Non-Stationary Bandits and Meta-Learning with a Small Set of Optimal Arms

MohammadJavad Azizi
azizij@google.com
Google

Thang Nhat Duong
thangduong@arizona.edu
The University of Arizona

Yasin Abbasi Yadkori
yadkori@google.com
Google DeepMind

András György
agyorgy@google.com
Google DeepMind

Claire Vernade
claire.vernade@uni-tuebingen.de
University of Tuebingen

Mohammad Ghavamzadeh
ghavamza@amazon.com
Amazon AGI

Abstract

We study a sequential decision problem where the learner faces a sequence of K-armed bandit tasks. The task boundaries might be known (the bandit meta-learning setting), or unknown (the non-stationary bandit setting). For a given integer $M \leq K$, the learner aims to compete with the best subset of arms of size M. We design an algorithm based on a reduction to bandit submodular maximization and show that for T time steps comprised of N tasks, in the regime of large N and small number of optimal arms M, its regret in both settings is smaller than the simple baseline of $\tilde{O}(\sqrt{KNT})$ that can be obtained by using standard algorithms designed for non-stationary bandit problems. For the bandit meta-learning problem with fixed task length τ, we show that the regret of the algorithm is bounded as $\tilde{O}(NM\sqrt{M\tau} + (M^4KN^2)^{1/3}\tau)$. Under some additional assumptions on the identifiability of the optimal arms in each task, we show a bandit meta-learning algorithm with an improved $\tilde{O}(N\sqrt{M\tau} + (NMK^{1/2})^{1/2}\tau^{3/4})$ regret, where the order of the leading term (the first term) is optimal up to logarithmic factors, and the algorithm does not need the knowledge of M, N, and T in advance.

1 Introduction

Recommendation platforms interact with customers and must discover which items in their large catalog give maximum satisfaction to each user. These interactions are often sequential and each can be modeled as a multi-armed bandit problem. When a recommendation platform targets a new sub-population (e.g., demographic) of customers, its administrators may naturally assume that only a small subset of their large catalog of items would be attractive to this new group of users. Under this assumption, it would be beneficial for the platform to identify this subset as soon as possible, and then narrow down its exploration within the subset instead of over the entire catalog. This problem can be naturally modeled as *meta-learning*, where each task is an instance of a multi-armed bandit problem and the similarity between the tasks is in the existence of a subset of arms (items) such that at least one of them has a high expected reward, or is even optimal, in every single task (for the customers in the sub-population). In this paper, we study this meta-learning problem and its extensions to the *non-stationary* setting (e.g., when the users' affinity changes within a session).

Formally, we consider the problem where a learner faces N instances of a K-armed bandit task sequentially. For simplicity, we assume that the tasks are of equal length and each task lasts for τ rounds (for some positive integer τ), and therefore the total duration of the game is $T = N\tau$. At the beginning of task $n \in [N]$,[1] an adversary chooses the mean reward vector of the arms, $r_n \in [0,1]^K$.

[1] For any integer K, we let $[K] = \{1, \ldots, K\}$, and for any (multi-)set S, denote by $|S|$ the number of distinct elements in S.

Then, the learner interacts with the bandit task specified by this mean reward vector for τ time steps: at time $t \in [T]$ belonging to task $n \in [N]$,[2] if the learner takes an action[3] $a \in [K]$, it receives a reward signal $r_n(a) + \eta_{n,t}(a)$, where for all n and t, the $\eta_{n,t}(a)$ are independent zero-mean, $[-1/2, 1/2]$-valued noise variables, so that the expected reward is $r_n(a)$. We denote by a_n^* the optimal arm in task n, and by $\mathcal{H}_{n,t}$ the history of the actions taken and rewards observed by the learner up to, but not including, time step t in task n. At time step t, the learner computes a distribution $\pi_{n,t}$ over the actions as a function of $\mathcal{H}_{n,t}$ and some other parameters of the problem, samples an action $A_{n,t}$ from $\pi_{n,t}$, and plays it. Later, we will relax these assumptions and discuss extensions to the cases where reward functions can change within a task, tasks have different lengths, and task boundaries might be unknown.

We are interested in minimizing the worst-case T-step *dynamic regret* of the learner relative to the set of optimal arms, defined as

$$R_T^{\mathrm{ml}} = \sup_{(r_n)_{n=1}^N} \quad \mathbb{E}\left[\sum_{n=1}^{N}\sum_{t=1}^{\tau} r_n(a_n^*) - r_n(A_{n,t})\right], \tag{1}$$

where the expectation is taken over the learner's random actions that may depend on the realization of the noise in the observed rewards, as well as any potential internal randomization of the learner.

We mainly consider problems where the set of optimal arms is small, which, in our recommendation example, corresponds to the case where only a small subset of the large catalog of items is attractive to users. More formally, we assume (for now) that

$$|\{a_n^*\}_{n=1}^N| \le M \tag{2}$$

for some $M < K$. We call such problems *sparse bandit meta-learning* problems. We are interested in designing algorithms that can exploit such sparsity structure.

A near-optimal solution under an identifiability assumption. In Section 2, we consider the *sparse bandit meta-learning* problem under some *identifiability assumption* for the optimal arms of each task, namely, that in each task the gap between the rewards of the optimal and second best arms is large enough. We propose two algorithms (both defined in Algorithm 1) with near-optimal performance guarantees. Our algorithms do not need N, T, M as input.

The algorithms are hierarchical: First, a top level algorithm is used to learn the best subset of arms of cardinality at most M,[4] which determines the set of arms to be used in the next bandit task. This algorithm is either in an exploration or exploitation mode. In exploration mode, a best-arm-identification (BAI) algorithm is run in the next bandit task, which, given the identifiability assumption, finds the best arm in the task (out of all K arms), while in exploitation mode, using the information about which arms were found to be the best in previous tasks, it selects an M-subset of arms and runs a base bandit algorithm in the next task (the difference between our two algorithms is how they perform this step). In the example of a recommendation platform serving many customers, this means that the recommender system selects a small collection of items for each user and tries to find the best item, or sequence of items, in the catalog for that user.

Denoting the minimax regret of the base algorithm in a k-armed bandit task of length τ by $B_{\tau,k}$ (without additional assumptions, typical "good" bandit algorithms achieve $B_{\tau,k} = \widetilde{O}(\sqrt{\tau k})$), an ideal algorithm for the sparse bandit meta-learning problem, which knows the set of optimal arms and runs the base algorithm with this arm set on every task, can achieve an $O(NB_{\tau,M})$ dynamic regret, improving the potentially much larger $O(NB_{\tau,K})$ regret achievable without Equation (2). A naive reduction to bandit submodular maximization would yield only an $\widetilde{O}(NB_{\tau,M} + N^{2/3})$ regret bound. In comparison, our first, computationally efficient algorithm `G-BASS` achieves an $O(NB_{\tau,M(1+\log N)}) + \widetilde{O}(N^{1/2})$ regret, paying a small $\log N$ factor for computational efficiency in the

[2] Note that $n = \lfloor (t-1)/\tau \rfloor + 1$.

[3] We use the terms "arm" and "action" interchangeably.

[4] Throughout we will call a subset of size at most M an *M-subset*.

leading term. Importantly, G-BASS does not need a priori knowledge of M. To remove the extra $\log N$ factor, we propose another method, E-BASS, which achieves the desired $O(NB_{\tau,M}) + \widetilde{O}(N^{1/2})$ regret, at the cost of an increased computational complexity, potentially exponential in M. Although G-BASS is similar to the greedy solution in offline submodular maximization, the proof technique we use for it is quite different, exploiting the special structure of the problem and resulting in an improved regret guarantee.

A general solution by a reduction to bandit subset selection. We also present a solution that is applicable even without an identifiability assumption. However, this solution requires N, T, M as input. Nevertheless, the results improve existing solutions in this setting as shown in Table 1.

Our general solution is in fact applicable even in an *agnostic* setting, as we explain next. In our solution, the learner competes with a sequence of arms $(a_n)_{n=1}^N$ that has at most $M \le K$ distinct elements $|\{a_n\}_{n=1}^N| \le M$. In this formulation, M is the learner's choice and indicates its prior belief about the number of *good* arms (in terms of having high expected reward) in the sequence of N bandit tasks (with the reward sequence $(r_n)_{n=1}^N$) that it is supposed to solve. We call a reward sequence $(r_n)_{n=1}^N$ *realizable* if there exists a set of arms of size at most M that contains an optimal arm (an arm with reward $\max_{a\in[K]} r_n(a)$) for every task $n \in [N]$. If this cannot be guaranteed, we refer to the setting as *agnostic*. In our motivating example of recommender systems, the realizable case is when there exists a set of at most M items that contains the most desirable item for every single customer (which may not be unique). This is obviously a strong assumption and may not hold in many cases. Instead, a more realistic setting is when there exists a subset of items of size at most M that contains a good (but not necessarily the best) item for most of the customers. This is an example of the agnostic case.

In fact, we obtain stronger results and derive regret bounds for the more general *adversarial* setting where the reward vector can change within each task. In this setting, we define the regret as

$$R_T = \sup_{\substack{(r_{n,t})_{n=1,t=1}^{N,\tau},\\ (a_n)_{n=1}^N \,:\, |\{a_n\}_{n=1}^N| \le M}} \mathbb{E}\left[\sum_{n=1}^{N}\sum_{t=1}^{\tau} r_{n,t}(a_n) - r_{n,t}(A_{n,t})\right], \tag{3}$$

where the learner competes with the best M-subset of arms across a sequence of N “adversarial" bandit tasks. Note that in the stochastic meta-learning setting, under assumption (2), this definition simplifies to that of R_T^{ml} given in (1) (with the expectation also corresponding to the noise in the reward function). Therefore, throughout we will use R_T to denote the regret.

To solve this more general problem, we take the same approach as before, and use a bi-level algorithm with a subset-selection method on top. However, since now we do not assume that the best arms can be identified in any task (and we do not even assume that the set of best arms is of cardinality at most M), we cannot apply the explicit exploration step with a BAI algorithm as before. Instead, for every task, the subset-selection algorithm selects an M-subset of arms and runs a base bandit algorithm with this set as its action space for τ steps, which returns the total reward obtained to the top algorithm as feedback. In Section 3.2, we present our resulting algorithm, called OS-BASS (Algorithm 2) and show that its regret scales as $\widetilde{O}((M^4KN^2)^{1/3}\tau + MN\sqrt{M\tau})$. The OS-BASS algorithm is based on the aforementioned reduction to the bandit subset-selection problem, uses a bandit submodular optimization method, and needs to know the number of change points N in advance. Without any restriction on the set of optimal arms, the optimal rate for the non-stationary bandit problem is $\widetilde{O}(\sqrt{KNT})$ (Auer et al., 2019b; Chen et al., 2019; Wei and Luo, 2021).[5] This bound can also be written as $\widetilde{O}(N\sqrt{K\tau})$. For small N (large τ), this baseline rate cannot be improved. Therefore, we are mainly interested in the regime of large number of tasks and small number of optimal arms for which our regret bound is better than that for the baseline in the *realizable* case.

Sparse non-stationary setting. We also study the more general *non-stationary setting* where the task boundaries are unknown (Russac et al., 2019; Auer et al., 2019a; Hong et al., 2020b; Wei

[5] This rate can be achieved by algorithms such as AdSwitch (Auer et al., 2019b) without any prior knowledge of T or N.

Table 1: A comparison of our results with existing results in literature. The order of regret bounds is given up to logarithmic factors (meta/non-st. refer to the meta-learning/non-stationary bandit settings, adv./stoch. refer to the adversarial/stochastic rewards, respectively).

Algorithm	Setting	Tasks	Comparator	Prior knowledge	Regret bounds
Independent EXP3 for each task	meta	adv.	n/a	none	$\sqrt{KNT} = N\sqrt{K\tau}$
EXP3.S (Auer et al., 1995)	non-st.	adv.	n/a	N	$\sqrt{KNT} = N\sqrt{K\tau}$
Zheng et al. (2019)	non-st.	adv.	agnostic	M, N	$(sN)^{1/3}(MT)^{2/3} + M\sqrt{sT}\log K + MK^3$
Balcan et al. (2022)	meta	adv.	agnostic	none	$NB_{\tau,M} + N^{1-\frac{1}{6\log K}}$
G-BASS (under identifiability assumptions)	meta	stoch.	realizable	none	$NB_{\tau,M} + \sqrt{MKB_{\tau,K}N\tau} + o(\sqrt{N})$
OS-BASS	meta	adv.	agnostic	M, N, T	$(M^4KN^2\log K)^{1/3}\tau + MN\sqrt{M\tau}$
OS-BASS	non-st.	stoch.	agnostic	M, N, T	$(MKN^2\log K)^{1/3}T/N + M\sqrt{MTN}$

and Luo, 2021; Suk and Kpotufe, 2022; Abbasi-Yadkori et al., 2023). A single user interacting with a recommender system is an example of the non-stationary setting (as the system does not necessarily know when the user's mood/affinity changes), while a recommendation platform serving many customers is an example of the meta-learning setting (we observe when the current session ends and a new customer arrives). More formally, in the non-stationary setting, the learner knows the number of tasks N and the horizon T before the game begins, but the start and duration of the tasks, $\{\tau_n\}_{n=1}^N$, are unknown.

Related work. Our solution is based on online subset selection, with many connections to the online learning and bandit literature. We discuss the most relevant works here; other related papers are discussed in Appendix G. Table 1 shows a comparison between our results and existing results in literature.

The sparse non-stationary bandit problem is the bandit variant of experts problem with small set of optimal arms whose study goes back to Bousquet and Warmuth (2002). The only result in the bandit setting that we are aware of is the work of Zheng et al. (2019), who show an algorithm for competing against a small set of optimal arms in an adversarial setting with sparse reward vectors. The solution of Zheng et al. (2019) has similarities with our approach, as both solutions are reduction-based and employ a meta-learner that plays with base algorithms. Similarly to our approach, the meta-learner of Zheng et al. (2019) needs to satisfy a static regret guarantee, while the base algorithm needs to satisfy a dynamic regret guarantee. The algorithm of Zheng et al. (2019) also requires the number of change points N and subset size M as input and its dynamic regret is $\tilde{O}((sN)^{1/3}(MT)^{2/3} + M\sqrt{sT\log K} + MK^3\log T)$, where $s = \max_n \|r_n\|_0$ is the number of non-zero elements in reward vectors. Notice that without this sparsity condition, the above dynamic regret is worse than the $\tilde{O}(\sqrt{KNT})$ regret of EXP3.S of Auer et al. (1995), a variant of the well-known EXP3 algorithm (given also by Auer et al., 1995) which is designed for the non-stationary adversarial multi-armed bandit setting, to compete with a sequence of arms with a given maximum number of switches (i.e., minimizing the regret in (3) for $M = K$). When s is a small constant, the above bound improves upon the regret of EXP3.S when $M^{4/3}(T/N)^{1/3} < K < (TN/M^2)^{1/5}$.

