Efficient Swap Regret Minimization in Combinatorial Bandits

Andreas Kontogiannis[*1,3], Vasilis Pollatos[*2,3], Panayotis Mertikopoulos[3,4], and Ioannis Panageas[3,5]

[1]National Technical University of Athens, School of Electrical and Computer Engineering
[2]National and Kapodistrian University of Athens, Department of Mathematics
[3]Archimedes, Athena Research Center, Greece
[4]Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG, 38000 Grenoble, France
[5]University of California, Irvine

July 11, 2026

Abstract

This paper addresses the problem of designing efficient no-swap regret algorithms for combinatorial bandits, where the number of actions N is exponentially large in the dimensionality of the problem. In this setting, designing efficient no-swap regret translates to sublinear – in horizon T – swap regret with polylogarithmic dependence on N. In contrast to the weaker notion of external regret minimization – a problem which is fairly well understood in the literature – achieving no-swap regret with a polylogarithmic dependence on N has remained elusive in combinatorial bandits. Our paper resolves this challenge, by introducing a no-swap-regret learning algorithm with regret that scales polylogarithmically in N and is tight for the class of combinatorial bandits. To ground our results, we also demonstrate how to implement the proposed algorithm efficiently – that is, with a per-iteration complexity that also scales polylogarithmically in N – across a wide range of well-studied applications.

[*]Equal contribution. Corresponding authors andreaskontogiannis@mail.ntua.gr, vaspoll@math.uoa.gr.

1 Introduction

Background on adversarial combinatorial bandits. Throughout this paper, we consider a general class of combinatorial online learning problems that unfold as follows:

1. At each day (or round) $t = 1, 2, \ldots, T$, of a repeated decision process, the learner selects a specific combination of up to m elements from some ground set with d elements in total.

2. This choice triggers a reward for each element in the ground set; subsequently the learner receives as payoff the sum of the rewards of the chosen elements, and the process repeats.

If the sequence of reward vectors is arbitrary rather than drawn from a stationary distribution, the problem is known as the *adversarial combinatorial bandit problem* [CBL12, LS20]. Owing to its flexibility, this learning paradigm models a wide array of applications, from path planning [KV05, BEDL06, GLLO07], to resource allocation, recommender systems, and beyond [KRS10, ABL14, WAEK15]. Crucially, the total number of actions N available to the learner typically grows *exponentially* in the number of elements of the ground set; i.e., $N = \mathcal{O}(d^m)$.

This "curse of dimensionality" has led to extensive research, resulting in several algorithms with optimal (or near-optimal) guarantees for *external* regret – the cumulative gap between the learner's reward and the best fixed choice in hindsight. In general, this literature revolves around two main axes: (a) the optimality of the regret guarantees provided in terms of d, m, and T, and (b) the per-iteration complexity of the methods proposed to achieve said guarantees.

To give an idea of the guarantees and techniques employed in this setting, the original ComBand algorithm of [CBL12] – also known in slightly different contexts as GeometricHedge [DKH07] or Exp2 [BCBK12] – enjoys an external regret bound of $\widetilde{\mathcal{O}}(\sqrt{mdT})$ in specific combinatorial bandit problems, such as m-sets, path planning, spanning trees, perfect matchings, etc. In general, the per-iteration complexity of ComBand can be exponential, but it can be implemented efficiently in many settings with an amenable combinatorial structure – using for example the kernelization approach of [KPF+25]. In a similar vein, to resolve implementation issues having to do with sampling from a low-support distribution, [KRS10, CTMSP+15] proposed an approach based on Carathéodory decomposition techniques, in order to perform efficient updates in d dimensions instead of N.

From external to swap regret: motivation and challenges. Driven in no small measure by applications to game theory and reinforcement learning, there is a strong push in the literature toward *more powerful* notions of regret – such as internal and, more importantly for our purposes, *swap regret*. There are two main reasons for this: First, from an online learning standpoint, external regret is not a particularly informative notion in environments where a fixed action policy severely underperforms relative to more reactive benchmarks. Second, from a multi-agent perspective, no-external-regret learning is not rationalizable because of well-known examples of negative external regret policies in which every player plays a strictly dominated strategy at all times [VZ13].

