + L\Delta\right) \\
&= \sum_{i=1}^{N} \mu_i(\lfloor\beta_i^\star\rfloor_\Psi) + \sum_{i=1}^{N} L\Delta \\
&\leq \text{OPT}_\Psi + NL\Delta\,. \qquad (9)
\end{aligned}$$

This yields $\text{OPT} - \text{OPT}_\Psi \leq NL\Delta$, which is the discretization regret incurred by the given discretization.

Selection Regret Given a discretization Ψ, we treat each target with a specified patrol effort as a base arm. If we assume the smallest discretization gap is Δ, we can bound the total number of base arms by N/Δ. We denote a feasible patrol coverage as a super arm: a set of base arms that must satisfy the budget constraints and the restriction of selecting exactly one effort level per target. The set of all feasible super arms implicitly forms a feasibility constraint, which fits into the context of the combinatorial multi-armed bandit problem (Chen et al. 2016). In addition, our Lipschitz reward function also satisfies the bounded smoothness condition with

$$\begin{aligned}
|\mu(\vec{\beta}) - \mu'(\vec{\beta})| &= \left|\sum_i \mu_i(\beta_i) - \mu_i'(\beta_i)\right| \\
&\leq N \max_{i,\beta_i} |\mu_i(\beta_i) - \mu_i'(\beta_i)| = N\|\mu - \mu'\|_\infty = f(\Lambda) \qquad (10)
\end{aligned}$$

where $f(\Lambda) = N\Lambda, \Lambda = \|\mu - \mu'\|_\infty$ is a linear function of Λ. So we can apply the regret bound of combinatorial multi-armed bandit with polynomial bounded smoothness function $f(x) = \gamma x^\omega$ (Theorem 2 given by Chen et al.)

$$\text{Reg}(T) \leq \frac{2\gamma}{2-\omega}(6m\log T)^{\frac{\omega}{2}} \cdot T^{1-\frac{\omega}{2}} + \left(\frac{\pi^2}{3}+1\right) mR_{\max}$$

where $\omega = 1, \gamma = N$, m is the number of base arms, and $R_{\max}$ is the maximum regret that can be achieved. In our domain, the number of base arms is bounded by $m \leq N/\Delta$. The maximum regret is bounded by the maximum reward achievable, which can be bounded by monotonicity and Lipschitz-continuity:

$$R_{\max} \leq \mu(\vec{\beta}^\star) = \sum_i \mu_i(\beta_i^\star) \leq NL\,.$$

By substituting $\omega = 1, \gamma = N$, and using upper bounds for m and $R_{\max}$, we get a valid regret bound of Algorithm 1 with fixed discretization gap Δ:

$$\text{Reg}_\Delta^{\text{discrete}}(T) \leq 2N\sqrt{\frac{6NT\log T}{\Delta}} + \left(\frac{\pi^2}{3}+1\right)\frac{N^2L}{\Delta}\,. \quad (11)$$

This regret is defined with respect to the optimum over all feasible discrete selections. Therefore, we can combine Inequalities 9 and 11 to derive the regret with respect to the true optimum

$$\begin{aligned}
\text{Reg}_\Delta(T) &\leq 2N\sqrt{\frac{6NT\log T}{\Delta}} + \left(\frac{\pi^2}{3}+1\right)\frac{N^2L}{\Delta} + NL\Delta T \\
&= O\left(\sqrt{\frac{N^3T\log T}{\Delta}} + \frac{N^2L}{\Delta} + NL\Delta T\right) \qquad (12)
\end{aligned}$$

where the gap in Inequality 9 is for each iteration, so must be multiplied by the number of timesteps T. □

B Proof of Theorem 3

Theorem 3. *Given the number of arms N, Lipschitz constant L, and time horizon T, the regret bound of Algorithm 2 with* SELFUCB *is be given by*

$$Reg(T) \leq O\left(L^{\frac{4}{3}}NT^{\frac{2}{3}}(\log T)^{\frac{1}{3}}\right). \qquad (6)$$

Proof. We want to set $2N\sqrt{6NT\log T/\Delta}$ and $NL\Delta T$ in Eq. 12 to be of the same order. This is equivalent to solving

$$\begin{aligned}
&O\left(N\sqrt{NT\log T/\Delta}\right) = O(NL\Delta T) \\
\iff &O\left(\frac{T}{\log T}\right) = O\left(\frac{N}{L^2\Delta^3}\right).
\end{aligned}$$