Balcan et al. (2022) studies bandit meta-learning problems with adversarial bandit tasks. They introduce a meta-learning algorithm that tunes the initialization and step-size of the online mirror decent base algorithm. Balcan et al. (2022) show that their algorithm achieves $\min_{\beta\in(0,1]} \tilde{O}(N\sqrt{H_\beta K^\beta \tau/\beta} + N^{1-\beta/6})$ total regret, where H_β is a notion of entropy. When a subset of size M contains the optimal arms of most tasks, then by the choice of $\beta = 1/\log(K)$ the regret bound simplifies to $\tilde{O}(N\sqrt{M\tau} + N^{1-1/(6\log K)})$. Compared to our results in the most general setting (cf. Section 3), their convergence rate of $\tilde{O}(1/N^{1/(6\log K)})$ is much slower than our convergence rate of $\tilde{O}(1/N^{1/3})$. However, their task-averaged regret is $O(\sqrt{M\tau})$, whereas it is $O(M\sqrt{M\tau})$ in our case. Finally, the regret bound of Balcan et al. (2022) adaptively holds for the best value of M, while M is an input to our algorithm. However, under the identifiability condition, our algorithm G-BASS has a better regret guarantee (matching the task-averaged regret while significantly improving on the other term), and it also automatically adapts to the problem parameters.

Meta-, multi-task, and transfer learning (Baxter, 2000; Caruana, 1997; Thrun, 1996) are related machine learning problems concerned with learning some shared information across tasks. In that sense, our work is connected to other theoretical studies (Franceschi et al., 2018; Denevi et al., 2018b;a; 2019; Kong et al., 2020; Khodak et al., 2019; Tripuraneni et al., 2021) though indeed we focus on the bandit learning setting. Various other ways of modelling structure have been proposed and studied in bandit meta-learning. A special case of our problem was studied by Azar et al. (2013) where K-armed bandit problems are sampled from a prior over a finite set of tasks. Park et al. (2021) consider a continual learning setting where the bandit environment changes under a Lipschitz condition. Kveton et al. (2020) observe that the hyperparameters of bandit algorithms can be learned by gradient descent across tasks. Learning regularization for bandit algorithms (Kveton et al., 2021; Cella et al., 2020) are also proposed, building on the biased regularization ideas from Baxter (2000). Interestingly, these contextual problems are also connected to latent and clustering of bandit models (Maillard and Mannor, 2014; Gentile et al., 2014; Hong et al., 2020a;b).

2 Sparse meta-learning under an identifiability condition

In this section, we study our sparse meta-learning setting in the *realizable* case under an identifiability assumption (Assumption 2.1) that the learner has access to an exploration method that reveals optimal actions. We further assume that the tasks are of equal length τ.

Assumption 2.1 (Efficient Identification)**.** *There exists a set of M arms that has a non-empty intersection with the set of optimal arms in each task. Also, the learner has access to a best-arm-identification (BAI) procedure that for some $\delta \in [0,1]$, with probability at least $1-\delta/N$, identifies the set of optimal arms if executed in a task (for at most τ steps).*

The assumption requires the BAI procedure to return only optimal arms. This choice is for simplicity and could easily be relaxed to allow it to return all arms with sub-optimality gap smaller than $\Theta(\sqrt{M\log(N/\delta)/\tau})$.

Assumption 2.1 is a special case of the priced feedback model in Streeter and Golovin (2007). If for any task n with optimal arms $S_n^* \subset [K]$, we have $r_n(a_n^*) - \max_{a\notin S_n^*} r_n(a) \geq \Delta$, $\forall a_n^* \in S_n^*$ (note that $r_n(a_n^*)$ is the same for all $a_n^* \in S_n^*$) for some $\Delta = \Theta(\sqrt{K\log(N/\delta)/\tau})$, a properly tuned *phased elimination* (PE) procedure (Auer and Ortner, 2010) returns the set of optimal arms with probability at least $1-\delta/N$. The cumulative worst-case regret of PE in a task with K arms is $B'_{\tau,K} = \Theta(B_{\tau,K})$ (Auer and Ortner, 2010), see (Lattimore and Szepesvári, 2020, Exercise 6.8) for details. With a slight abuse of notation, in this section, we use $B_{\tau,K}$ to denote $\max\{B_{\tau,K}, B'_{\tau,K}\}$.

We disentangle exploration (Exr) and exploitation (Ext) at a meta-level. In Exr mode, the learner executes a BAI on all arms and (by Assumption 2.1) with high probability observes the set of optimal actions S_n^*. The price of this information is a large regret denoted by $\mathbf{C}_{\texttt{info}}$, which for a properly tuned PE, we know $\mathbf{C}_{\texttt{info}} = B_{\tau,K} > B'_{\tau,M}$. So, since we aim for $R_T \leq \widetilde{O}(NB_{\tau,M}) + o(N)$, we should keep the number of Exr calls small. In Ext mode, the learner executes a base bandit algorithm on a chosen subset S_n constructed using the previously identified optimal actions $\mathcal{I}_n = \bigcup_{j<n:E_j=\textsc{Exr}}\{S_j^*\}$.

If $S_n \cap S_n^* \neq \emptyset$, the regret of Base is bounded by $\mathbf{C}_{\texttt{hit}} = B_{\tau,s_n}$, where $s_n = |S_n|$. Otherwise, since the performance gap between the optimal arms and the arms in S_n can be arbitrary, the regret in

Algorithm 1 BASS: BAndit Subset Selection *(for the meta-learning setting)*

```
Options: Greedy G-BASS (G), Elimination-based E-BASS (E)
Input: Base (an efficient $K$-armed bandit algorithm), BAI (a best arm identification algorithm),
    Exr probabilities $p_n$, (E) subset size $M$
Initialize: Let (G) $\mathcal{I}_0 = \emptyset$; (E) $\mathcal{X}_0$ be the set of all $M$-subsets of $[K]$.
for $n = 1, \dots, N$ do
    Set $E_n = $ Exr w.p. $p_n$, otherwise set $E_n = $ Ext
    if $E_n = $ Exr or $n = 1$ then
        Run BAI on all arms of task $n$ and observe the best arms $S_n^*$
        (G) Set $\mathcal{I}_n = \mathcal{I}_{n-1} \cup \{S_n^*\}$
        (E) Set $\mathcal{X}_n = \{S \in \mathcal{X}_{n-1} : S \cap S_n^* \neq \emptyset\}$, i.e., elements of $\mathcal{X}_{n-1}$ with non-empty overlap with
            $S_n^*$
    else
        (G) Find $S_n$ by Greedy s.t. $\forall S \in \mathcal{I}_n, S_n \cap S \neq \emptyset$
        (E) Sample $S_n$ uniformly at random from $\mathcal{X}_n$
        Run the Base algorithm on $S_n$
    end if
end for
```

task n can be as large as $\mathbf{C}_{\texttt{miss}} = \tau$. Note that to keep $\mathbf{C}_{\texttt{hit}}$ small, the subset S_n should be as small as possible. Ideally, S_n should be a subset of size M that has non-empty overlap with all members of $\mathcal{I}_n$. However, the problem of finding such S_n is the so-called *hitting set* problem, which is known to be NP-Complete (Feige et al., 2004).

A simple greedy algorithm can be used to obtain an approximate solution efficiently (see, e.g, Streeter and Golovin, 2007): The greedy algorithm builds a subset incrementally and in each stage, adds the action that is optimal for the largest number of remaining tasks. It has polynomial computation complexity and finds a subset of size at most $M(1 + \log N)$ that contains an optimal action for each task. We say an action $a \in [K]$ covers task j if $a \in S_j^*$. The greedy method, denoted by Greedy, starts with an empty set and at each stage, it adds the action that covers the largest number of uncovered tasks in $\mathcal{I}_n$, until all tasks are covered.

We also propose E-BASS, that is based on an elimination procedure: the learner maintains an active set of possible M-subsets compatible with the Exr history, and eliminates all subsets that are inconsistent with $\mathcal{I}_n$. In the Ext mode, a subset is selected uniformly at random from the set of active subsets. As we will show, this algorithm improves the regret by a factor of $\log N$, but is not computationally efficient as it needs to sample from an exponentially large collection of active subsets.

The analysis of the greedy solution, G-BASS, depends on the *cost-to-go* function of the following game between the learner and the environment. At segment n, the learner may choose $E_n \in \{\text{Exr}, \text{Ext}\}$ with probability $p_n = P(E_n = \text{Exr})$ and cost $\mathbf{C}_{\texttt{info}}$. The environment may choose a best arm a_n^* that the learner already knows about which costs $\mathbf{C}_{\texttt{hit}}$ (i.e., $a_n^* \in S_n$) or choose an optimal arm set S_n^* so that $S_n^* \cap S_n = \emptyset$, with cost $\mathbf{C}_{\texttt{miss}}$. Let $q_n = P(S_n^* \cap S_n = \emptyset)$. The regret of G-BASS is bounded by the cost of the learner in this simple game, if we assume $\delta = 0$ in Assumption 2.1. The learner is a (randomized) function of $\mathcal{I}$, hence we can write the minimax cost-to-go function as

$$\begin{aligned} V_N(\mathcal{I}) &= 0, \\ V_n(\mathcal{I}) &= \min_p \max_q \big\{ p\mathbf{C}_{\texttt{info}} + q(1-p)\mathbf{C}_{\texttt{miss}} \\ &\quad + (1-q)(1-p)\mathbf{C}_{\texttt{hit}} + (1-pq)V_{n+1}(\mathcal{I}) + pqV_{n+1}(\mathcal{I} \cup \{S_n^*\}) \big\}, \qquad \text{for} \quad n < N. \end{aligned} \tag{4}$$

For the last equality note that when the environment reveals a new action (happens with probability q) and the learner explores (with probability p), its current knowledge set $\mathcal{I}$ is incremented. The optimal cost-to-go function V_n in (4) corresponds to the case of $\delta = 0$ in Assumption 2.1, and $V_0(\emptyset)$

gives the minimax regret for the family of algorithms with the limited choice described. Therefore, when the BAI algorithm is successful, we have $R_T \le V_0(\emptyset)$, almost surely. For $\delta > 0$, using a union bound, we can show $R_T \le V_0(\emptyset) + \delta N\tau$. Setting $\delta = 1/(N\tau)$ ensures that $\delta N\tau$ is negligible. Finally, if the BAI algorithm only returns a set of approximately optimal arms satisfying $r_n(a) \ge r_n(a_n^*) - \Delta$ for all arms a selected, the meta-regret can be bounded trivially as $R_T \le V_0(\emptyset) + (\delta + \Delta)N\tau$.

Case 1 (a unique optimal arm): Before going to the general case, we first consider the case where there is a unique and identifiable optimal arm in each task.

Assumption 2.2 (Unique Identification)**.** *Assumption 2.1 holds, and each task has a unique best arm.*

Theorem 2.3. *Under the Unique Identification assumption,*

$$V_0(\emptyset) \le NB_{\tau,M} + M\sqrt{2(\boldsymbol{C}_{info} - \boldsymbol{C}_{hit})(\boldsymbol{C}_{miss} - \boldsymbol{C}_{hit})N}.$$

Therefore, under Assumption 2.2, the regret of `G-BASS` *with $p_n = \Theta(1/\sqrt{N-n})$ satisfies $R_T \le NB_{\tau,M} + M\sqrt{B_{\tau,K}N\tau} + \delta N\tau$.*

We prove Theorem 2.3 in Appendix B.1 by solving the min-max problem in (4) for V_n. Interestingly, the exploration probability p_n increases as $\Theta(1/\sqrt{N-n})$. This might seem *counter-intuitive* at first as typically the exploration rate decreases in most online learning algorithms. The intuition is that as n gets closer to N, if $s_n < M$, the adversary has less remaining budget left to make the learner suffer a big cost. Therefore, the adversary increases its probability of choosing a *new* optimal arm, and thus, the learner needs to explore more.

Case 2 (general case): We now consider the general case with potentially multiple optimal arms in each task.

Theorem 2.4. *Let $M' = M(1 + \log N)$ and Assumption 2.1 holds. Then, the regret of* `G-BASS` *with exploration probability $p_n = \sqrt{\frac{|S_n|K\tau}{nB_{\tau,K}}}$ is bounded as*

$$R_T \le NB_{\tau,M'} + MB_{\tau,K} + \sqrt{M'KB_{\tau,K}N\tau} + \delta N\tau \; .$$

The proof is similar in spirit to that of Theorem 2.3 and is deferred to Appendix B.3. This regret guarantee holds in the realizable setting, but `G-BASS` does not need M as input. The next theorem (proof in Appendix E.1) shows that the regret of `E-BASS` is bounded as $NB_{\tau,M} + o(N)$, which is smaller than that for `G-BASS` by a factor of $\log N$. However, `E-BASS` is not computationally efficient and also requires M as input.

Theorem 2.5. *Let Assumption 2.1 holds. Then, the regret of the* `E-BASS` *algorithm with exploration probability $p_n = \left(\frac{\tau}{K}\right)^{1/4}\sqrt{\frac{\log K}{N}}$ is bounded as $R_T \le NB_{\tau,M} + O(\tau^{3/4}K^{1/4}\sqrt{NM\log K})$.*

Remark 2.6 (connections to partial monitoring games)**.** *The setting of this section can be viewed more generally as a partial monitoring game. Partial monitoring is a general framework in online learning that disentangles rewards and observations (information). In our bandit meta-learning problem, different actions of the meta-learner (EXR and EXT) provide different levels of information and have different costs. Thus, the problem can be reduced to a partial monitoring game on $\mathcal{X}$, the set of M-subsets of $[K]$. More details are in Appendix E.*

3 Sparse meta-learning without identifiability assumptions

As explained in introduction, our general approach to solve the problems studied in this paper is to reduce them to the bandit subset-selection problem. This approach is applicable even if each task is an adversarial bandit problem and we are in an agnostic setting. However, to keep the presentation simple, we will consider stochastic bandit tasks in a realizable setting.

3.1 Reduction to subset selection and bandit submodular maximization

In task n, the learner selects a subset of arms $S_n \in \mathcal{S} = \{S : S \in 2^{[K]}, |S| \leq M\}$, runs a base bandit algorithm on that subset for τ steps, and receives the pseudo-reward[6]

$$\sum_{t=1}^{\tau} r_n(A_{n,t}) = \tau \max_{a \in S_n} r_n(a) - \sum_{t=1}^{\tau} \big(\max_{a \in S_n} r_n(a) - r_n(A_{n,t}) \big) := f_n(S_n) - \tau \varepsilon_n,$$

where $A_{n,t}$ is the learner's action in time step t of task n, $f_n(S) := \tau \max_{a \in S} r_n(a)$ being the max-reward function for the set of arms S and reward function r_n, and finally ε_n is the average "noise" per time step observed by the learner. It is easy to see that f_n is a submodular function. See Appendix A for definitions.

We require the base algorithm to have a guarantee for the regret, measured relative to the optimal action for every time step,

$$\tau \mathbb{E}[\varepsilon_n] = \sup_{r_n} \mathbb{E}\left[\sum_{t=1}^{\tau} \max_{a \in S_n} r_n(a) - r_n(A_{n,t}) \right] \leq B_{\tau,M} \ . \tag{5}$$

We can bound the regret of any method that solves the sparse bandit meta-learning problem using the above reduction to subset selection as follows (proof in Appendix C):

Lemma 3.1. *The regret of any policy running a base algorithm that satisfies (5) in each task $n \in [N]$ on a selected subset of arms $S_n \in \mathcal{S}$ can be bounded as*

$$R_T \leq \sup_{f_1, \dots, f_N \in \mathcal{F}} \max_{S \in \mathcal{S}} \mathbb{E}\Big[\sum_{n=1}^{N} f_n(S) - f_n(S_n) + B_{\tau,M} \Big].$$

This way we reduce our sparse bandit meta-learning problem to minimizing a notion of regret where in each task $n \in [N]$, the learner selects a subset $S_n \in \mathcal{S}$ and observes (pseudo)-reward $f_n(S_n) - \tau \varepsilon_n$. Since f_n is submodular, this reduction allows us to leverage the literature on online submodular maximization to obtain a bound on R_T.