Both concerns above are mitigated by the notion of *swap regret*, so dubbed by [BM07], and tracing its roots to the internal regret considerations already present in the original work of [FV97] on calibrated forecasting.[1] Swap regret quantifies the additional utility a learner could have achieved, in hindsight, by retroactively modifying her played strategies according to the *best fixed swap function* as opposed to the best fixed action in hindsight of the external regret.

[1]The main difference between internal and swap regret is that the latter allows multiple strategy swaps at the same time. We focus on the latter because of the added generality.

Algorithm	Swap Regret	Practical Bound	Per-Iteration
[Sto05]	$\mathcal{O}\left(d^m\sqrt{mT\log d}\right)$	$\text{poly}(d^m)$	$\exp(d^m)$
[BM07]	$\mathcal{O}\left(d^m\sqrt{md^mT\log d}\right)$	$\text{poly}(d^m)$	$\text{poly}(d^m)$
[Ito20]	$\mathcal{O}\left(d^m\sqrt{T}\right)$	$\text{poly}(d^m)$	$\text{poly}(d^m)$
[DDFG24]	$\mathcal{O}\left(\frac{T\log(m\log T)}{\log(T/d^m)}+\sqrt{d^mT\log\left(d^mT\right)}\right)^*$	$\text{poly}(d^m)$	$\text{poly}(d^m,\log T)$
This work	$\mathcal{O}\left(\frac{T\log(d\log T)}{\log(T)}\right)$	$\text{poly}(d,m)$	$\text{poly}(m,d,\log T)$
Lower Bound†	$\Omega\left(\frac{T}{\log^6 T}\right)$, for $T\le\exp(\Omega(d^{1/14}))$		

Table 1: Comparative analysis of results on swap regret for combinatorial bandits and algorithms' per-iteration complexity. Practical bound denotes the swap regret in the practical regime where $T=\text{poly}(d,m)$. $*$: [DDFG24] considers a stronger notion of swap regret, where the max operator is inside expectation, but it is also inefficient as it has $\mathcal{O}(N)$-per-iteration complexity. $\dagger$: The lower bound is a direct application of Theorem 4.1 in [DFG+24] (see Corollary 4.3). It holds for $m\ge d^{12/13}$.

On the downside, swap regret minimization is a significantly more challenging problem than vanilla, external regret minimization. In his PhD thesis, [Sto05] proposed an algorithm with a swap regret bound of $\mathcal{O}(N\sqrt{T\log N})$, albeit with a per-iteration complexity of $\exp(\mathcal{O}(N))$. At around the same period, [BM07] proposed a $\mathsf{poly}(N)$-time algorithm for the full information setting which achieves a swap regret bound of $\mathcal{O}(\sqrt{NT\log N})$, using a swap-to-external regret reduction argument. [BM07] also extended their algorithm to the bandit setting, achieving a slightly worse regret bound of $\mathcal{O}(N\sqrt{NT\log N})$. This bound was later improved by [Ito20] in the bandit setting, leading to an algorithm with a $\mathsf{poly}(N)$ per-iteration complexity that exhibits $\mathcal{O}(N\sqrt{T})$ swap regret bound.

One of the most challenging questions in the online learning community is how to design computationally efficient, no-swap-regret algorithms for which both the swap regret and the per-iteration complexity depend polylogarithmically on N. This was highlighted as an open question in [BM07]. In the full information setting, the question was resolved very recently by [DDFG24, PR24], with both approaches obtaining a swap regret bound of $\mathcal{O}(T/\log T)$. In the bandit setting, without assuming any structure, existing lower bounds for the bandit setting (e.g., see the lower bound for external regret in [ACBFS02], which carry over to swap regret) preclude polylogarithmic dependence on N. The approach of [DDFG24] was also applied to the unstructured bandit setting, but, similarly to previous works, the resulting algorithm is impractical for large action spaces, as both the regret bound and the per-iteration complexity are polynomial in N.