Applying the fact $O(T/\log T) = c \implies T = O(c\log c)$ for any constant c yields

$$T_\Delta = O\left(\frac{N}{L^2\Delta^3}\log\frac{N}{L^2\Delta^3}\right)$$

which is the optimal stopping condition to let the two terms of the regret bound in Eq. 12 be of the same order. We set

$$T_\Delta = \frac{N}{L^2\Delta^3}\log\frac{N}{L^2\Delta^3}\,.$$

Substitute this into Eq. 12 with $t \leq T_\Delta = \frac{N}{L^2\Delta^3}\log\frac{N}{L^2\Delta^3}$ and $\log t \leq \log T_\Delta = \log\left(\frac{N}{L^2\Delta^3}\log\frac{N}{L^2\Delta^3}\right) \leq 2\log\frac{N}{L^2\Delta^3}$ to get

$$\begin{aligned}
&\text{Reg}_\Delta(t) \leq \text{Reg}_\Delta(T_\Delta) \\
&\leq 2N\sqrt{\frac{6NT_\Delta\log T_\Delta}{\Delta}} + \left(\frac{\pi^2}{3}+1\right)\frac{N^2L}{\Delta} + NL\Delta T_\Delta \\
&= (4\sqrt{3}+1)\frac{N^2\log\frac{N}{L^2\Delta^3}}{L\Delta^2} + \left(\frac{\pi^2}{3}+1\right)\frac{N^2L}{\Delta}\,.
\end{aligned}$$

Due to the discretization regret term, the bound given by Eq. 12 is not a sublinear function in T, so is not no-regret. To achieve no-regret, we need to adaptively adjust the discretization levels. Therefore, as described in Algorithm 2, we gradually reduce the discretization gap $\Delta_i = 2^{-i}$ depending on the number of timesteps elapsed.

For total timesteps T, let $k \in \mathbb{N} \cup \{0\}$ such that $\sum_{i=0}^{k-1} T_{\Delta_i} \leq T \leq \sum_{i=0}^{k} T_{\Delta_i}$. Then we can bound the regret at time T by the regret at time $\sum_{i=0}^{k} T_{\Delta_i}$, which yields

$$\begin{aligned}
&\text{Reg}(T) \leq \text{Reg}\left(\sum_{i=0}^{k} T_{\Delta_i}\right) \leq \sum_{i=0}^{k} \text{Reg}_{\Delta_i}(T_{\Delta_i}) \\
&\leq \sum_{i=0}^{k} \left((4\sqrt{3}+1) \frac{N^2 \log \frac{N}{L^2 \Delta_i^3}}{L \Delta_i^2} + \left(\frac{\pi^2}{3}+1\right) \frac{N^2 L}{\Delta_i} \right) \\
&\leq \sum_{i=0}^{k} \left((4\sqrt{3}+1) \frac{N^2}{L} 2^{2i} \log\left(\frac{N}{L^2} 2^{3i}\right) + \left(\frac{\pi^2}{3}+1\right) N^2 L 2^i \right) \\
&\leq (4\sqrt{3}+1) \frac{N^2}{L} 2^{2k+2} \log\left(\frac{N}{L^2} 2^{3k+3}\right) + \left(\frac{\pi^2}{3}+1\right) N^2 L 2^{k+1} .
\end{aligned}$$

Since we know

$$\begin{aligned}
&T = O\left(\sum_{i=0}^{k} T_{\Delta_i}\right) \\
\implies &T = O\left(\sum_{i=0}^{k} \frac{N}{L^2 \Delta_i^3} \log \frac{N}{L^2 \Delta_i^3}\right) \\
\implies &T = O\left(\sum_{i=0}^{k} \frac{N}{L^2} 2^{3i} \log\left(\frac{N}{L^2} 2^{3i}\right)\right) \\
\implies &T = O\left(\frac{N}{L^2} 2^{3k+3} \log\left(\frac{N}{L^2} 2^{3k+3}\right)\right) ,
\end{aligned}$$

we therefore have $T = O(\frac{N}{L^2} 2^{3k} \log(\frac{N}{L^2} 2^{3k}))$, which yields $2^{3k} = O(\frac{L^2 T}{N \log T})$. Replacing all 2^k with $O((\frac{L^2 T}{N \log T})^{\frac{1}{3}})$ provides a regret bound dependent on T and N only:

$$\begin{aligned}
&\text{Reg}(T) \leq O\left(\frac{N^2}{L} 2^{2k} \log\left(\frac{N}{L^2} 2^{3k}\right) + N^2 L 2^k\right) \\
&\leq O\left(\frac{N^2}{L} \left(\frac{L^2 T}{N \log T}\right)^{\frac{2}{3}} \log\left(\frac{T}{\log T}\right) + N^2 L \left(\frac{L^2 T}{N \log T}\right)^{\frac{1}{3}}\right) \\
&\leq O\left(L^{\frac{1}{3}} N T^{\frac{2}{3}} (\log T)^{\frac{1}{3}} + L^{\frac{5}{3}} N^{\frac{5}{3}} T^{\frac{1}{3}} (\log T)^{-\frac{1}{3}}\right) . \qquad (13)
\end{aligned}$$

We only care about the regret order in terms of the dominating variable T, so the second term is dominated by the first term. Therefore, $\text{Reg}(T) \leq O\left(L^{\frac{1}{3}} N T^{\frac{2}{3}} (\log T)^{\frac{1}{3}}\right)$. □

Theorem 4. *Consider a combinatorial multi-arm bandit problem. If the bounded smoothness function given is $f(x) = \gamma x^{\omega}$ for some $\gamma > 0, \omega \in (0, 1]$ and the Lipschitz upper confidence bound is applied to all m base arms, the cumulative regret at time T is bounded by*

$$\textit{Reg}(T) \leq \frac{2\gamma}{2-\omega} (6m \log T)^{\frac{\omega}{2}} \cdot T^{1-\frac{\omega}{2}} + \left(\frac{\pi^2}{3}+1\right) m R_{max}$$

where R_{max} is the maximum regret achievable.

To prove Theorem 4, we first describe a weaker version in Lemma 7. Lemma 7 shows that if all arms are sufficiently sampled, the probability that we hit a bad super arm is small. Lemma 7 attributes all the bad super arm selections with error greater than R_{min} to a maximum regret R_{max}, which overestimates the overall regret and can be tightened with a more careful analysis (omitted here due to space).

Lemma 7. *Given the prior knowledge of Lipschitz-continuity, monotonicity, and the definition of* UCB *in Sec. 4, the regret bound of CUCB algorithm (Chen et al. 2016) holds with*

$$\textit{Reg}(T) \leq \left(\frac{6m \log T}{(f^{-1}(R_{min}))^2} + \frac{\pi^2}{3} m^2 + m\right) R_{max} . \qquad (14)$$

Proof. The main proof relies on whether the tighter UCB defined in Sec. 4 works for the combinatorial multi-armed bandit problem. Specifically, we must confirm that the proof of CUCB given by Chen et al. (2016) works with a different confidence bound definition.

Define $T_{i,t}$ as the number of pulls of arm i up to time t. We assume there are m total base arms. Let $\mu = [\mu_1, \mu_2, \ldots, \mu_m]$ be the vector of true expectations of all base arms. Let $X_{i,t}$ be the revelation of arm i from a pull at time t and $\bar{X}_{i,s} = (\sum_{j=1}^{s} X_{i,j})/s$ be the average reward of the first s pulls of arm i. Therefore, we write $\bar{X}_{i,T_{i,t}}$ to represent the average reward of arm i at time t as defined in Sec. 4. Let S_t be the super arm pulled in round t. We say round t is bad if S_t is not the optimal arm.

Define event $E_t = \{\forall i \in [m], |\bar{X}_{i,T_{i,t-1}} - \mu_i| \leq r'_t(i)\}$, where recall that $r_t(i) = \sqrt{3 \log t (2T_{i,t-1})^{-1}}$ is the standard confidence radius, and $r'_t(i) = \text{UCB}_t(i) - \bar{X}_{i,T_{i,t-1}} \leq \text{SELFUCB}_t(i) - \bar{X}_{i,T_{i,t-1}} = r_t(i)$ is the tighter confidence bound we introduce in Sec. 4. We want to bound the probability that all the sampled averages are close to the true mean value μ_i with distance at most the confidence radius $r'_t(i)\ \forall i$.