Bandit submodular maximization. Streeter and Golovin (2007) studied the online submodular maximization problem in four different settings: (i) the full-information setting where the function f_n is fully observed at the end of each segment n; (ii) a partially transparent model where the value of f_n is revealed for some subsets; (iii) the priced feedback model where the learner can observe f_n by paying a price; and (iv) the bandit setting where only $f_n(S_n)$ is observed. They proved $O(\sqrt{N})$ and $O(N^{2/3})$ regret bounds for the first two and the last two settings, respectively. Radlinski et al. (2008) proposed an algorithm similar to the one in Streeter and Golovin (2007) for a particular ranking problem in the partially transparent feedback setting, and also obtained an $O(\sqrt{N})$ regret bound. The priced feedback setting (iii) is similar to problems where the best arms can be identified in every task, which we study in Section 2.

3.2 A general solution based on submodular maximization

In this section, we present an algorithm, called `OS-BASS` (Online Submodular BAndit Subset Selection), for our sparse meta-learning problem. The approach is in fact more general, and as we will show in the next section, it can be applied in the sparse non-stationary setting as well. `OS-BASS`, whose pseudo-code is shown in Algorithm 2, is based on the reduction to subset selection described in Section 3.1 and the `OG°` algorithm (for bandit submodular maximization) by Streeter and Golovin (2007). It requires the knowledge of the number of tasks N and time horizon T. Note that while N and T are environment parameters, M and the expert algorithms are selected by the learner.

`OS-BASS` also takes a `Base` algorithm as input. We choose `Base` to be an algorithm designed to solve multi-armed bandit problems such as the well-known UCB (Auer et al., 2002) when the tasks are stochastic bandits, or EXP3 (Auer et al., 1995) when the tasks are adversarial bandits.

[6] We refer to it as pseudo-reward since it uses the mean rewards and not the actual random rewards.

Algorithm 2 OS-BASS: Online Submodular BAndit Subset Selection *(applicable to the sparse non-stationary/meta-learning settings and agnostic/realizable cases)*

```
Input: number of tasks $N$, time horizon $T$, subset size $M$, expert algorithms $\mathcal{E}_1, \cdots, \mathcal{E}_{\tilde{M}}$
Set segment length to $\tau = T/N$
for $n \in [N]$ do
  for $i \in [\tilde{M}]$ do
    Select an arm $a_i$ using $\mathcal{E}_i$
  end for
  Define subset $S_n = \{a_1, \ldots, a_{\tilde{M}}\}$
  Set $E_n =$ EXR w.p. $\gamma_n$, otherwise set $E_n =$ EXT
  if $E_n =$ EXT then
    Run Base on subset $S_n$ for $\tau$ steps
  else
    Select an index $i \in [\tilde{M}]$ uniformly at random
    Select a new arm $a'_i \in [K]$ uniformly at random
    Define subset $S_{n:i} \leftarrow \{a_1, \ldots, a_{i-1}, a'_i\}$
    Run Base on subset $S_{n:i}$ for $\tau$ steps
    Give average reward over the segment as a reward to $\mathcal{E}_i$ for arm $a'_i$, and give zero reward
      for all other arms and experts
  end if
end for
```

OS-BASS applies $\tilde{M} = \lceil M \log N \rceil$ expert algorithms (i.e., regret minimization algorithms in the full information setting), denoted by $\mathcal{E}_1, \ldots, \mathcal{E}_{\tilde{M}}$, to the K arms, where the role of $\mathcal{E}_i$ is to learn the i'th "best" arm. In each task $n \in [N]$, each expert $\mathcal{E}_i$, $\forall i \in [\tilde{M}]$, selects an arm $a_i \in [K]$. The requirement for an expert algorithm is that it should achieve an $O(\sqrt{v \log(K)})$ regret over v time steps relative to the best action selected in hindsight. This can be achieved by all standard expert algorithms, such as exponential weights (Cesa-Bianchi and Lugosi, 2006). Then, with probability $1 - \gamma_n$, OS-BASS *exploits* (EXT) the set of $\tilde{M}$ arms selected by the experts, i.e., a Base algorithm is executed on this set for τ steps. With probability γ_n, OS-BASS *explores* (EXR), i.e., first a random index $i \in [\tilde{M}]$ and an arm $a'_i \in [K]$ are chosen uniformly at random, and then the Base algorithm is executed on the set $S_{n:i}$ consisting of i arms, the $i-1$ arms selected by $\{\mathcal{E}_1, \ldots, \mathcal{E}_{i-1}\}$ plus a'_i, for τ steps. In EXR, the exploring expert, $\mathcal{E}_i$, is updated with a reward equal to the average reward of Base in the τ-step task for arm a'_i (approximately $f_n(S_{n:i})/\tau$ up to error ε_n/τ), and zero for all other arms. All other experts are updated with reward zero for all the arms (this is simply to construct an unbiased estimate and to use importance sampling). This way, $\mathcal{E}_1, \ldots, \mathcal{E}_i$ jointly learn the identity of the best set of size i w.r.t. (approximate) reward $\sum_{n=1}^{N} f_n$,[7] and thus, approximate a greedy solution for finding the top M arms. Note that OS-BASS provides no reward for the experts during EXT. It is reasonable to expect that such reward can improve the empirical performance of the algorithm, but it is not going to improve its theoretical guarantees. OS-BASS could be viewed as a simulation of the offline greedy procedure that incrementally constructs its solution (Streeter and Golovin, 2007). When an index i is chosen, the algorithm is learning the i'th choice of the offline greedy procedure, and thus, it only plays a subset of the arms of size i.

Theorem 3.2. *In the sparse meta-learning setting with N bandit tasks of equal length $\tau = T/N$, the regret of OS-BASS with EXP3 as Base and $\gamma_n = \left(\frac{\tilde{M} K \log K}{n}\right)^{1/3}$ is $R_T = \tilde{O}\big((\tilde{M}^4 K N^2 \log K)^{1/3}\tau + \tilde{M} N \sqrt{\tilde{M}\tau}\big)$.*

The proof is in Appendix D.3. In the regime of large number of tasks N and small number of optimal arms M, our bound improves upon the $\tilde{O}(N\sqrt{K\tau})$ bound of the trivial solution of running an independent UCB (or EXP3) algorithm in each task.

[7] Note that the "real" reward of expert $\mathcal{E}_i$ for an action a is $f_n(S_{n:i-1} \cup \{a\}) - f_n(S_{n:i-1})$.

Note that the value of γ_n in Theorem 3.2 uses count n instead of the total number of tasks N. However, the value of N still appears in $\tilde{M}$, and thus, the algorithm still requires knowledge of N, although a knowledge of an upper bound on $\log N$ would be sufficient.

Remark 3.3. *The above result is also applicable in the adversarial setting where the rewards can change in every time step, and regret is measured with respect to the best arm in each task.*

Remark 3.4 (tasks with variable lengths)**.** *Consider problems with non-equal task lengths where the learner only gets to know the length of each task when it begins. To handle this situation, we construct an exponential grid for the task lengths with $b := \log(\max_n \tau_n) \le \log T$ buckets, where each bucket $i \in [b]$ is defined as $[\underline{\tau}_i := 2^{i-1}, \overline{\tau}_i := 2^i]$. We then run a copy of `OS-BASS` on the tasks falling in each bucket as they arrive. Let $N^{(i)}$ denote the number of tasks that fall in bucket i. Then by Theorem 3.2, the total regret satisfies $R_T \le \sum_{i=1}^b \tilde{O}\big(\tilde{M}^{4/3}(N^{(i)})^{2/3}K^{1/3}\overline{\tau}_i + N^{(i)}\tilde{M}^{3/2}\sqrt{\overline{\tau}_i}\big)$. Similarly, this regret is better than the simple baseline $\sum_{n=1}^N \sqrt{K\tau_n} = \sum_{i=1}^b N_i\sqrt{K\overline{\tau}_i}$ when $M = o(K^{1/3}/T^{1/5})$.*

4 The sparse non-stationary setting

In this section, we show that a similar reduction to subset selection can also be used in the more general non-stationary setting. However, instead of learning the subset containing the best arms task by task (whose boundaries are unknown in the non-stationary setting), we conveniently divide the time horizon T into N segments of equal length $\tau = T/N$ and learn the optimal arms segment by segment. This approach clearly requires the knowledge of the number of tasks N in advance. Note that the segments and tasks would coincide in the meta-learning setting when all tasks have equal length.

Consider the n'th segment, i.e., $[(n-1)\tau + 1, n\tau)$. Without loss of generality, we assume that a new task starts at the beginning of each segment. To satisfy this assumption we break the tasks that run over the end of their segment, which will result in at most $N-1$ extra new tasks (we will have at most $2N-1$ tasks). If we denote by N_n, the number of tasks in segment n, we may write $\sum_{n=1}^N N_n \le 2N-1$. Finally, we denote by $\tau_{n,u}$ and $r_{n,u}$, the length and mean reward of the u'th task in segment n.

We require the base algorithm to have a guarantee for the *dynamic regret*, measured relative to the sequence of optimal actions selected for every time step, defined as

$$\sup_{(r_n)_{n=i}^j} \mathbb{E}\left[\sum_{n=i}^{j}\sum_{t=1}^{\tau_n} \max_{a\in[K]} r_n(a) - r_n(A_{n,t})\right] \le B_{L,j-i,K}, \tag{6}$$

for any i and j such that $1 \le i \le j \le N$, and with $L = \sum_{n=i}^j \tau_n$. In (6), we naturally extend the upper-bound notation to indicate the number of task changes in the segment of length L, and typically have $B_{L,j-i,K} = \widetilde{O}(\sqrt{L(j-i+1)K})$. `AdSwitch` (Auer et al., 2019b) is an example of such a base algorithm that does not need to know the task boundaries and lengths (see Section 4.1).

In segment n, the learner selects a subset of arms $S_n \in \mathcal{S} = \{S : S \in 2^{[K]}, |S| \le M\}$, runs a base bandit algorithm on that subset for τ steps, and receives the pseudo-reward[8]

$$\begin{aligned}\sum_{u=1}^{N_n}\sum_{t=1}^{\tau_{n,u}} r_{n,u}(A_{n,u,t}) &= \sum_{u=1}^{N_n}\Big(\tau_{n,u}\max_{a\in S_n} r_{n,u}(a) - \sum_{t=1}^{\tau_{n,u}}\big(\max_{a\in S_n} r_{n,u}(a) - r_{n,u}(A_{n,u,t})\big)\Big)\\ &:= f_n(S_n) - \tau\varepsilon_n,\end{aligned}$$

where $A_{n,u,t}$ is the learner's action in time step t of task u of segment n, $f_n(S) := \sum_{u=1}^{N_n}\tau_{n,u}f_{n,u}(S)$ with $f_{n,u}(S) := \max_{a\in S} r_{n,u}(a)$ being the max-reward function for the set of arms S and reward function $r_{n,u}$, and finally ε_n is the average "noise" per time step observed by the learner. We prove in Appendix A that $f_n \in \mathcal{F}$, the family of submodular functions, since it is an affine combination of

[8] We refer to it as pseudo-reward since it uses the mean rewards and not the actual random rewards.

finitely many submodular functions. Moreover, since we assume that the base algorithm satisfies (6), we have $\tau\mathbb{E}[\varepsilon_n] \le B_{\tau,N_n,M}$.

We can bound the regret of any method that solves the sparse non-stationary bandit problem using the above reduction to subset selection as follows (proof in Appendix C):

Lemma 4.1. *The regret of any policy running a base algorithm that satisfies* (6) *in each segment* $n \in [N]$ *on a selected subset of arms* $S_n \in \mathcal{S}$ *can be bounded as*

$$R_T \le \sup_{f_1,\dots,f_N \in \mathcal{F}} \max_{S\in\mathcal{S}} \mathbb{E}\Big[\sum_{n=1}^{N} f_n(S) - f_n(S_n) + B_{\tau,N_n,M}\Big].$$

This way we reduce our sparse non-stationary bandit problem to a bandit subset-selection problem, where the decision space is the set of M-subsets of $[K]$ and the reward in each segment n (of the top algorithm) is the maximum cumulative reward in this segment over the chosen subset. For example, when $\tau \ll N$, i.e., there are relatively many tasks, an $o(N)$-regret for the bandit subset-selection problem translates to an $O(N\sqrt{M\tau}) + o(N)$ regret bound for our sparse non-stationary bandit problem. Note that the leading term in the regret is $O(N\sqrt{M\tau})$, which is an upper-bound on the regret of a bandit strategy that runs the base algorithm on the best subset in each segment, and is in fact the best possible (minimax) regret rate achievable given an optimal M-subset. The subset-selection problem is a bandit submodular maximization problem, for which `OG°`, an online greedy approximation algorithm by Streeter and Golovin (2007), achieves regret $O(N^{2/3})$.[9]

4.1 `OS-BASS` in the non-stationary setting

In this section, we consider the implementation of `OS-BASS` in the non-stationary setting (task change points are unknown and tasks can be of different lengths). As discussed earlier, our approach is to divide the time horizon into segments of equal length, and to test a different M-subset in each segment by running a `Base` bandit algorithm on that subset. To choose `Base`, we should note that in the non-stationary bandit problem, every segment may contain multiple tasks. Thus, we need an algorithm that is able to solve non-stationary bandit problems without knowing the change points. We use `AdSwitch` (Auer et al., 2019b) as the `Base` algorithm in `OS-BASS`. This choice sets $B_{\tau,N_n,\tilde{M}} = \sqrt{\tilde{M}N_n\tau}$, $\forall n \in [N]$, in Lemma 3.1. In the following theorem, we prove a regret bound (proof in Appendix D.3) for `OS-BASS` in the non-stationary setting and with the choice of `AdSwitch` as the `Base`.

Theorem 4.2. *The regret of `OS-BASS` in the sparse non-stationary setting with* `AdSwitch` *as* `Base` *and exploration probability* $\gamma_n = \left(\frac{\tilde{M}K\log K}{N}\right)^{1/3}$ *is*

$$R_T = \tilde{O}\left((\tilde{M}^4KN^2\log K)^{1/3}\tau + \tilde{M}N\sqrt{\tilde{M}\tau}\right). \qquad (7)$$

Looking at the regret bound in (7), we notice that for small N (large τ), the baseline rate $\widetilde{O}(N\sqrt{K\tau})$ (the rate for standard non-stationary bandit algorithms, e.g., `AdSwitch`) cannot be improved. Thus, we are mainly interested in the regime of large number of switches N and small number of optimal arms M. If $N \ge \left(T^3(K\log K)^2/\tilde{M}\right)^{1/5}$ and $M \le (K\log K)^{1/3}$, then the regret in (7) is of $\widetilde{O}(\tilde{M}N\sqrt{\tilde{M}\tau})$, which improves upon the baseline rate $\widetilde{O}(N\sqrt{K\tau})$. However, in the case of small N, the baseline rate can be better than our bound, and thus, the learner does not need to identify the best M arms and should simply play a standard non-stationary bandit algorithm. At the extreme case $N = O(1)$, i.e., only a few changes in the environment, the regret bound $\widetilde{O}(N\sqrt{K\tau})$ cannot be improved. On the other hand, when N is large compared to T, it is easy to establish a $O(N\sqrt{M\tau})$ lower bound, and thus, our $\widetilde{O}(MN\sqrt{M\tau})$ bound is optimal up to a factor of M. Closing this gap remains an open question. We can improve the bound in certain regimes by further tuning the parameters, which we discuss in more details in Appendix D.3.