Although the aforementioned approaches are applicable to the combinatorial bandit setting, they fail to achieve the desired polylogarithmic dependence of the swap regret and the per-iteration complexity on N. In this paper, we seek to examine whether the *extra structure present in combinatorial bandits* can be exploited to yield a positive answer to the following question:

> *Can we design efficient no-swap-regret learning algorithms for combinatorial bandits, achieving swap regret guarantees and per-iteration complexity that scale polylogarithmically in the exponentially large number of actions $N=\mathcal{O}(d^m)$?*

Our contributions. Our paper answers the above question affirmatively. Specifically, we provide the first algorithm that achieves no-swap-regret with polylogarithmic dependence on the

large action space of combinatorial bandits. Our main contribution is summarized in the following theorem.

Theorem (Abridged; Formally stated in Theorem 4.1)**.** *There exists a combinatorial bandit algorithm whose swap regret is at most* $\mathcal{O}\left(T \log(d \log T)/\log T\right)$.

A detailed comparison with prior works is presented in Table 1. Importantly, only our swap regret upper bound remains meaningful in the realistic regime where $T = \text{poly}(d, m)$. Remarkably, via an application (see Corollary 4.3) of the lower bound result of [DFG+24], in the above regime of interest, our upper bound is tight, in the sense that there is no T^p-no-swap-regret scheme (where p is a constant less than 1) with polylogarithmic dependence on d^m. In addition, using well-established techniques from combinatorial bandits, our algorithm can be implemented efficiently requiring time $\text{poly}(d, m)$ in a wide array of well-known applications, including online shortest paths, m-sets, spanning trees and permutations.

Technical Overview. Our approach is inspired by the results of [DDFG24, PR24], which provide a reduction from swap to external regret for the full information setting that avoids introducing a polynomial dependence of the regret on the number of actions. Our algorithmic framework (Algorithm 1) utilizes a master learner (that serves as the online learner of the combinatorial bandit setting) which is a uniform mixture over parallel learners, dubbed ScaleLearners. Each ScaleLearner deploys an abstract combinatorial bandit algorithm, dubbed Lazy-CombAlg, and is parameterized by a different scale of laziness (that is, the learners only update their policies periodically and not at each day as is typical). In our framework, Lazy-CombAlg is effectively a lazy version of a no-external-regret combinatorial bandit algorithm. We propose an implementation of Algorithm 1 via an efficient implementation for Lazy-CombAlg, namely Lazy-ComBCP (Algorithm 2), which is based on barycentric spanners and Carathéodory decomposition.

In contrast to the approaches of [DDFG24, PR24] in the full information setting, where the learning process is deterministic, we must contend with the stochasticity inherent in the partial information setting. In our framework, the master only observes bandit reward feedback. Based on this limited feedback, the master constructs an unbiased reward estimator. This same estimator is broadcasted to the ScaleLearners in order to be manipulated as their own reward estimator. The learning process follows an *idiomatic bandit online learning protocol* where each ScaleLearner updates its policy using the received reward estimators, coming from the master, which are *unbiased* w.r.t the master's policy but *biased* w.r.t. the ScaleLearner's policy. Crucially, the above protocol allows us to decompose the swap regret of the master learner into a sum of external regrets of the deployed Lazy-CombAlg instances of the ScaleLearners (see Lemma 3.2). The main challenge in bounding the external regret of each learner is that the stochasticity of the expectation of the regret is w.r.t. the master's policy, rather than the learner's as would be the standard. In the standard analysis of no-external-regret combinatorial bandit algorithms (e.g., [CBL12]), one often needs to control a term of the form $\mathbb{E}[M^T \Sigma^+ M]$ [2] in order to bound the variance of the reward estimator. Interestingly, in contrast to the above, to show no external regret of each learner (Theorems 3.9) in our framework, it turns out that we need to control $\mathbb{E}[M^T \Sigma^{+2} M]$ instead (see Lemma 3.14), which requires to carefully control the inverse of the minimum eigenvalue of the induced co-occurrence matrix $\mathbb{E}[MM^T]$ (see Lemma 3.13). Finally, we show that a regret upper bound of $\mathcal{O}(T \log(d \log T)/\log T)$ can yield anytime guarantees (Lemma 4.2) for any online learning algorithm achieving this bound – including

[2] M denotes the combinatorial action vector and Σ^+ denotes the pseudo-inverse of the *co-occurrence matrix* $\mathbb{E}[MM^T]$, see also Algorithm 1.

the proposed algorithms.