At time t, the probability that the sampled mean $\bar{X}_{i,T_{i,t-1}}$ of arm i lies within the ball with center μ_i and radius $r_t(i)$ is bounded by $1 - 2t^{-3}$ (double-sided Hoeffding bound), which characterizes the probability of having one single SELFUCB bound hold for arm i. Moreover, at a fixed time t, if we have all m SELFUCB bounds hold, then all the m UCB bounds will automatically hold because each bound introduced by the Lipschitz continuity from any arm is a valid bound and thus UCB bound, the minimum of all the valid bounds, must also hold. Therefore, using the notation of SELFUCB radius $r'_t(i)$, we have

$$\begin{aligned}
\text{prob}\{\neg E_t\} =& \text{prob}\{|\bar{X}_{i,T_{i,t-1}} - \mu_i| \leq r'_t(i)\ \forall i \in [m]\} \\
\geq& \text{prob}\{|\bar{X}_{i,s} - \mu_i| \leq r'_t(i)\ \forall i \in [m],\ \forall s \in [t]\} \\
\geq& \text{prob}\{|\bar{X}_{i,s} - \mu_i| \leq r_t(i)\ \forall i \in [m],\ \forall s \in [t]\} \\
\geq& 1 - mt \cdot 2t^{-3} \\
=& 1 - 2mt^{-2}
\end{aligned}$$

where the second inequality is due to the fact that $|\bar{X}_{i,s} - \mu_i| \leq r_t(i)\ \forall i \in [m] \Rightarrow |\bar{X}_{i,s} - \mu_i| \leq r'_t(i)\ \forall i \in [m]$ under the Lipschitz condition, and the last inequality is due to

the union bound across all m arms and t time steps. Equivalently, we have $\neg\text{prob}\{\neg E_t\} \leq 2mt^{-2}$.

According to the definition $r'_t(i) = \text{UCB}_t(i) - \bar{X}_{i,T_{i,t-1}}$, E_t is true (thus $|\bar{X}_{i,T_{i,t-1}} - \mu_i| \leq r'_t(i)$) implies $|\text{UCB}_t(i) - \mu_i| \leq 2r'_t(i)$ and $\text{UCB}_t(i) \geq \mu_i\ \forall i$. Let $\Lambda := \sqrt{\frac{3\log t}{2l_t}}$ and $\Lambda'_t := \max_{i\in S_t} r'_t(i) \leq \max_{i\in S_t} r_t(i) := \Lambda_t$. Then we have

$$E_t \Rightarrow\ \forall i \in S_t, |\text{UCB}_t(i) - \mu_i| \leq 2r'_t(i) \leq 2\Lambda'_t$$
$$\{S_t \text{ is bad},\ \forall i \in S_t, T_{i,t-1} > l_t\} \Rightarrow \Lambda > \Lambda_t \geq \Lambda'_t\ .$$

Let $\text{UCB}_t = [\text{UCB}_t(1), \text{UCB}_t(2), \ldots, \text{UCB}_t(m)]$ be a vector of UCBs of all subarms. If the above two events on the left-hand side both hold $\{E_t, S_t \text{ is bad},\ \forall i \in S_t, T_{i,t-1} > l_t\}$ at time t, if we denote $\texttt{reward}_\mu(S) = \sum_{i\in S}\mu_i$, we have

$$\begin{aligned}
&\texttt{reward}_\mu(S_t) + f(2\Lambda) > \texttt{reward}_\mu(S_t) + f(2\Lambda'_t)\\
\geq&\texttt{reward}_{\text{UCB}_t}(S_t) = \text{opt}_{\text{UCB}_t} \geq \texttt{reward}_{\text{UCB}_t}(S^\star_\mu)\\
\geq&\texttt{reward}_\mu(S^\star_\mu) \geq \text{opt}_\mu
\end{aligned} \tag{15}$$

where the first inequality is due to the monotonicity of function f and $\Lambda > \Lambda'_t$; the second is due to the definition of bounded smoothness function f in Eq. 10 and the condition that $E_t \Rightarrow |\text{UCB}_t(i) - \mu_i| \leq 2\Lambda'_t\ \forall i \in S_t$; the third is due to the optimal selection of S_t with respect to UCB_t; the fourth is due to the optimality of $\text{opt}_{\text{UCB}_t}$, where $S^\star_\mu$ is the true optimal solution with respect to the real μ; the fifth is due to the upper confidence bound between $\text{UCB}_t \geq \mu$ when E_t is true; and the sixth is again due to the optimality of $S^\star_\mu$.