[9] The superscript ^{o} in `OG°` stands for the “opaque" feedback model in Streeter and Golovin (2007).

5 Experiments

In this section, we study the performance of our algorithms on synthetic environments. The experiments include: 1) `G-BASS`[10] from Section 2, 2) Algorithm 2, `OS-BASS`, 3) `MOSS` which is agnostic MOSS (Audibert and Bubeck, 2009) running independently on the tasks without any knowledge of the optimal M-subset, 4) `Opt-MOSS`, an oracle MOSS that plays only the arms in the optimal M-subset, and its performance constitutes an empirical lower bound on the achievable regret, and 5) `OG°`, which is `OS-BASS` with optimized γ. Error bars are ± 1 standard deviation computed over 5 independent runs.

We study the impact of four variables on the regret: number of tasks N, length of each task τ, number of arms in each task K, and the optimal subset size M. To do so, we fix a default setting of (N, τ, K, M) and for each experiment we let one of these parameters vary. The problems are generated by an *oblivious* adversary (see Appendix F for further details).

Figure 1 demonstrates the impact of M and N. Further experiments in Appendix F illustrate the effect of all four variables including τ and K. Under Assumption 2.1 (left two plots), `G-BASS` outperforms all methods with a regret close to that of the oracle `Opt-MOSS`. It also outperforms `OS-BASS` by using its effective BAI module. Without this assumption, `OS-BASS` outperforms the other algorithms, while `G-BASS` naturally has high variance.

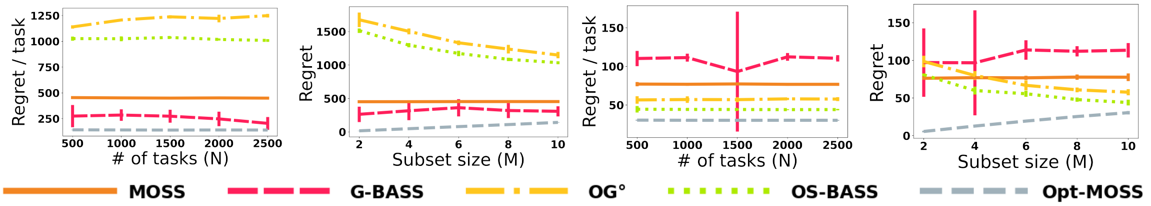

Figure 1: Default setting: $(N, \tau, K, M) = (500, 4500, 30, 10)$. In the right two plots $\tau = 450$. Left to right: Regret as a function of N and M under Assumption 2.1. Regret as a function of N and M without Assumption 2.1.

6 Discussion and future work

We study a problem of N tasks of K-armed bandits arriving sequentially in the sparse meta-learning and non-stationary bandits settings. We design an algorithm based on a reduction to bandit submodular optimization, and prove that its regret with respect to the best M-subset is $\tilde{O}(NM\sqrt{M\tau} + (M^4 K N^2 \log K)^{1/3}\tau)$, where $\tau = T/N$. Under additional identifiability assumptions, we develop a meta-learning algorithm with an improved, essentially optimal regret.

In our most general solution, we assume the number of tasks N is known, and M is given as input to the algorithm. Designing an algorithm with a regret bound that holds for unknown N and simultaneuously for all M is left for further work.

References

Yasin Abbasi-Yadkori, András György, and Nevena Lazić. A new look at dynamic regret for non-stationary stochastic bandits. *Journal of Machine Learning Research*, 24(288):1–37, 2023.

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*, 2009.

P. Auer and R. Ortner. Ucb revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, pages 55—-65, 2010.

[10] `E-BASS` is computationally too expensive so we only run it on smaller settings in Appendix F.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *36th Annual Symposium on Foundations of Computer Science*, 1995.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 2002.

Peter Auer, Yifang Chen, Pratik Gajane, Chung-Wei Lee, Haipeng Luo, Ronald Ortner, and Chen-Yu Wei. Achieving optimal dynamic regret for non-stationary bandits without prior information. In *COLT*, 2019a.

Peter Auer, Pratik Gajane, and Ronald Ortner. Adaptively tracking the best bandit arm with an unknown number of distribution changes. In *Proceedings of the Thirty-Second Conference on Learning Theory*, pages 138–158. PMLR, 25–28 Jun 2019b. URL http://proceedings.mlr.press/v99/auer19a.html.

Mohammad Gheshlaghi Azar, Alessandro Lazaric, and Emma Brunskill. Sequential transfer in multi-armed bandit with finite set of models. In *NIPS*, 2013.

Maria-Florina Balcan, Keegan Harris, Mikhail Khodak, and Zhiwei Steven Wu. Meta-learning adversarial bandits. *arXiv preprint arXiv:2205.14128*, 2022.

Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12: 149–198, 2000.

Donald A. Berry, Robert W. Chen, Alan Zame, David C. Heath, and Larry A. Shepp. Bandit problems with infinitely many arms. *The Annals of Statistics*, 1997.

T. Bonald and A. Proutiere. Two-target algorithms for infinite-armed bandits with Bernoulli rewards. In *In Advances in Neural Information Processing Systems*, 2013.

Olivier Bousquet and Manfred K. Warmuth. Tracking a small set of experts by mixing past posteriors. *Journal of Machine Learning Research*, 2002.

Alexandra Carpentier and Michal Valko. Simple regret for infinitely many armed bandits. In *ICML*, 2015.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Leonardo Cella, Alessandro Lazaric, and Massimiliano Pontil. Meta-learning with stochastic linear bandits. *Arxiv*, 2020.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games.* Cambridge University Press, 2006.

Hock Peng Chan and Shouri Hu. Infinite arms bandit: Optimality via confidence bounds, 2020.

Yifang Chen, Chung-Wei Lee, Haipeng Luo, and Chen-Yu Wei. A new algorithm for non-stationary contextual bandits: Efficient, optimal and parameter-free. In *COLT*, 2019.

Giulia Denevi, Carlo Ciliberto, Dimitris Stamos, and Massimiliano Pontil. Learning to learn around a common mean. In *Advances in Neural Information Processing Systems*, volume 31, 2018a.

Giulia Denevi, Carlo Ciliberto, Dimitris Stamos, and Massimiliano Pontil. Incremental learning-to-learn with statistical guarantees, 2018b.

Giulia Denevi, Carlo Ciliberto, Riccardo Grazzi, and Massimiliano Pontil. Learning-to-learn stochastic gradient descent with biased regularization. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

M. Dimakopoulou, N. Vlassis, and T. Jebara. Marginal posterior sampling for slate bandits. In *IJCAI*, 2019.

Uriel Feige, László Lovász, and Prasad Tetali. Approximating min sum set cover. *Algorithmica*, 40 (4):219–234, 2004.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimilano Pontil. Bilevel programming for hyperparameter optimization and meta-learning, 2018.

Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Joey Hong, Branislav Kveton, Manzil Zaheer, Yinlam Chow, Amr Ahmed, and Craig Boutilier. Latent bandits revisited. In *NeurIPS*, 2020a.

Joey Hong, Branislav Kveton, Manzil Zaheer, Yinlam Chow, Amr Ahmed, Mohammad Ghavamzadeh, and Craig Boutilier. Non-stationary latent bandits. *arXiv*, 2020b.

S. Kale, L. Reyzin, and R. E. Schapire. Non-stochastic bandit slate problems. In *NIPS*, 2010.

Anand Kalvit and Assaf Zeevi. From finite to countable-armed bandits. In *Conference on Neural Information Processing Systems*, 2020.

Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Provable guarantees for gradient-based meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 424–433, 2019.

Weihao Kong, Raghav Somani, Zhao Song, Sham Kakade, and Sewoong Oh. Meta-learning for mixed linear regression. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Branislav Kveton, Martin Mladenov, Chih-Wei Hsu, Manzil Zaheer, Csaba Szepesvári, and Craig Boutilier. Differentiable meta-learning in contextual bandits. *arXiv:2006.05094v1*, 2020.

Branislav Kveton, Mikhail Konobeev, Manzil Zaheer, Chih wei Hsu, Martin Mladenov, Craig Boutilier, and Csaba Szepesvari. Meta-thompson sampling. *Arxiv*, 2021.

T. Lattimore and C. Szepesvári. *Bandit Algorithms.* Cambridge University Press, 2020. ISBN 9781108687492. URL https://books.google.com/books?id=xe3vDwAAQBAJ.

N. Littlestone and M.K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994. ISSN 0890-5401. doi: https://doi.org/10.1006/inco.1994.1009. URL https://www.sciencedirect.com/science/article/pii/S0890540184710091.

Odalric-Ambrym Maillard and Shie Mannor. Latent bandits. In *ICML*, 2014.

Nadav Merlis and Shie Mannor. Batch-size independent regret bounds for the combinatorial multi-armed bandit problem. In *COLT*, 2019.

Hyejin Park, Seiyun Shin, Kwang-Sung Jun, and Jungseul Ok. Transfer learning in bandits with latent continuity. *arXiv*, 2021.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *ICML*, 2008.

Jason Rhuggenaath, Alp Akcay, Yingqian Zhang, and Uzay Kayma. Algorithms for slate bandits with non-separable reward functions. *arXiv*, 2020.

Yoan Russac, Claire Vernade, and Olivier Cappé. Weighted linear bandits for non-stationary environments. In *NeurIPS*, 2019.

Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. Technical report, Carnegie Mellon University, Pittsburgh, PA, School of computer science, 2007. URL https://apps.dtic.mil/sti/pdfs/ADA476871.pdf. Available at https://apps.dtic.mil/sti/pdfs/ADA476871.pdf.

Joe Suk and Samory Kpotufe. Tracking most significant arm switches in bandits. In *35th Annual Conference on Learning Theory*, 2022.

Sebastian Thrun. Is learning the n-th thing any easier than learning the first? In *Advances in neural information processing systems*, pages 640–646. MORGAN KAUFMANN PUBLISHERS, 1996.

Nilesh Tripuraneni, Chi Jin, and Michael I. Jordan. Provable meta-learning of linear representations. *Arxiv*, 2021.

Yizao Wang, Jean-Yves Audibert, and Rémi Munos. Algorithms for infinitely many-armed bandits. In *NIPS*, 2008.

Chen-Yu Wei and Haipeng Luo. Non-stationary reinforcement learning without prior knowledge: An optimal black-box approach. *COLT*, 2021.

Kai Zheng, Haipeng Luo, Ilias Diakonikolas, and Liwei Wang. Equipping experts/bandits with long-term memory. In *NeurIPS*, 2019.

A Submodular functions

In this section, we verify that $f(S) \doteq f(r, S) = \max_{a \in S} r(a)$ is a submodular function. It is obviously monotone. We have

$$f(S_1 \cup S_2 \cup \{a\}) - f(S_1 \cup S_2) \leq f(S_1 \cup \{a\}) - f(S_1)$$

This is because (1) if $r(a) \leq f(S_1)$ then the inequality holds as $f(S_1 \cup \{a\}) = f(S_1)$ and $f(S_1 \cup S_2 \cup \{a\}) = f(S_1 \cup S_2)$. (2) if $r(a) > f(S_1)$ then (2.i) if $r(a) > f(S_1 \cup S_2)$ the inequality holds as $r(a) - f(S_1 \cup S_2) \leq r(a) - f(S_1) \iff f(S_1 \cup S_2) \geq f(S_1)$, which holds by monotonicity of f, (2.ii) if $r(a) \leq f(S_1 \cup S_2)$ then $f(S_1 \cup S_2 \cup \{a\}) = f(S_1 \cup S_2)$ and the inequality simplifies to $0 \leq r(a) - f(S_1)$ which holds as we assumed $r(a) > f(S_1)$ in (2).

B Proofs for Section 2

B.1 Proof of Theorem 2.3

The proof relies on solving the min-max problem in (4). First, we consider the case that the best-arm-identification can be performed with probability 1 (i.e., $\delta = 0$ in the efficient identification assumption). From symmetry, it is easy to see that $V_n(\mathcal{I}_n)$ only depends on the size of $\mathcal{I}_n$, and not the actual arms in $\mathcal{I}_n$. Therefore, to simplify notation and emphasize the dependence on the number of discovered optimal arms, we use below $V_n(|\mathcal{I}_n|) := V_n(\mathcal{I}_n)$. Let $n_M = \text{argmin}_n\{|\mathcal{I}_n| = M\}$ be the first round when all optimal arms have been discovered. Then from any $n > n_M$, the adversary no longer can reveal new arms ($q = 0$), and the learner should no longer explore ($p = 0$), and so

$$\forall n \geq n_M,\ V_n(M) = (N - n)\mathbf{C}_{\texttt{hit}}.$$

Denoting $s = |\mathcal{I}_n|$, the min-max optimization objective in (4) can be written as

$$\begin{aligned} L(q,p) = \mathbf{C}_{\texttt{hit}} + p(\mathbf{C}_{\texttt{info}} - \mathbf{C}_{\texttt{hit}}) + V_{n+1}(s) + \\ q(1-p)(\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}}) - p\left[q^1(V_{n+1}(s)\right. \\ -V_{n+1}(s+1)) + \dots \\ \left. +q^{M-s}(V_{n+1}(s) - V_{n+1}(M))\right]\ , \end{aligned}$$

where q^i denotes the probability that the environment reveals i optimal arms in the round, and $q = \sum_{i=1}^{M-s} q^i$. Given that $V_{n+1}(s) - V_{n+1}(s+1) < \dots < V_{n+1}(s) - V_{n+1}(M)$, the maximizing q is such that $q^i = 0$ for $i > 1$ and $q = q^1$. Using this, the saddle point can be obtained by solving $\partial L(q,p)/\partial q = 0$ and $\partial L(q,p)/\partial p = 0$:

$$\begin{aligned} p = p_n &= \frac{\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}}}{\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}} + V_{n+1}(s) - V_{n+1}(s+1)} \\ q_n &= \frac{\mathbf{C}_{\texttt{info}} - \mathbf{C}_{\texttt{hit}}}{\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}} + V_{n+1}(s) - V_{n+1}(s+1)}\ . \end{aligned} \tag{8}$$

Plugging these values in (4), we get

$$V_n(s) = V_{n+1}(s) + \mathbf{C}_{\texttt{hit}} + \frac{(\mathbf{C}_{\texttt{info}} - \mathbf{C}_{\texttt{hit}})(\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}})}{\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}} + V_{n+1}(s) - V_{n+1}(s+1)}\ .$$

Given N and M, the policy of the learner and the adversary can be computed by solving the above recursive equation. Given that for any $s < M$, $V_n(s+1) \geq V_{n+1}(s+1) + \mathbf{C}_{\texttt{hit}}$,

$$V_n(s) - V_n(s+1) \leq V_{n+1}(s) - V_{n+1}(s+1) + \frac{(\mathbf{C}_{\texttt{info}} - \mathbf{C}_{\texttt{hit}})(\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}})}{\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}} + V_{n+1}(s) - V_{n+1}(s+1)}\ .$$

Let $G_n(s) = V_n(s) - V_n(s+1) \geq 0$ be the cost difference in state s relative to state $s+1$. We have

$$G_n(s) \leq G_{n+1}(s) + \frac{(\mathbf{C}_{\texttt{info}} - \mathbf{C}_{\texttt{hit}})(\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}})}{\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}} + G_{n+1}(s)} \ , \tag{9}$$

and indeed by a telescopic argument,

$$R_T - NB_{\tau,M} = V_0(0) - V_0(M) = \sum_{s=0}^{M-1} (V_0(s) - V_0(s+1)) = \sum_{s=0}^{M-1} G_0(s) \ .$$

The proof is completed by bounding $G_0(s)$ by backward induction on $n \leq N$:

$$G_{N-n}(s) \leq \sqrt{2(\mathbf{C}_{\texttt{info}} - \mathbf{C}_{\texttt{hit}})(\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}})n} \ .$$

The proof of this inequality relies on standard algebraic manipulations that can be found in Appendix B.2. When the BAI routine returns the best arm with probability at least $1 - \delta/N$, with a simple union bound argument, the probability that $\mathcal{I}_n$ ever contains wrong elements is bounded by δ and the above derivations again hold.