2 Preliminaries

Combinatorial Bandits. We consider the classic adversarial combinatorial bandit setting, proposed by [CBL12]. In this online learning setting, the decision maker has access to a finite action set $\mathcal{A} \subset \{0,1\}^d$, that is a subset of the d-dimensional hypecube. Specifically, for a given $m \leq d$, any action $M \in \mathcal{A}$ is a binary vector with at most m ones, that is, $\|M\|_1 \leq m$.

The online learning setting is as follows: The decision maker participates in a repeated process over days, where at each day $t = 1, 2, \ldots, T$, she has a distribution (denoted by policy) p_t over $\mathcal{A}$ and samples from it an action M_t. Then, she receives from the environment a reward $r_t = R_t \cdot M_t \in [0,1]$, where R_t is the reward vector for that day, with $R_t(j)$, for all $j \in [d]$, being the reward for the j-th coordinate. In this paper, we assume that the rewards are selected by an oblivious adversary, that is a type of opponent or environment that chooses the entire sequence of rewards in advance, before the learning algorithm starts. This means the adversary does not adapt to the actions or internal randomness of the learning algorithm.

Notions of Regret. The most standard and well-studied objective for the combinatorial bandit setting is the *external regret* which measures the difference between the algorithm's cumulative reward and the best fixed action in hindsight. Formally, for a given horizon of T days, the external regret is as follows:

$$\mathrm{Reg}_T = \max_{M^* \in \mathcal{A}} \sum_{t=1}^{T} R_t \cdot M^* - \mathbb{E}\left[\sum_{t=1}^{T} R_t \cdot M_t\right]. \tag{1}$$

Although the bound on external regret is crucial, it may be less appealing in highly dynamic environments where no single action consistently performs well throughout the entire history of rewards.

In this paper, we study *swap regret*, a notion of regret stronger than the external. Swap regret compares the decision maker's policy p_t against all strategies that can be derived from p_t by applying a swap function $\phi : \mathcal{A} \to \mathcal{A}$ to actions sampled from p_t. Formally, we define Φ to be the set containing all swap functions that map from $\mathcal{A}$ to $\mathcal{A}$. The swap regret is defined as follows:

$$\mathrm{SwapReg}_T = \max_{\phi \in \Phi} \mathbb{E}\left[\sum_{t=1}^{T} \Big(R_t \cdot \phi(M_t) - R_t \cdot M_t \Big)\right]. \tag{2}$$

3 Swap-ComBCP: A No-Swap-Regret Combinatorial Bandit Algorithm

In this section, we present our algorithmic approach. Typically, to bound swap regret, one first reduces it to external regret and then minimizes the latter (e.g., [BM07, DDFG24]). In this work, for the purposes of the reduction from swap to external regret, we use a backbone algorithmic framework for combinatorial bandits (see Algorithm 1) which achieves a decomposition of swap regret into a sum of external regret terms, as we show in Lemma 3.2. Then, to derive a vanishing swap regret guarantee, we need to control these external regret terms. Typically, in a swap-to-external-regret reduction, bounding the resulting external regret terms is a straightforward task; however, in our

setting it poses significant challenges, which will be discussed later in detail. We combine Algorithm 1 with an efficient combinatorial bandit algorithm, namely Lazy-ComBCP (Algorithm 2), and bound the external regret of the latter in Theorem 3.9. With this bound at hand, we get no-swap-regret guarantees for the overall algorithm (which we denote by Swap-ComBCP), stated in our main result (Theorem 4.1).