By substituting $l_t = \frac{6\log t}{(f^{-1}(R_{\text{min}}))^2}$ into $\Lambda = \sqrt{\frac{3\log t}{2l_t}}$, we have $f(2\Lambda) = R_{\text{min}}$. Eq. 15 contradicts the definition of minimum regret R_{min}, the smallest non-zero regret (since S_t is bad thus not optimal but it is also closer to the optimal with distance smaller than R_{min}). That is:

$$\begin{aligned}
&\text{prob}\{E_t, S_t \text{ is bad},\ \forall i \in S_t, T_{i,t-1} > l_t\} = 0\\
\Rightarrow&\text{prob}\{S_t \text{ is bad},\ \forall i \in S_t, T_{i,t-1} > l_t\} \leq \text{prob}\{\neg E_t\} \leq 2mt^{-2}\ .
\end{aligned}$$

Lastly, we bound the total number of bad arm pulls by the probability that E_t did not happen and a time-dependent term as shown by Chen et al. (2016) (proof omitted for space):

$$\begin{aligned}
\mathbb{E}\left[\#\text{ bad arm pulls}\right] \leq& m(l_T + 1) + \sum_{i=1}^{m} 2mt^{-2}\\
\leq& \frac{6m\log T}{(f^{-1}(R_{\text{min}}))^2} + m + \frac{\pi^2}{3}m\ .
\end{aligned}$$

We then bound the cumulative regret by attributing the maximum regret to each of the bad arm pulls, which yields:

$$\text{Reg}(T) \leq \left(\frac{6m\log T}{(f^{-1}(R_{\text{min}}))^2} + m + \frac{\pi^2}{3}m\right) R_{\text{max}}\ . \qquad \square$$

Proof of Theorem 4. As shown by Chen et al. (2016), the first term in Eq. 14 can be further broken down by having a finer regret attribution. Now we assign a maximum regret to all the bad super arm pulls, while we can in fact more carefully analyze the cumulative regret by counting the number of bad arm pulls resulting in a certain amount of regret. The only part different between ours and Chen et al. (2016) is the last term in Eq. 14, which is not affected by the additional analysis. So we conclude that the same argument still applies, where when the bounded smoothness function is of the form $f(x) = \gamma x^\omega$, we can get a similar result:

$$\text{Reg}(T) \leq \frac{2\gamma}{2-\omega}(6m\log T)^{\frac{\omega}{2}} T^{1-\frac{\omega}{2}} + \left(\frac{\pi^2}{3} + 1\right) mR_{\text{max}}\ . \qquad \square$$

Finally, we are ready to prove Theorem 5.

Theorem 5. *Given the minimum discretization gap Δ, number of targets N, Lipschitz constant L, and time horizon T, the regret bound of Algorithm 1 with* UCB *is*

$$\textit{Reg}_\Delta(T) \leq O\left(NL\Delta T + \sqrt{N^3\Delta^{-1}T\log T} + N^2L\Delta^{-1}\right) \tag{7}$$

and the regret bound of Algorithm 2 with UCB *is*

$$\textit{Reg}(T) \leq O\left(L^{\frac{4}{3}}NT^{\frac{2}{3}}(\log T)^{\frac{1}{3}}\right)\ . \tag{8}$$

Proof. Eqs. 7 and 8 can be proved by the same argument of the proof of Theorem 2 and 3. As shown in Eq. 13, we get:

$$\begin{aligned}
&\text{Reg}(T)\\
\leq& O\left(\frac{N^2}{L}\left(\frac{L^2T}{N\log T}\right)^{\frac{2}{3}}\log\left(\frac{T}{\log T}\right) + N^2L\left(\frac{L^2T}{N\log T}\right)^{\frac{1}{3}}\right)\\
\leq& O\left(L^{\frac{1}{3}}NT^{\frac{2}{3}}(\log T)^{\frac{1}{3}} + L^{\frac{5}{3}}N^{\frac{5}{3}}T^{\frac{1}{3}}(\log T)^{-\frac{1}{3}}\right)\\
=& O\left(L^{\frac{1}{3}}NT^{\frac{2}{3}}(\log T)^{\frac{1}{3}}\right)
\end{aligned} \tag{16}$$

$\square$

C Experiment details

The features associated with each target in Srepok Wildlife Sanctuary are distance to roads, major roads, rivers, streams, waterholes, park boundary, international boundary, patrol posts, and villages; forest cover; elevation and slope; conservation zone; and animal density of banteng, elephant, muntjac, pig.

For LIZARD, CUCB, and zooming, we speed up the confidence radius from Eq. 2 as $r_t(i,j) = \sqrt{\frac{\epsilon}{2n_t(i,j)}}$ with $\epsilon = 0.1$ to address the limited time horizon, which we empirically found to improve performance (no choice of ϵ affects the order of the algorithm's performance).