B.2 Complement to the proof of Theorem 2.3

We are left to prove that

$$G_n(s) \leq G_{n+1}(s) + \frac{(\mathbf{C}_{\texttt{info}} - \mathbf{C}_{\texttt{hit}})(\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}})}{\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}} + G_{n+1}(s)}, \tag{10}$$

given in (9) implies that for any $n \leq N$,

$$G_n(s) \leq \sqrt{2(\mathbf{C}_{\texttt{info}} - \mathbf{C}_{\texttt{hit}})(\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}})(N-n)} \ . \tag{11}$$

We proceed by (backward) induction. First, by definition, $G_N(s) = V_N(s) - V_N(s+1) = 0$ for all s, thus (11) holds for $n = N$. Next, assume that (11) holds for $\{N, N-1, \ldots, n+1\}$, and we show that it also holds for n.

Consider positive constants $b \geq a$ and consider the function $h(z) = z + \frac{ab}{b+z}$ defined on $[0, c]$ for some $c > 0$. Then $h'(z) = 1 - ab/(b+z)^2 \geq 0$. Therefore, h is maximized at $z = c$. Since the right-hand side of (10) is of the form $h(G_{n+1}(s))$ with $a = \mathbf{C}_{\texttt{info}} - \mathbf{C}_{\texttt{hit}}$ and $b = \mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}}$, which indeed satisfy $b \geq a$. By this argument, the induction assumption, and $0 \leq G_{n+1}(s) \leq \sqrt{ab(N-n-1)}$ by the induction hypothesis, we obtain that

$$\begin{aligned} G_n(s) &\leq \sqrt{2(\mathbf{C}_{\texttt{info}} - \mathbf{C}_{\texttt{hit}})(\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}})(N-n-1)} \\ &\qquad + \frac{(\mathbf{C}_{\texttt{info}} - \mathbf{C}_{\texttt{hit}})(\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}})}{\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}} + \sqrt{2(\mathbf{C}_{\texttt{info}} - \mathbf{C}_{\texttt{hit}})(\mathbf{C}_{\texttt{miss}} - \mathbf{C}_{\texttt{hit}})(N-n-1)}} \\ &= \sqrt{2ab(N-n-1)} + \frac{ab}{b + \sqrt{2ab(N-n-1)}} \end{aligned} \tag{12}$$

It remains to show that the right-hand side above is bounded from above by $\sqrt{2ab(N-n)}$. This follows since

$$\begin{aligned} \sqrt{2ab(N-n)} - \sqrt{2ab(N-n-1)} &= \frac{\sqrt{2ab}}{\sqrt{N-n} + \sqrt{N-n-1}} \\ &= \frac{ab}{\sqrt{ab(N-n)/2} + \sqrt{ab(N-n-1)/2}} \geq \frac{ab}{b + \sqrt{2ab(N-n-1)}} \end{aligned}$$

where the last inequality holds because

$$\begin{aligned} b + \sqrt{2ab(N-n-1)} &\geq \left[\sqrt{ab} + \sqrt{ab(N-n-1)/2}\right] + \sqrt{ab(N-n-1)/2} \\ &\geq \sqrt{ab(N-n)/2} + \sqrt{ab(N-n-1)/2} \end{aligned}$$

(where we used that $1 + \sqrt{z} \geq \sqrt{z+1}$ for $z \geq 0$). Thus, $G_n(s) \leq \sqrt{2ab(N-n)}$, proving the induction hypothesis (11) for n.

B.3 Proof of Theorem 2.4

The proof relies on the analysis of the optimization problem defined as in Eq. (4) with $p_n = \hat{p}_{n,|S_n|} = \sqrt{|S_n|K\tau/(nB_{\tau,K})}$ (no minimization over the exploration probability of the learner in task n). As in the proof of Theorem 2.3, we assume that the best arm identification is successful, and the extension to $\delta \neq 0$ can be done the same way. After some algebraic manipulation, similarly to the proof of Theorem 2.3, the optimization objective can be written as

$$\begin{aligned} L(q) = V_{n+1}(S_n) + B_{\tau,|S_n|} + \hat{p}_{n,|S_n|}(B_{\tau,K} - B_{\tau,|S_n|}) \\ + q\left\{(1-\hat{p}_{n,|S_n|})(\tau - B_{\tau,|S_n|}) - \hat{p}_{n,|S_n|}(V_{n+1}(S_n) - V_{n+1}(S'_n))\right\} \end{aligned}$$

where S'_n is the new greedy subset selected by the learner in time step $n+1$ if $S^*_n \cap S_n = \emptyset$ and the learner chooses to explore at time n (we use the notation S'_n instead of S_{n+1} to emphasize that this corresponds to the aforementioned choices of the learner and the adversary). Given that L is linear in q, the optimal adversary choice is either $q = 0$ or $q = 1$ (similarly as in Theorem 2.3, it is suboptimal for the adversary to reveal multiple optimal arms). We have

$$q = \begin{cases} 0 & \text{if } (1-\hat{p}_{n,|S_n|})(\tau - B_{\tau,|S_n|}) - \hat{p}_{n,|S_n|}(V_{n+1}(S_n) - V_{n+1}(S'_n)) \leq 0\,, \\ 1 & \text{otherwise} \end{cases}$$

When $q = 0$, $V_n(S_n) = V_{n+1}(S_n) + B_{\tau,|S_n|} + \hat{p}_{n,|S_n|}(B_{\tau,K} - B_{\tau,|S_n|})$, and given that $|S_n| \leq M'$, the total contribution of these rounds to the regret is bounded by

$$NB_{\tau,M'} + \sqrt{\frac{\tau M'KN}{B_{\tau,K}}}B_{\tau,K}\ .$$

Next, consider the rounds where $q = 1$. Among these rounds, consider rounds where the adversary chooses a particular arm $a \in S^*$ and the learner chooses to explore (Exr). This arm is not added to the future Ext subset of the learner if instead another arm is used to cover this round. This means that after at most K such rounds, the learner adds a to the Ext subset. Since the learner's regret in the exploration rounds is $B_{\tau,K}$, in these rounds the cumulative regret is bounded by $MKB_{\tau,K}$. Since the random choices made by the learner and the adversary are independent in the same round, we discover the first arm in at most $K/\hat{p}_{N,1}$ tasks in expectation, the second in at most $K/\hat{p}_{N,2}$ tasks in expectation, and so on. Thus, since the size of our cover is at most M', we get

$$K\sum_{s=1}^{M'}\frac{1}{\hat{p}_{N,s}} = \sqrt{\frac{KNB_{\tau,K}}{\tau}}\sum_{s=1}^{M'}\frac{1}{\sqrt{s}} \leq 2\sqrt{\frac{M'KNB_{\tau,K}}{\tau}}\ .$$

The adversary reveals all positions after $2\sqrt{\frac{M'KNB_{\tau,K}}{\tau}}$ such tasks in expectation where the adversary's choice is $q = 1$. If in these tasks the learner chooses to exploit, it can suffer a regret τ, leading to a total expected regret of at most $2\sqrt{M'KNB_{\tau,K}\tau}$. Thus, the total regret of rounds with $q = 1$ is bounded by

$$2\sqrt{M'KNB_{\tau,K}\tau} + MKB_{\tau,K}\ .$$

Therefore,

$$R_T = V_0(\emptyset) \leq NB_{\tau,M'} + MKB_{\tau,K} + 3\sqrt{M'K\tau B_{\tau,K}N}\ .$$

C Proof of Lemmas 3.1 and 4.1

We have that

$$\begin{aligned}
R_T &= \sup_{(r_n)_{n=1}^N,\, (a_n)_{n=1}^N \,:\, |\{a_n\}_{n=1}^N| \le M} \mathbb{E}\left[\sum_{n=1}^N \sum_{t=1}^{\tau_n} (r_n(a_n) - r_n(A_{n,t}))\right] \\
&= \sup_{\substack{r_1,\dots,r_N \\ \in[0,1]^K}} \max_{S\in\mathcal{S}} \mathbb{E}\left[\sum_{n=1}^N \sum_{u=1}^{N_n} \sum_{t=1}^{\tau_{n,u}} (\max_{a\in S}(r_{n,u}(a) - r_{n,u}(A_{n,u,t}))\right] && (\mathcal{S} \text{ is defined in Section 3.1}) \\
&= \sup_{\substack{r_1,\dots,r_N \\ \in[0,1]^K}} \max_{S\in\mathcal{S}} \mathbb{E}\left[\sum_{n=1}^N \sum_{u=1}^{N_n} \left(\sum_{t=1}^{\tau_{n,u}} (\max_{a\in S} r_{n,u}(a) - \max_{a\in S_n} r_{n,u}(a)) + \sum_{t=1}^{\tau_{n,u}} (\max_{a\in S_n} r_{n,u}(a) - r_{n,u}(A_{n,u,t}))\right)\right] \\
&\le \sup_{(f_{n,u}\in\mathcal{F})_{n\in[N],u\in[N_n]}} \max_{S\in\mathcal{S}} \mathbb{E}\left[\sum_{n=1}^N \left(\sum_{u=1}^{N_n} \tau_{n,u}(f_{n,u}(S) - f_{n,u}(S_n)) + \tau\varepsilon_n\right)\right] \\
&= \sup_{f_1,\cdots,f_N\in\mathcal{F}} \max_{S\in\mathcal{S}} \mathbb{E}\left[\sum_{n=1}^N (f_n(S) - f_n(S_n) + B_{\tau,N_n,M})\right] .
\end{aligned}$$

D Proofs for Section 3.2

First, we present the relevant results from Streeter and Golovin (2007) with appropriate modifications. We start with the regret analysis of the OG algorithm, which is designed to solve the online submodular maximization in the full feedback model.

D.1 The OG algorithm

For a submodular function g, consider an ordered set of actions $\bar{S} = \langle \bar{a}_1, \bar{a}_2, \cdots \rangle$ that satisfies the following *greedy* condition for any j:

$$g(\bar{S}_i \cup \bar{a}_i) - g(\bar{S}_i) \ge \max_{a\in[K]} \{g(\bar{S}_i \cup a) - g(\bar{S}_i)\} - \alpha_i, \tag{13}$$

where α_i are some positive error terms. Let $\bar{S}_0 = \emptyset$, $\bar{S}^i = \langle \bar{a}_1, \bar{a}_2, \cdots, \bar{a}_{i-1} \rangle$, and for a sequence of actions $S \subset [K]$, let $S_{\langle M \rangle}$ be the set of actions in S truncated at the M'th action. The following result shows near-optimality of $\bar{S}$ as constructed above. Recall that $\tilde{M} = \lceil M \log N \rceil$.

Theorem D.1 (Based on Streeter and Golovin (2007), Theorem 6)**.** *Consider the greedy solution in Equation (13). Then*

$$g(\bar{S}_{\langle \tilde{M} \rangle}) > \left(1 - \frac{1}{N}\right) \max_{S\in\mathcal{S}} g(S_{\langle M \rangle}) - \sum_{j=1}^{\tilde{M}} \alpha_i .$$

Proof. Let $C^* = \max_{S\in\mathcal{S}} g(S_{\langle M \rangle})$ and for any j let $\Delta_j = C^* - g(\bar{S}_j)$. Then, by Fact D.2 below, we have $C^* \le g(\bar{S}_j) + M(s_j + \alpha_j)$. Therefore, $\Delta_j \le M(s_j + \alpha_j) = M(\Delta_j - \Delta_{j+1} + \alpha_j)$ which means $\Delta_{j+1} \le \Delta_j(1 - \frac{1}{M}) + \alpha_j$. Unrolling this $\tilde{M}$ times (and noting $1 - \frac{1}{N} < 1$) gives

$$\begin{aligned}
\Delta_{\tilde{M}+1} &\le \Delta_1 \left(1 - \frac{1}{M}\right)^{\tilde{M}} + \sum_{j=1}^{\tilde{M}} \alpha_j \\
&< \Delta_1 \tfrac{1}{N} + \sum_{j=1}^{\tilde{M}} \alpha_j \le C^* \tfrac{1}{N} + \sum_{j=1}^{\tilde{M}} \alpha_j .
\end{aligned}$$

This concludes the proof since $C^* - g(\bar{S}_{\tilde{M}+1}) = \Delta_{\tilde{M}+1}$ and $g(\bar{S}_{\tilde{M}+1}) = g(\bar{S}_{\langle \tilde{M} \rangle})$. □

Algorithm 3 `OG` algorithm

```
Input: Subset size $M$, Expert algorithms $\mathcal{E}_1, \cdots, \mathcal{E}_{\tilde{M}}$;
for $n \in [N]$ do
    Let $S_{n,0} = \emptyset$
    for $j \in \{1, \cdots, \tilde{M}\}$ do
        Let action $a_j^n \in [K]$ be the choice of expert $\mathcal{E}_j$
        Let $S_{n,j} \leftarrow S_{n,j-1} \cup \{a_j^n\}$
    end for
    Play subset $S_n$ and observe function $g_n$.
    for $j \in \{1, \cdots, \tilde{M}\}$ and any $a \in [K]$ do
        Let $x_{j,a}^n \leftarrow g_n(S_{n,j-1} \cup \{a\}) - g_n(S_{n,j-1})$
        Expert $\mathcal{E}_j$ receives payoff vector $(x_{j,a}^n)_{a \in [K]}$
    end for
end for
```

Fact D.2 (Streeter and Golovin (2007), Fact 1)**.** *For any subset of arms S, and any positive integer j, and any $t > 0$, we have $g(S_{\langle t \rangle}) \leq g(\bar{S}_j) + t(s_j + \alpha_j)$ where $s_j = g(\bar{S}_{j+1}) - g(\bar{S}_j)$.*

Proof. The proof is akin to Fact 1 of Streeter and Golovin (2007) and it goes $g(S_{\langle t \rangle}) \leq g(\bar{S}_j \cup S_{\langle t \rangle}) \leq g(\bar{S}_j) + t(s_j + \alpha_j)$. The first inequality holds because g is a monotone function. The second inequality is by definition of s_j and Condition 1, (Streeter and Golovin, 2007, Lemma 1) – for any submodular function g, and any $S_1, S \in \mathcal{S}$, $\frac{g(S_1 \cup S) - g(S_1)}{|S|} \leq \max_{a \in \mathcal{A}} g(S_1 \cup \{a\}) - g(S_1)$ and we replace $S_1 \leftarrow \bar{S}_j$, $S \leftarrow S_{\langle t \rangle}$, so $|S| = t$. □

Consider a sequence of submodular functions $g_1, \cdots, g_N$ for a fixed $N \in \mathbb{N}$. Define the coverage regret of a submodular maximization policy by

$$R_{\text{coverage}}(N) := \left(1 - \frac{1}{N}\right) \max_{S \in \mathcal{S}} \sum_{n=1}^{N} g_n(S_{\langle M \rangle}) - \sum_{n=1}^{N} g_n(S_n) .$$

Algorithm 3 is the `OG` algorithm of Streeter and Golovin (2007) for the full feedback model modified for our setting with $\tilde{M}$ experts. In this algorithm, N is the number of rounds (analogous to segments/tasks). The algorithm uses a set of experts and each expert is a *randomized weighted majority* (RWM) algorithm (Littlestone and Warmuth, 1994). See Chapter 4.2 of Cesa-Bianchi and Lugosi (2006) for more information. The following lemma connects the coverage regret of the `OG` algorithm and the regret of the experts.