Multi-Scalar Laziness Against Swap Regret. Swap regret is a difficult solution concept to handle, because the reference swap function ϕ may exploit patterns that arise due to the variability of the decision maker's policy over time. Thus, one way to attack swap regret is by reducing the variability of the policy. In the extreme case, where the policy remains fixed during the whole horizon of play, swap deviations can achieve no higher utility than deviations to a fixed action, that is swap regret is upper bounded by the external regret. We state this formally in the following proposition.

Proposition 3.1 (Swap Regret is upper bounded by External Regret for time-invariant policies)**.** *For any policy $p \in \Delta(\mathcal{A})$ fixed in an interval $t = 1, \dots, T$, any sequence of rewards R_t and any swap function $\phi : \mathcal{A} \to \mathcal{A}$ it holds that*

$$\sum_{t=1}^{T} \sum_{M \in \mathcal{A}} p(M) R_t \cdot \phi(M) \leq \max_{M \in \mathcal{A}} \left\{ \sum_{t=1}^{T} R_t \cdot M \right\}. \tag{3}$$

Of course, keeping the policy fixed during the whole horizon would potentially result in extremely poor performance due to the lack of adaptivity. Thus, to exploit the above property we have to keep a balance between stability and adaptivity. For this purpose, as in [DDFG24, PR24], we use the concept of a *lazy* online learner, that is a learner that does not update its policy at every time-step but instead it only updates periodically, and keeps its policy freezed in the time period between two consecutive updates.

But again, a single lazy external regret learner is not enough to achieve vanishing swap regret against any adversary. To keep the right *balance in laziness*, one would need to use multiple learners, each with a different scale of laziness, and achieve the necessary balance by creating an *ensemble* of them. Lazy learners that freeze longer observe the learning task at a higher scale, thus we refer to them as ScaleLearners and the ensemble algorithm as the multi-scale master learner.

3.1 The algorithmic framework

Our framework (Algorithm 1) consists of the master learner (with policy $\hat{p}_t$) which coordinates *parallel* thread learners, referred to as ScaleLearners, all of which share the global variable T. Each ScaleLearner thread (dubbed ScaleLearner$_k$) is parameterized by a different scale k of laziness. For each $k \in [K]$, the corresponding ScaleLearner$_k$ divides the online learning horizon T into $\left\lceil \frac{T}{H^k} \right\rceil$ equal intervals and restarts its policy at the beginning of each interval (Step 13). At each interval (k, l), where $k \in [K]$ and $l \in \left\{1, \dots, \left\lceil \frac{T}{H^k} \right\rceil\right\}$, ScaleLearner$_k$ deploys Lazy-CombAlg$_{k,l}$, which – for now – serves as an abstract combinatorial bandit learner. We describe it in detail in Section 3.2. The common feature among all Lazy-CombAlg$_{k,l}$ learners is that they each perform H updates. What distinguishes them is how many days they parse between consecutive updates. In particular, Lazy-CombAlg$_{k,l}$ parses H^{k-1} days between two consecutive updates staying lazy in between. We denote this period by the term *meta-day*. Based on this, parameter k is used to control the level of laziness of the ScaleLearners, with the master policy being a uniform mixture of the policies of the ScaleLearners (Step 4), aiming to balance between laziness and no-regret learning.