Lemma D.3 (Lemma 3 of Streeter and Golovin (2007))**.** *Let $G_j(N)$ be the cumulative regret of expert $\mathcal{E}_j$ in `OG` algorithm, and let $G(N) = \sum_{j=1}^{\tilde{M}} G_j(N)$. Then, $R_{coverage}(N) \leq G(N)$.*

Proof. As we will show, the `OG` algorithm is an approximate version of the offline greedy subset selection, defined by Equation (13), for function $g = \frac{1}{N} \sum_{n=1}^{N} g_n$. First, let's view the sequence of actions selected by $\mathcal{E}_j$ as a single "batch-action" $\tilde{a}_j$, and extend the domain of each g_n to include the batch-actions by defining $g_n(S \cup \{\tilde{a}_j\}) = g_n(S \cup \{a_j^n\})$ for all $S \in \mathcal{S}$. Thus, the online algorithm produces a single set $\tilde{S} = \{\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_{\tilde{M}}\}$. By definition we have

$$\frac{G_j(N)}{N} = \max_{a \in [K]} \left(g(\tilde{S}_{\langle j-1 \rangle} \cup \{a\}) - g(\tilde{S}_{\langle j-1 \rangle})\right) - \left(g(\tilde{S}_{\langle j-1 \rangle} \cup \{\tilde{a}_j\}) - g(\tilde{S}_{\langle j-1 \rangle})\right) ,$$

where $\tilde{S}_{\langle j \rangle}$ is $\tilde{S}$ truncated at j'th action. Thus the `OG` algorithm simulates the greedy schedule (13) for function g, where the j'th decision is made with additive error $\alpha_j = \frac{G_j(N)}{N}$. By Theorem D.1 and the fact that function g is submodular, we get that $R_{\text{coverage}}(N) \leq \sum_{j=1}^{\tilde{M}} G_j(N) = G(N)$. □

Algorithm 4 `OG°` algorithm

```
Input: Subset size $M$, Expert algorithms $\mathcal{E}_1, \cdots, \mathcal{E}_{\tilde{M}}$, Probabilities of exploration $\{\gamma_n\}_{n=1}^N$;
for $n \in [N]$ do
    Observe $g_n$
    For $i \in [\tilde{M}]$, let $a_i$ be the choice of $\mathcal{E}_i$
    Set $S_n = \{a_1, \cdots, a_{\tilde{M}}\}$
    With prob. $\gamma_n$, $E_n = $ EXR, otherwise $E_n = $ EXT
    if $E_n = $ EXT then
        All experts receive the zero vector as the payoff vector
    else
        Choose $i \in [\tilde{M}]$ uniformly at random
        Choose a new action $a'_i$ uniformly at random
        Replace $i$'th element of $S_n$ with $a'_i$: $S_{n:i} \leftarrow \{a_1, \cdots, a_{i-1}, a'_i\}$
        Expert $\mathcal{E}_i$ receives a payoff vector that is zero everywhere except at position $a'_i$ that has the
          value of $g_n(S_{n:i})$
        All other experts receive the zero vector as the payoff vector
    end if
end for
```

By Lemma 4 of Streeter and Golovin (2007), $\mathbb{E}[G(N)] = O(\sqrt{\tilde{M}N\log(K)})$.

D.2 The `OG°` algorithm

Algorithm 4 is based on the `OG°` algorithm of Streeter and Golovin (2007) for the bandit (opaque) feedback model. This algorithm is very similar to `OS-BASS` algorithm so we omit the description. The difference is that in the `OS-BASS` algorithm, the meta-learner observes the value of the submodular function up to a noise term $\varepsilon_n = (1/\tau)B_{\tau,N_n,\tilde{M}}$. So we extend the analysis of Streeter and Golovin (2007) to the case that the observation of the submodular function is corrupted by a noise term.

Lemma D.4. *Consider an expert prediction problem with K actions, and let x_a^n be the payoff for action $a \in [K]$ in round n. Let $\mathcal{E}$ be a an expert algorithm that gets payoff vector $(x_a^n)_{a\in[K]}$ in round n, let e_n be its action in round n, and let $R(N)$ be its worst-case expected regret over N rounds: $R(N) = \max_a \sum_{n=1}^N (x_a^n - x_{e_n}^n)$. Let $\mathcal{E}'$ and $\tilde{\mathcal{E}}$ be the same algorithm but with payoff vectors $(\hat{x}_a^n)_{a\in[K]}$ and $(\tilde{x}_a^n)_{a\in[K]}$ instead of $(x_a^n)_{a\in[K]}$. These feedbacks are such that $\mathbb{E}[\hat{x}_a^n] = \gamma_n x_a^n + \delta_n$ for some constant $\gamma_n \in [0,1]$ and δ_n, and*

$$\mathbb{E}[\tilde{x}_a^n] \in [\mathbb{E}[\hat{x}_a^n - \gamma_n \varepsilon'_n], \mathbb{E}[\hat{x}_a^n]]$$

for some $\varepsilon'_n \geq 0$. Let u_n be the action of algorithm $\tilde{\mathcal{E}}$ in round n. Then the worst-case expected regret of $\tilde{\mathcal{E}}$ is bounded as

$$\max_a \sum_{n=1}^N (x_a^n - x_{u_n}^n) \leq \frac{1}{\min_n \gamma_n} R(N) + \sum_{n=1}^N \mathbb{E}[\varepsilon'_n] \,.$$

Proof. By the regret guarantee of the expert algorithm,

$$R(N) \geq \max_{a\in[K]} \sum_{n=1}^N (\tilde{x}_a^n - \tilde{x}_{u_n}^n) \,.$$

Thus, for any a,

$$\begin{aligned}
R(N) &\geq \mathbb{E}\left[\sum_{n=1}^{N} \tilde{x}_a^n - \tilde{x}_{u_n}^n | \tilde{x}\right] \\
&\geq \sum_{n=1}^{N} \mathbb{E}[\hat{x}_a^n - \gamma_n \varepsilon'_n - \hat{x}_{u_n}^n] \\
&= \sum_{n=1}^{N} \mathbb{E}[\gamma_n x_a^n + \delta_n - \gamma_n \varepsilon'_n - \gamma_n x_{u_n}^n - \delta_n] \\
&= \sum_{n=1}^{N} \gamma_n \mathbb{E}[x_a^n - \varepsilon'_n - x_{u_n}^n] \\
&\geq \min_n \gamma_n \sum_{n=1}^{N} \mathbb{E}[x_a^n - \varepsilon'_n - x_{u_n}^n] \,.
\end{aligned}$$

Therefore,

$$\frac{1}{\min_n \gamma_n} R(N) + \sum_{n=1}^{N} \mathbb{E}[\varepsilon'_n] \geq \sum_{n=1}^{N} \mathbb{E}[x_a^n - x_{u_n}^n] \,,$$

and the result follows as the above inequality holds for all a. □

The next lemma bounds the coverage regret of the $\mathtt{OG}^\circ$ algorithm.

Lemma D.5 (Coverage Regret). *Let $\gamma_n = \left(\frac{\tilde{M} K \log K}{N}\right)^{1/3}$ for all n. Assume the jth expert $\tilde{\mathcal{E}}_j$ in the $\mathtt{OG}^\circ$ algorithm gets a payoff vector $(\tilde{x}_a^n)_{a\in[K]}$ in round n such that the following holds:*

$$\gamma' \left(g_n(S_{n:j-1} \cup \{a\}) - \varepsilon'_n\right) \leq \mathbb{E}[\tilde{x}_a^n] \leq \gamma g_n(S_{n:j-1} \cup \{a\}) \,.$$

where $\gamma = \left(\frac{\tilde{M} K \log K}{N}\right)^{1/3}$ and $\gamma' = \frac{\gamma}{\tilde{M} K}$. Then for the sequence of subsets $(S_n)_{n=1}^N$ chosen by the $\mathtt{OG}^\circ$ algorithm,

$$R_{coverage}(N) \leq (\tilde{M}^4 N^2 K \log k)^{1/3} + \tilde{M} \sum_{n=1}^{N} \mathbb{E}[\varepsilon'_n] \,.$$

Proof. We start with another expert $\mathcal{E}'$ that gets payoff vector $\hat{x}^n$ such that $\mathbb{E}[\hat{x}_a^n] = \gamma' g_n(S_{n-1:j} \cup \{a\})$ for any action a. Then we can write

$$\mathbb{E}[\hat{x}_a^n] = \gamma' x_a^n + \delta_n$$

for $x_a^n = (g_n(S_{n-1:j} \cup \{a\}) - g_n(S_{n-1:j}))$ and $\delta_n = \gamma' g_n(S_{n-1:j})$, where $\gamma' = \frac{\gamma}{\tilde{M} K}$. Let N_{Exr} be the number of exploration rounds. Let $G'_j(N)$ be the total regret of expert $\mathcal{E}'_j$. By Lemma D.4 and the regret guarantee of the expert algorithm, the total regret of expert $\mathcal{E}'_j$ is bounded as

$$\begin{aligned}
\mathbb{E}[G'_j(N)] &\leq \frac{1}{\gamma'} \mathbb{E}\sqrt{\left(\max_a \sum_{n=1}^{N} \hat{x}_a^n\right) \log K} \\
&\leq \frac{1}{\gamma'} \mathbb{E}\sqrt{N_{\text{Exr}} \log K} \\
&\leq \sqrt{\frac{N}{\gamma'} \log K} \,,
\end{aligned}$$

where we used Jensen's inequality and $\mathbb{E}[N_{\text{Exr}}] = \gamma' N$ in the last step. Let $\tilde{G}_j(N)$ be regret of expert $\tilde{\mathcal{E}}_j$. We observe that $\mathbb{E}[\tilde{x}_a^n] \in [\mathbb{E}[\hat{x}_a^n - \gamma' \varepsilon'_n], \mathbb{E}[\hat{x}_a^n]]$. Given that the $\mathtt{OG}^\circ$ algorithm takes random actions in the exploration rounds, it incurs an extra $\gamma' N$ regret, and therefore together with Lemma D.4, we have $\mathbb{E}[\tilde{G}_j(N)] \leq \mathbb{E}[G'_j(N)] + \sum_{n=1}^{N} \mathbb{E}[\varepsilon'_n] + \gamma N$. By summing over $j \in [\tilde{M}]$,

$$\mathbb{E}\left[\sum_{j=1}^{\tilde{M}} \tilde{G}_j(N)\right] \leq \tilde{M}\sqrt{\frac{N}{\gamma'} \log K} + \tilde{M}\sum_{n=1}^{N} \mathbb{E}[\varepsilon'_n] + \gamma \tilde{M} N \; .$$

By Lemma D.3 we get

$$\begin{aligned}\mathbb{E}[R_{\text{coverage}}(N)] &\leq \tilde{M}\sqrt{\frac{N}{\gamma'} \log K} + \tilde{M}\sum_{n=1}^{N} \mathbb{E}[\varepsilon'_n] + \gamma \tilde{M} N \\ &= \tilde{M}\sqrt{\frac{N}{\gamma} \tilde{M} K \log K} + \tilde{M}\sum_{n=1}^{N} \mathbb{E}[\varepsilon'_n] + \gamma \tilde{M} N \; .\end{aligned}$$

Finally, choosing $\gamma = \left(\frac{\tilde{M} K \log K}{N}\right)^{1/3}$[11] yields

$$\tilde{M}\sqrt{\frac{N}{\gamma} \tilde{M} K \log K} + \gamma \tilde{M} N = (\tilde{M}^4 N^2 K \log k)^{1/3} \; .$$

Therefore

$$\mathbb{E}[R_{\text{coverage}}(N)] \leq (\tilde{M}^4 N^2 K \log k)^{1/3} + \tilde{M}\sum_{n=1}^{N} \mathbb{E}[\varepsilon'_n] \; .$$

□

D.3 The `OS-BASS` algorithm for non-stationary bandits

Now we are ready to bound the regret of the `OS-BASS` algorithm (shown in Algorithm 2).

Theorem 4.2. *The regret of `OS-BASS` in the sparse non-stationary setting with* `AdSwitch` *as* `Base` *and exploration probability* $\gamma_n = \left(\frac{\tilde{M} K \log K}{N}\right)^{1/3}$ *is*

$$R_T = \tilde{O}\left((\tilde{M}^4 K N^2 \log K)^{1/3} \tau + \tilde{M} N \sqrt{\tilde{M} \tau}\right) . \tag{7}$$

Proof. Fix a sequence of N tasks with unknown and potentially variable task lengths $\{\tau_n\}_{n \in [N]}$. Let π_{OS} be the policy used by Algorithm 2. By the decomposition in Lemma 3.1 the regret of

[11] To be more precise, $\gamma = (3/2)\left(\frac{\tilde{M} K \log K}{N}\right)^{1/3}$.

Algorithm 2 when updated every τ steps (with $N = T/\tau$ updates) satisfies the following,

$$\begin{aligned}
R(\pi_{\text{OS}}, T, N, \tilde{M}) &= \sup_{f_n \in \mathcal{F}} \max_{S \in \mathcal{S}} \mathbb{E} \sum_{n=1}^{N} \big(f_n(S) - f_n(S_n)\big) + B_{\tau, N_n, \tilde{M}} \\
&\le \sup_{f_n \in \mathcal{F}} \max_{S \in \mathcal{S}} \mathbb{E} \Bigg[\sum_{n=1}^{N} \tfrac{1}{N} f_n(S) \\
&\qquad\qquad + \sum_{n=1}^{N} (1 - \tfrac{1}{N}) f_n(S) - \sum_{n=1}^{N} f_n(S_n) + B_{\tau, N_n, \tilde{M}} \Bigg] \\
&\le \frac{N\tau}{N} + \tau \mathbb{E}[R_{\text{coverage}}] + \sum_{n=1}^{N} \sqrt{\tilde{M} \tau N_n} \\
&\le \tau + \tau \mathbb{E}[R_{\text{coverage}}] + \sqrt{N \tilde{M} \tau \sum_{n=1}^{N} N_n} \qquad (14) \\
&\le \tau + \tau \mathbb{E}[R_{\text{coverage}}] + \sqrt{N \tilde{M} \tau 2T/\tau}\ , \qquad (15)
\end{aligned}$$

where in Equation (14) we use the Cauchy-Schwarz inequality. For Equation (15) we used $\sum_{n=1}^{N} N_n \le 2N = 2T/\tau$ and the inequality in the discussion at the beginning of Section 3.1. Let $\varepsilon_n = B_{\tau, N_n, \tilde{M}}/\tau$. By Lemma D.5 we can set $\gamma = \left(\frac{\tilde{M} K \log K}{N}\right)^{1/3}$ and bound $\mathbb{E}[R_{\text{coverage}}]$ to get

$$\begin{aligned}
R(\pi_{\text{OS}}, T, N, \tilde{M}) &\le \tau + \tau \left((\tilde{M}^4 N^2 K \log k)^{1/3} + \tilde{M} \sum_{n=1}^{N} \varepsilon_n \right) + \sqrt{2N\tilde{M}T} \\
&\le \tau + \tau (\tilde{M}^4 N^2 K \log k)^{1/3} + \tilde{M} \sqrt{2N\tilde{M}T} \\
&= T/N + T N^{-1/3} (\tilde{M}^4 K \log K)^{1/3} + \tilde{M} \sqrt{2\tilde{M}NT}\ .
\end{aligned}$$

Here, the second inequality follows from Cauchy-Schwarz inequality and the same argument as above for bounding $\sum_{n=1}^{N} N_n$, and the last step is just because $\tau = T/N$.