Algorithm 1 Master Multi-scale Algorithm

1: **Require**: T, sequence of rewards $R_1, \dots, R_T$ | **Parameters**: K, H such that $T \in [H^{K-1}, H^K]$
2: **for** $t = 1, 2, ..., T$ **do**
3: Let $p_{k,t} \in \Delta^{|\mathcal{A}|}$ be the policy of the k-th ScaleLearner$_k$ ($k \in [K]$)
4: Let $\widehat{p}_t \in \Delta^{|\mathcal{A}|}$ be the policy of the master learner that plays uniformly over ScaleLearners; that is: $\widehat{p}_t = \frac{1}{K} \sum_{k \in [K]} p_{k,t}$
5: *Sampling:* Play action $M_t \sim \widehat{p}_t$ and receive reward $r_t = R_t \cdot M_t$ /* R_t is unobserved */
6: *Estimation:* Let $\Sigma_t = \mathbb{E}[MM^\top]$, where M has law $\widehat{p}_t$. Set $\widetilde{R}_t = r_t \Sigma_t^\dagger M_t$
7: *Broadcast* $(t, \widetilde{R}_t)$ to all ScaleLearner$_k$, for all $k \in [K]$
8:
9: **Procedure** ScaleLearner$_k$
10: **Require**: Lazy-CombAlg /* A lazy combinatorial bandit algorithm (e.g., Algorithm 2) */
11: **for** $l = 1, 2, \dots, \lceil T/H^k \rceil$ **do**
12: *Restart*: initialize the parameters of Lazy-CombAlg$_{k,l}$
13: *Run* Lazy-CombAlg$_{k,l}$ /* Play for H^k days, policy updated every H^{k-1} days */

The above steps were also used in the reduction of [DDFG24, PR24] for the full information setting. The key difference here is that we need to deal with the stochasticity of the combinatorial bandit setting. In our framework, the master samples an action $M_t \sim \hat{p}_t$ and observes a bandit reward r_t (Step 5). In contrast to the full information setting where the reward feedback is deterministic and given for all available actions, in our setting it is not straightforward what reward feedback the ScaleLearners should utilize. Following a well-established technique from combinatorial bandits (e.g., [CBL12, DKH07]), the master computes the reward estimate $r_t(\mathbb{E}_{\hat{p}_t}[MM^T])^+ M_t$ (Step 6), which is unbiased w.r.t. to its own policy $\hat{p}_t$. Then, the master broadcasts this estimate to the ScaleLearners of the ensemble (Step 7). Notably, the above stand in stark contrast to the approach in the bandit algorithm of [DDFG24], where the reward estimator is constructed based on the policy of the corresponding ScaleLearner.

Most importantly, our algorithmic framework entails an *idiomatic bandit learning protocol*: In contrast to the standard online learning protocol, ScaleLearners do not actually play any action, they only have a policy distribution over actions at each time-step. Each ScaleLearner updates its policy using the received reward estimators, coming from the master, which are *unbiased* w.r.t. the master's policy but *biased* w.r.t. the ScaleLearner's policy. This protocol allows us to derive a decomposition of the master's swap regret into the external regrets of the lazy learners through the following lemma.

Lemma 3.2 (Swap Regret Decomposition under Bandit Feedback)**.** *For any $\phi \in \Phi$, the swap regret of Algorithm 1 can be bounded as follows:*

$$\mathrm{SwapReg}_T(\phi) \leq \frac{1}{K} \sum_{k=1}^{K-1} \sum_{l=1}^{\lceil \frac{T}{H^k} \rceil} \mathrm{Reg}(\textsc{Lazy-CombAlg}_{k,l}) + \frac{T}{K}.$$

Remark 3.3. The role of the restarting period, H^k, is pivotal for the analysis in order to decompose

Algorithm 2 LAZY-COMBCP$_{k,l}$

Compute a 2-*approximate barycentric spanner*, S, of $\mathcal{A}$, and let $\mu = \frac{1}{d}\mathbb{1}\{M \in S\}$

Initialize $q_{k,l,1} = [1/d, ..., 1/d] \in \Delta^d$, $\gamma = H^{-1/3}$, $\eta = \frac{1}{d^3\sqrt{m}H^{k-1/3}}$

$$H' = \begin{cases} H & \text{if } l \le \frac{T}{H^k} \quad /^*\text{Play for } H \text{ meta-days,} \\ \left\lceil \frac{T - \lfloor T/H^k \rfloor H^k}{H^{k-1}} \right\rceil & \text{otherwise} \quad \text{or until the time limit } T \text{ is reached}^*/ \end{cases}$$

for $h = 1, 2, \ldots, H$ **do**

  Carathéodory Decomposition: $mq_{k,l,h} = \sum_{M \in \mathcal{M}} \widetilde{p}_{k,l,h}(M) M$

  Mixing: Let $p_{k,l,h} = (1-\gamma)\widetilde{p}_{k,l,h} + \gamma\mu$

  Meta-day: Fix $p_{k,l,h}$ for H^{k-1} days and aggregate the rewards of these days

$$\widetilde{X}_{k,l,h}(i) = \sum_{\tau=(\ell-1)H^k+(h-1)H^{k-1}+1}^{\min((\ell-1)H^k+hH^{k-1},T)} \widetilde{R}_\tau(i), \quad \forall i \in [d]$$

  Update: $\widetilde{q}_{k,l,h+1}(i) = q_{k,l,h}(i)\exp(\eta\widetilde{X}_{k,l,h}(h)),\ \forall i \in [d]$

  Projection: $q_{k,l,h+1} = \arg\min_{q \in \mathcal{P}} \mathrm{KL}(q, \widetilde{q}_{k,l,h+1})$

the swap regret of Algorithm 1 into the individual external regrets of each learner. For the interested reader, we provide an intuitive proof sketch illustration of Lemma 3.2 in Fig. 1 (Appendix C).

Remark 3.4. It is worth noting that beyond the laziness part, a key challenge in bounding the external regret of LAZY-COMBALG$_{k,l}$ is that the stochasticity of the expectation in the external regret is under the law of the master policy $\hat{p}_t$, rather than the LAZY-COMBALG$_{k,l}$'s policy $p_{k,t}$ as would be the standard. We discuss the above in more detail in Section 3.2.

Based on Lemma 3.2, our ultimate goal will be to design LAZY-COMBALG such that the external regret terms appearing in Lemma 3.2 are vanishing, thereby achieving a no-swap-regret guarantee for Algorithm 1. A key design goal is also to ensure that the per-iteration complexity of the overall framework scales polynomially with respect to d and m.

3.2 Instantiating LAZY-COMBALG

In this section, we present our no-external-regret combinatorial bandit algorithm, namely LAZY-COMBCP (see Algorithm 2), which can efficiently implement the LAZY-COMBALG learner in Algorithm 1. We denote the corresponding overall framework (i.e., Algorithm 1 combined with Algorithm 2) by **SWAP-COMBCP**.

Before discussing the LAZY-COMBCP algorithm, we will introduce the following useful notation. Let $p_{k,l,h}$ be the policy of LAZY-COMBCP$_{k,l}$ at the h-th meta-day of its interval of play, that is from $t = (\ell-1)H^k + (h-1)H^{k-1} + 1$ to $t = \min((\ell-1)H^k + hH^{k-1}, T)$. Moreover, let $M^*_{k,l} \in \arg\max_{M \in \mathcal{A}} \sum_{h=1}^{H} X_{k,l,h} \cdot M$; i.e. $M^*_{k,l}$ is an optimal action in hindsight for LAZY-COMBCP$_{k,l}$, within the interval l, for H meta-days and bandit feedback $X_{k,l,h}$ at meta-day h. Due to space constraints, when the indices k, l are fixed, that is when we analyze the external regret of LAZY-COMBCP$_{k,l}$ learner, we sometimes abuse the notation, and we write $\widetilde{p}_h = \widetilde{p}_{k,l,h}$, $\widetilde{X}_h = \widetilde{X}_{k,l,h}$, $M_\tau = M_t$, $\Sigma_\tau = \Sigma_t$ and $\mathbb{E}_{k,l,h}[\cdot] = \mathbb{E}_t[\cdot] = \mathbb{E}[\cdot|\mathcal{F}_t]$ where $t = (\ell-1)H^k + (h-1)H^{k-1} + \tau$.