□

If $N \ge N_1 \doteq \left(\frac{T^3 (K \log K)^2}{\tilde{M}}\right)^{1/5}$ and $M \le (K \log K)^{1/3}$ (large number of changes and small number of optimal arms), then our regret upper bound is $\tilde{O}(\tilde{M}^{3/2}\sqrt{NT})$, and the regret of `OS-BASS` improves upon the $\tilde{O}(\sqrt{KTN})$ bound of standard non-stationary bandit algorithms (such as `AdSwitch`).

If $N \le N_1$ and $M \le (K \log K)^{1/3}$, and we can obtain an improved bound by using a larger number of segments. Note that we could replace N with an arbitrary number of segments, N', in the analysis above. By choosing $N' = \left(\frac{T^3 (K \log K)^2}{\tilde{M}}\right)^{1/5}$ and $M \le (K \log K)^{1/3}$ segments, each of size $\tau' = T/N'$, the bound improves to $\tilde{O}(\tilde{M}^{7/5} (K \log K)^{1/5} T^{4/5})$.

If $N \le N_2 \doteq \tilde{M}^{14/5} (T/K)^{3/5} (\log K)^{2/5}$ (even small number of changes), then $\sqrt{KNT} \le \tilde{M}^{7/5} (K \log K)^{1/5} T^{4/5}$. In this case, the simple baseline of $\tilde{O}(\sqrt{KTN})$ is smaller than our bound, and the learner should simply play a standard non-stationary bandit algorithm. Notice that $N_2 \le N_1$ as long as $M \le K^{1/3}$.

E Partial monitoring and bandit meta-learning

Partial monitoring is a general framework in online learning that disentangles rewards and observations (information). It is a game where the learner has Z actions and the adversary has D actions,

Algorithm 5 The partial monitoring algorithm

```
Exploration probability $p \in (0,1)$, learning rate $\eta > 0$, base costs $\mathbf{C}_{\texttt{info}}, \mathbf{C}_{\texttt{hit}}, \mathbf{C}_{\texttt{miss}}$
for $n = 1, 2, \dots, N$ do
    With probability $p$, let $E_n = \text{Exr}$ and otherwise $E_n = \text{Ext}$
    if $E_n = \text{Exr}$ then
        Observe the best arms $S_n^*$ of this round and for all $i \in \text{Ext}$ experts, observe cost $C_{i,S_n^*}$ and let $\widehat{C}_n(i) = (C_{i,S_n^*} - \mathbf{C}_{\texttt{hit}})/p$
        Update exponential weights $Q_{n,i} \propto \exp(-\eta \sum_{\tau=1}^{n} \widehat{C}_n(i))$  Suffer cost $\mathbf{C}_{\texttt{info}}$
    else
        Sample $S_n \sim Q_{n-1}$
        Suffer (but do not observe) cost $\mathbf{C}_{\texttt{hit}}$ if $S_n^* \cap S_n \neq \emptyset$ and suffer cost $\mathbf{C}_{\texttt{miss}}$ otherwise
    end if
end for
```

and it is characterized by two $Z \times D$ matrices (not observed): matrix C maps the learner's action to its cost given the adversary's choice, and matrix X maps the learner's action to its observation given the adversary's choice. In all generality, we consider bandit meta-learning problems with $Z+1$ learner actions: an Exr action that provides information for a cost $\mathbf{C}_{\texttt{info}}$, and Z other actions that do not provide information but have a hidden cost $\mathbf{C}_{\texttt{hit}}$ or $\mathbf{C}_{\texttt{miss}}$ depending on whether the chosen action had low or high cost respectively.

As defined in the introduction, a bandit subset-selection problem is realizable when there is a subset of size M that contains an optimal arm in all rounds. Otherwise, the problem is called agnostic.

In our bandit subset-selection problem, $Z = \binom{K}{M} \leq K^M$ and the adversary can have up to 2^K choices depending on the realizable or agnostic nature of the problem. We have $D = M$ if the problem is realizable and if the adversary is constrained to picking a unique optimal arm in each round. For example, let $M = 2$ and $K = 4$. There are $Z + 1 = \binom{4}{2} + 1 = 7$ learner actions and only $D = 2$ possible choices for the adversary

$$\begin{pmatrix} \text{Exr} \\ \{1,2\} = x^* \\ \{1,3\} \\ \{1,4\} \\ \{2,3\} \\ \{2,4\} \\ \{3,4\} \end{pmatrix} \to C = \begin{pmatrix} \mathbf{C}_{\texttt{info}} & \mathbf{C}_{\texttt{info}} \\ \mathbf{C}_{\texttt{hit}} & \mathbf{C}_{\texttt{hit}} \\ \mathbf{C}_{\texttt{hit}} & \mathbf{C}_{\texttt{miss}} \\ \mathbf{C}_{\texttt{hit}} & \mathbf{C}_{\texttt{miss}} \\ \mathbf{C}_{\texttt{miss}} & \mathbf{C}_{\texttt{hit}} \\ \mathbf{C}_{\texttt{miss}} & \mathbf{C}_{\texttt{hit}} \\ \mathbf{C}_{\texttt{miss}} & \mathbf{C}_{\texttt{miss}} \end{pmatrix}, \qquad X = \begin{pmatrix} 1 & 2 \\ \perp & \perp \\ \vdots & \vdots \\ \perp & \perp \end{pmatrix}.$$

The symbol $\perp$ is used to denote no observations. We use $C_{i,y}$ to denote the cost of action $i \in \{\text{Exr}, x_1, \dots, x_Z\}$ when adversary chooses $a \in [D]$. Thanks to this reduction, we can leverage the partial monitoring literature to obtain an algorithm and the corresponding bounds for our problem as well. We detail this process below. Note that using the vocabulary of online learning, the learner's actions are referred to as "experts".

Next, we describe an algorithm based on the Exponentially Weighted Average (EWA) forecaster. The learner estimates the cost matrix by importance sampling when action Exr is chosen. When Ext is chosen, the learner samples an expert according to EWA weights that depend on the estimated cost matrix. The pseudo-code of the method is shown in Algorithm 5.

To analyze the algorithm, we consider the realizable and agnostic cases. In the realizable case, there a subset of size M that contains an optimal arm in all rounds. In this case, the exponential weights distribution reduces to a uniform distribution over the subsets that satisfy this condition.

Theorem E.1. *Consider the partial monitoring algorithm shown in Algorithm 5. In the agnostic case, with the choice of* $p = O\left(\left(\frac{\mathbf{C}_{miss}^2 \log Z}{\mathbf{C}_{info}^2 N}\right)^{1/3}\right)$ *and* $\eta = O\left(\left(\frac{\log^2 Z}{\mathbf{C}_{info} \mathbf{C}_{miss}^2 N^2}\right)^{1/3}\right)$*, the regret of* *the algorithm is bounded as $O((\boldsymbol{C}_{info}\boldsymbol{C}^2_{miss}N^2\log Z)^{1/3})$. In the realizable case, with the choice of $p=\sqrt{\frac{\boldsymbol{C}_{miss}\log Z}{\boldsymbol{C}_{info}N}}$ and $\eta=1$, the regret of the algorithm is bounded as $O(\sqrt{\boldsymbol{C}_{info}\boldsymbol{C}_{miss}N\log Z})$.*

Proof. Let function $f_n:[Z+1]\times[D]\to\mathbb{R}^{Z+1}$ be defined by

$$f_n(k,X_{k,y})_i=\mathbf{1}\{k=\textsc{Exr}\}(C_{i,y}-\mathbf{C}_{\texttt{hit}})\ .$$

Therefore, $\sum_{k=1}^{Z+1}f_n(k,X_{k,y})_i=C_{i,y}-\mathbf{C}_{\texttt{hit}}$. With probability p, let $E_n=\textsc{Exr}$ and otherwise $E_n=\textsc{Ext}$. Let $C_n(i)=C_{i,Y_n}$. Define cost estimator

$$\widehat{C}_{n,i}=\frac{f_n(E_n,X_{E_n,Y_n})_i}{p}=\frac{\mathbf{1}\{E_n=\textsc{Exr}\}(C_n(i)-\mathbf{C}_{\texttt{hit}})}{p}\ .$$

Let Q_n be the weights of the EWA forecaster defined using the above costs. For any i, we have $\mathbb{E}(\widehat{C}_n(i))=C_n(i)-\mathbf{C}_{\texttt{hit}}$. Let E_n be the learner's decision in round n, that is either $\textsc{Exr}$ or a subset chosen by EWA, in which case it is denoted by x_n. We have

$$\mathbb{E}(C_n(E_n))=p\mathbf{C}_{\texttt{info}}+(1-p)\mathbb{E}(C_n(S_n))\ .$$

Let S^* be the optimal subset. By the regret bound of EWA (Cesa-Bianchi and Lugosi, 2006),

$$\sum_{n=1}^N\widehat{C}_n(S_n)-\sum_{n=1}^N\widehat{C}_n(S^*)\le\frac{\log Z}{\eta}+\frac{\eta}{2}\sum_{n=1}^N\|\widehat{C}_n\|_\infty^2\ .$$

Thus,

$$\begin{aligned}\sum_{n=1}^N\mathbb{E}(C_n(E_n))-\sum_{n=1}^N\mathbb{E}(C_n(S^*))&\le\mathbf{C}_{\texttt{info}}\sum_{n=1}^N p+\frac{\log Z}{\eta}+\frac{\eta}{2}\sum_{n=1}^N\mathbb{E}(\|\widehat{C}_n\|_\infty^2)\\&\le\mathbf{C}_{\texttt{info}}\sum_{n=1}^N p+\frac{\log Z}{\eta}+\frac{\eta\mathbf{C}^2_{\texttt{miss}}}{2}\sum_{n=1}^N\frac{1}{p}\ .\end{aligned}$$

With the choice of $p=O((\mathbf{C}_{\texttt{miss}}/\mathbf{C}_{\texttt{info}})^{2/3}(\log^{1/3}Z)/N^{1/3})$ and $\eta=O((\log^{2/3}Z)/(\mathbf{C}^{2/3}_{\texttt{miss}}\mathbf{C}^{1/3}_{\texttt{info}}N^{2/3}))$, the regret of the partial monitoring game is bounded as $O(\mathbf{C}^{2/3}_{\texttt{miss}}\mathbf{C}^{1/3}_{\texttt{info}}N^{2/3}\log^{1/3}Z)$. The regret scales logarithmically with the number of experts, and is independent of the number of adversary choices.

Next, we show a fast $O(\sqrt{N})$ rate when the optimal expert always has small cost. More specifically, we assume that $C_n(S^*)=\mathbf{C}_{\texttt{hit}}$ for the optimal expert S^*. The fast rate holds independently of the relative values of $\mathbf{C}_{\texttt{hit}},\mathbf{C}_{\texttt{info}}$, and $\mathbf{C}_{\texttt{miss}}$. The algorithm can also be implemented efficiently.

Let $\widehat{\ell}_n=p\widehat{C}_n/\mathbf{C}_{\texttt{miss}}$, which is guaranteed to be in $[0,1]$. Notice that $\sum_{n=1}^N\widehat{\ell}_n(S^*)=0$ as $C_n(S^*)=\mathbf{C}_{\texttt{hit}}$ by assumption. In this case, the regret of EWA is known to be logarithmic:

$$\sum_{n=1}^N\widehat{\ell}_n(S_n)-\sum_{n=1}^N\widehat{\ell}_n(S^*)=O(\log Z)\ .$$

Thus,

$$\sum_{n=1}^N\mathbb{E}(C_n(E_n))-\sum_{n=1}^N\mathbb{E}(C_n(S^*))\le\mathbf{C}_{\texttt{info}}\sum_{n=1}^N p+\frac{\mathbf{C}_{\texttt{miss}}\log Z}{p}\ .$$

Therefore, with the choice of $p=\sqrt{\frac{\mathbf{C}_{\texttt{miss}}\log Z}{\mathbf{C}_{\texttt{info}}N}}$,

$$\sum_{n=1}^N\mathbb{E}(C_n(E_n))-\sum_{n=1}^N\mathbb{E}(C_n(S^*))\le O(\sqrt{\mathbf{C}_{\texttt{info}}\mathbf{C}_{\texttt{miss}}N\log Z})\ .$$

The meta-regret scales logarithmically with the number of experts, and is independent of the number of adversary choices. Given that the optimal expert is known to have small loss in all rounds, the learner can eliminate all other experts. Therefore, the EWA strategy reduces to a uniform distribution over the surviving experts.

□

E.1 Proof of Theorem 2.5

E-BASS is constructed as a special case of the EWA algorithm above, where the sampling distribution at each Ext round is simply the uniform distribution over the surviving experts. The proof of Theorem 2.5 is therefore a direct consequence of the more general analysis done for the EWA forecaster in Theorem E.1 above.

Proof. The BAI algorithm might return a number of extra arms in addition to the optimal arm. However, since with high probability the optimal arm is always in the surviving set, the cost estimate for the optimal subset is always zero, and costs of all other subsets are under-estimated. Therefore, if S_n is the expert (subset) selected in task n and S^* is the optimal subset, by fast rates of the previous section,

$$\sum_{n=1}^{N} \mathbb{E}(C_n(S_n)) - \sum_{n=1}^{N} \mathbb{E}(C_n(S^*)) \leq O(\sqrt{\mathbf{C}_{\mathtt{info}}\mathbf{C}_{\mathtt{miss}} N \log Z}) \, .$$

Given that with high probability the optimal arm is always in the surviving set and therefore $C_n(S^*) = \mathbf{C}_{\mathtt{hit}}$,

$$\begin{aligned} R_T = \sum_{n=1}^{N} \mathbb{E}\left(\tau r_n(a_n^*) - \sum_{t=1}^{\tau} r_n(A_{n,t})\right) &\leq \sum_{n=1}^{N} \mathbb{E}(C_n(S_n)) \leq N\mathbf{C}_{\mathtt{hit}} + O(\sqrt{\mathbf{C}_{\mathtt{info}}\mathbf{C}_{\mathtt{miss}} N \log Z}) \\ &= N\sqrt{M\tau} + O(\sqrt{\mathbf{C}_{\mathtt{info}}\mathbf{C}_{\mathtt{miss}} N \log Z}) \\ &= N\sqrt{M\tau} + O(\tau^{3/4} K^{1/4} \sqrt{NM \log(K)} \, , \end{aligned}$$

where the first inequality holds by the fact that $\mathbb{E}(C_n(S_n))$ is an upper bound on the regret for task n. □

F Further experimental details and results

This section consists of further experimental details and results. We use the code in the following repository: https://anonymous.4open.science/r/meta-bandit-760E/README.md. We used a server machine with the following configuration: OS: Ubuntu 18.04 bionic, Kernel: x86_64 Linux 4.15.0-176-generic, CPU: Intel Core i9-10900K @ 20x 5.3GHz, GPU: GeForce RTX 2080 Ti, RAM: 128825 MiB, DISK: 500 GB SSD.

F.1 Setup

In each experiment, the adversary first samples the size M set of optimal arms, $S^* := \cup_n S_n^*$, uniformly at random (without replacement) from $[K]$. The mean reward of task n, $r_n \in \mathcal{R} = [0,1]^K$, is then generated according to the experiment setup as described in the following.

The optimal arm: We categorize the experiments into three settings based on how the optimal arm is generated: i) *non-oblivious adversarial*, ii) *oblivious adversarial*, and iii) *stochastic*.

i) In the adversarial setting with a *non-oblivious* adversary, the adversary peeks into the learner's set of discovered arms, S_n, at the end of each task. With probability q_n (see Equation (8)), the adversary chooses a new optimal arm uniformly at random from $[K]\backslash S_n$. Otherwise, the next optimal arm is chosen uniformly at random from S_n.

ii) The *oblivious* adversary is applicable against any learner even if the learner does not maintain a set of discovered arms. Here the adversary simulates an imaginary `G-BASS` algorithm with a minimax optimal p_n (see Equation (8)). Then it samples new optimal arms and generates the reward sequence with respect to this imaginary learner. This is the same as the non-oblivious adversary except here the adversary plays against an imaginary learner.

iii) In the *stochastic* setting, for each task n, the environment samples the optimal arm uniformly at random from the optimal set, i.e., $a_n^* \sim \text{Uniform}(S^*)$.

Note that in the *non-oblivious* setting, the rewards are generated at the start of each task, according to the learner's discovered arms. In the other settings, however, rewards of all the tasks could be generated at the very beginning, independently of the learner.

The sub-optimal arms (min gap): Based on the discussion after Assumption 2.1, the minimum gap for the assumption to hold is

$$r_n(a^*) - \max_{a \neq a^*} r_n(a) \geq \Delta \ ,$$

where $\Delta = \Theta(\sqrt{K \log(N/\delta)/\tau})$. After generating the optimal arm, depending on the setting, the rewards of other arms are generated in two ways: 1) with a minimum gap condition uniformly at random in $[0, r_n(a_n^*) - \Delta)$ and 2) without a minimum gap condition uniformly at random in $[0, r_n(a_n^*)]$. In the second case, the mean reward is generated such that the gap condition is violated by at least 1 sub-optimal arm.

Task length and PE: As we know, task length plays an important role in regard to PE performance. In the case where Assumption 2.1 holds, we set the phase length based on Δ and make sure τ is longer than the length of the first phase of PE. For more details, see the analysis of PE (Auer and Ortner, 2010) in exercise 6.8 (elimination algorithm) of Lattimore and Szepesvári (2020).

Assumption 2.1: We have two types of experiments considering Assumption 2.1: i) In the experiments where Assumption 2.1 is supposed to hold, we make sure the task length is longer than the first phase of PE and the minimum gap condition holds (case 1 in the discussion on the min gap). ii) In the tasks where Assumption 2.1 is supposed to be violated, we use case 2 in the discussion on the min gap above with a small τ so that PE fails.

F.2 Further experiments

Next, we report the experimental results under different conditions. Error bars are ±1 standard deviation, computed over 5 independent runs.

Figure 2 shows the result when Assumption 2.1 holds, where `G-BASS` almost matches the `Opt-MOSS`, outperforming the other algorithms. Figure 4 shows the results when Assumption 2.1 does not hold. In this case, we observe that `OS-BASS` outperforms the other algorithms and is close to `Opt-MOSS`. Here `G-BASS` is less effective and sometimes has large variance due to the failure of PE.

Figure 5 demonstrates the performance of `E-BASS` when Assumption 2.1 holds. We can see that `E-BASS` outperforms all other baselines. For large M, `G-BASS` seems to be more effective than the others. Figure 6 compares `E-BASS` to the other algorithms when Assumption 2.1 does not hold. `OS-BASS` is competitive with `E-BASS` and outperforms it for larger M. Comparing Figure 5 and Figure 6, we can see that `G-BASS` and `E-BASS` perform better if Assumption 2.1 holds.

Figures 7 and 8 show the experimental results with a non-oblivious adversary. We observe similar trends as in the previous experiments.

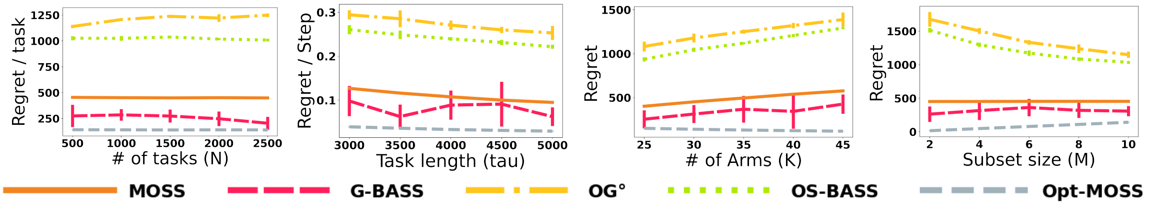

Figure 2: Oblivious adversarial setting with Assumption 2.1. Default setting: $(N, \tau, K, M) = (500, 4500, 30, 10)$. G-BASS is near-optimal on all tasks. Left to Right: Regret as a function of N, τ, K, and M.

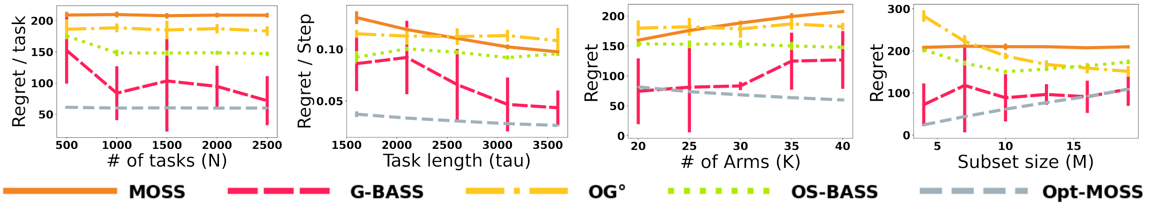

Figure 3: Oblivious adversarial setting where assumption 2.1 is almost satisfied (only the minimum gap condition violated, large task length). Default setting: $(N,\ \tau,\ K, M) = (1000,\ 1600,\ 40,\ 10)$. Left to Right: Regret as a function of N, τ, K, M.

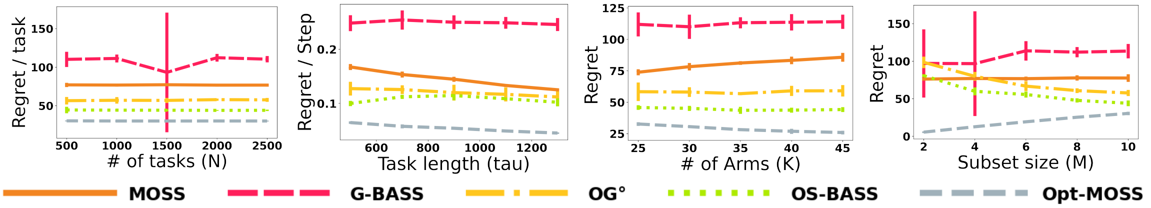

Figure 4: Oblivious adversarial setting without assumption 2.1 (no minimum gap and small task length). Default setting: $(N, \tau, K, M) = (500, 450, 30, 10)$. OS-BASS is near-optimal on all tasks and outperforms OG°. Left to Right: Regret as a function of N, τ, K, M.

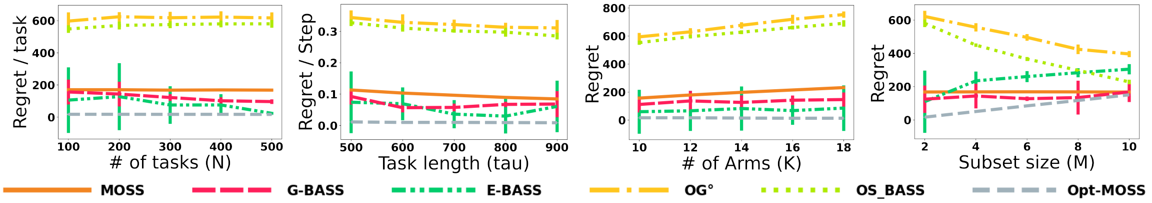

Figure 5: E-BASS's performance in the oblivious adversarial setting with Assumption 2.1. Default setting: $(N,\ \tau,\ K, M) = (400,\ 2000,\ 11,\ 2)$. E-BASS outperforms other algorithms. Left to Right: Regret as a function of N, τ, K, M.

The results for the stochastic setting are shown in Figures 9 and 10. In Figure 9 it seems that G-BASS performs the best while MOSS is closer to the oracle Opt-MOSS. However, in Figure 10 OS-BASS outperforms G-BASS and the other algorithms and gets closer to the oracle baseline. We can see in the stochastic setting the variance is higher than the adversarial setting.

In all the experiments, OS-BASS outperforms OG° which confirms the choice of γ and τ in our analysis for OS-BASS.

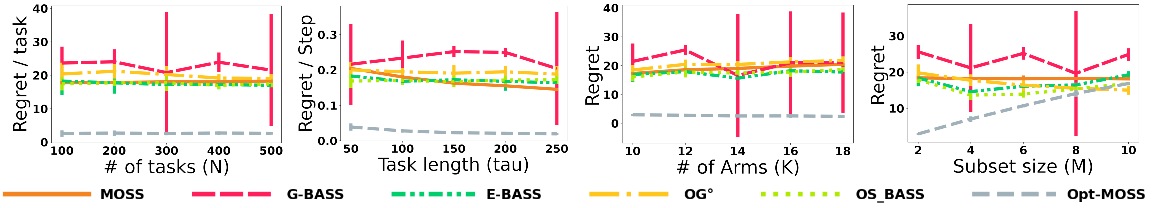

Figure 6: E-BASS's performance in the oblivious adversarial setting without assumption 2.1 (no minimum gap and small task length). Default setting: $(N,\ \tau,\ K, M) = (400,\ 100,\ 11,\ 2)$. E-BASS and OS-BASS win all the settings, while MOSS is competitive. Left to Right: Regret as a function of $N,\ \tau,\ K, M$.

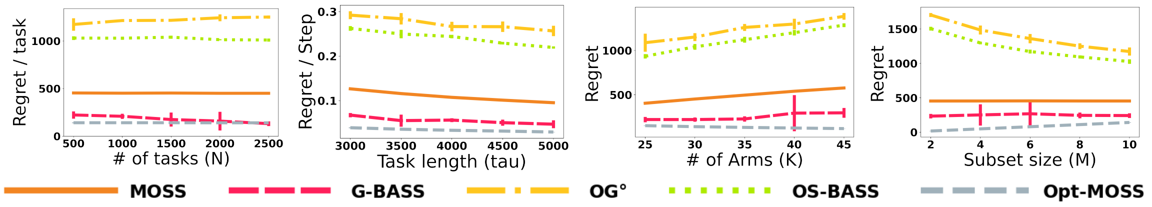

Figure 7: The non-oblivious adversarial setting, where Assumption 2.1 holds. Default setting: $(N, \tau, K, M) = (500, 4500, 30, 10)$. G-BASS is near-optimal on all tasks. Left to Right: Regret as a function of $N,\ \tau,\ K, M$.

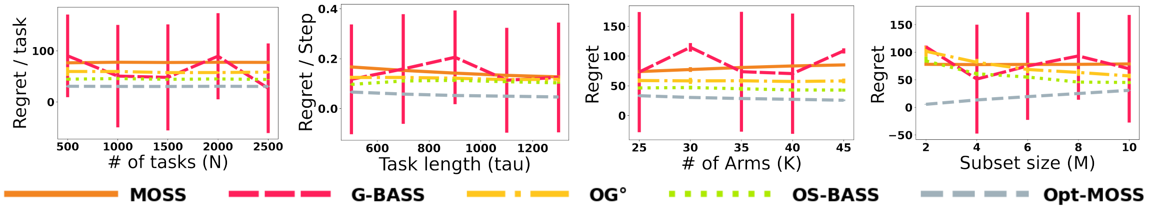

Figure 8: Non-oblivious adversarial setting without assumption 2.1 (no minimum gap and small task length). Default setting: $(N, \tau, K, M) = (500, 450, 30, 10)$. OS-BASS mostly outperforms the other method. G-BASS has a high variance as PE fails in this experiment. Left to Right: Regret as a function of $N,\ \tau,\ K, M$.

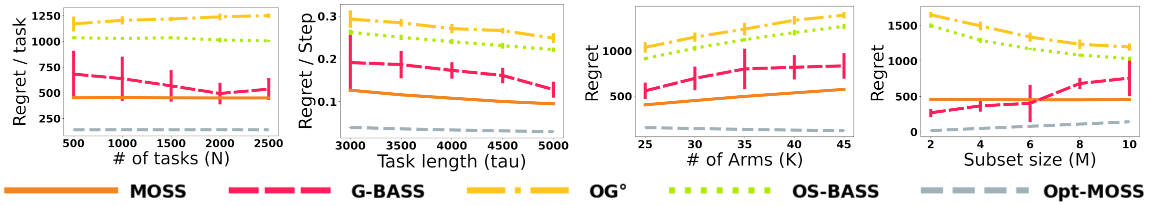

Figure 9: Stochastic setting, where Assumption 2.1 holds. Default setting: $(N, \tau, K, M) = (500, 4500, 30, 10)$. G-BASS and MOSS have the best performance in all the experiments. Left to Right: Regret as a function of $N,\ \tau,\ K, M$.

G Other Related Work

Slate bandits. The reduction in Section 3.1 is an instance of slate bandit problems with a non-separable cost function (Dimakopoulou et al., 2019; Rhuggenaath et al., 2020; Kale et al., 2010). Rhuggenaath et al. (2020) study the problem in the stochastic setting, where the reward parameter is fixed throughout the game. Merlis and Mannor (2019) study a problem that includes

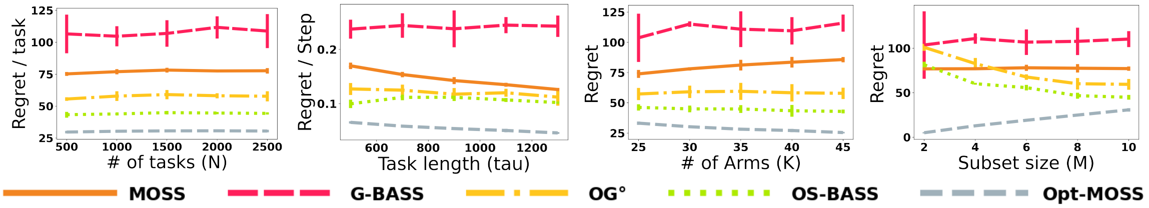

Figure 10: Stochastic setting without assumption 2.1 (no minimum gap and small task length). Default setting: $(N, \tau, K, M) = (500, 450, 30, 10)$. `OS-BASS` has the best performance in all experiments. Left to Right: Regret as a function of N, τ, K, M.

the probabilistic maximum coverage as a special case. They obtain problem-dependent logarithmic and problem-independent $O(\sqrt{N})$ regret bounds. However, the feedback structure in this work is richer than our setting. Applied to our problem, they assume that in each segment, for each item and task pair, a random variable is observed whose expected value is the probability that the item is the optimal arm in that task.

Bandits with very large action spaces. As K grows very large, our bandit meta-learning problem is akin to infinitely many armed bandits (Berry et al., 1997; Wang et al., 2008; Bonald and Proutiere, 2013; Carpentier and Valko, 2015; Chan and Hu, 2020) and countable-armed bandits (Kalvit and Zeevi, 2020) though these settings do not have a meta-learning aspect.

H Code

The code is available at `https://github.com/duongnhatthang/meta-bandit`