Now, we are ready to present Lazy-CombBCP$_{k,l}$ (Algorithm 2). We adopt barycentric spanners (**B**), Carathéodory decomposition (**C**) and projection (**P**) techniques as key ingredients in our algorithm. Due to space constraints, background for the notion of barycentric spanners can be found in Appendix F.1. Algorithm 2 assumes that $\|M\|_1 = m$, for any $M \in \mathcal{A}$ (see also the discussion in Section 4.2 regarding the more general case where $\|M\|_1 \leq m$).

As suggested by our algorithmic framework (Algorithm 1), Lazy-CombBCP$_{k,l}$ operates lazily over H' meta-days, where H' is equal to H in general, unless ScaleLearner$_k$ is performing its final restart and H^k does not perfectly divide T (Step 3). Each full meta-day consists of H^{k-1} consecutive days. The learner performs updates on the aggregated reward estimates $\widetilde{X}$ (which have been broadcasted from the master) only at the end of each meta-day, staying lazy in between.

More specifically, our algorithm operates in the coordinate space $\mathbb{R}^d$ as follows: Let $\mathcal{M}$ be the convex hull of $\mathcal{A}$, and let $\mathcal{P}$ be the scaled convex hull of $\mathcal{A}$; that is, $\mathcal{P}$ is the convex hull of the actions $M \in \mathcal{A}$ divided by m. The algorithm uses a distribution $q \in \Delta^d$, which is updated (Step 8) based on the aggregated reward estimates (Step 7). In order to guarantee that the updated distribution is in $\mathcal{P}$, we project it onto $\mathcal{P}$ using the KL divergence (Step 9). Based on the definition of $\mathcal{P}$, it holds that $mq \in \mathcal{M}$, and thus mq can be decomposed (Step 5) into an efficient distribution, $\widetilde{p} \in \Delta^{|\mathcal{A}|}$, with small support (of at most d actions). To efficiently explore all d coordinates, the learner's policy is a mixture of $\widetilde{p}$ with an exploration policy μ, which is the uniform distribution over the elements of an approximate barycentric spanner over $\mathcal{A}$ (Step 6).

As the next lemma shows, the use of the barycentric spanner allows us to lower bound the minimum eigenvalue of Σ_τ, which is of great importance in order to control the boundedness and the variance of the aggregated reward estimates $\widetilde{X}$.

Lemma 3.5. *Let $\lambda_{min}(\mu)$ be the minimum eigenvalue of the co-occurrence matrix, Σ_τ, under the law of the barycentric spanner exploration policy μ. It holds that $\lambda_{min}(\mu) \geq \frac{1}{4d^3}$.*

Remark 3.6 (Efficient Sampling)**.** The use of the barycentric spanner, along with the decomposition step, is crucial for efficient sampling (Step 4 in Algorithm 1). Both the exploration policy μ and the policy $\tilde{p}_{k,l,h}$ have small support of size at most $\Theta(d)$, thus allowing efficient sampling from the ScaleLearner's policy $p_{k,t}$, and subsequently from the master learner's policy $\hat{p}_t$ (see Algorithm 1, Step 4).

Remark 3.7 (Mixing step)**.** Our mixing step in Algorithm 2 fixes an issue of CombExp [CTMSP+15], which first decomposes the mixture of q and the distribution over coordinates induced by the uniform distribution over $\mathcal{A}$. In particular, in settings of interest where the coordinates are unevenly represented in $\mathcal{A}$ (e.g., paths), the exploration of [CTMSP+15] may require a number of steps exponential in d to compete with the optimal action in hindsight. For a more detailed discussion, we refer the reader to Appendix F.3.

Next, we show two important properties of the aggregated reward estimate $\widetilde{X}$. As we show in the following lemma, the aggregated reward estimate is bounded and remains unbiased, importantly, with respect to the master's policy, since the expectation's randomness stems from the master's policy rather than the policy of Lazy-CombBCP$_{k,l}$.

Lemma 3.8. *For all $k \in \{1, ..., K-1\}$, $l \in \left[\left\lceil \frac{T}{H^k} \right\rceil\right]$ and $h \in [H]$, Lazy-CombBCP$_{k,l}$ satisfies the following properties:*

1. *(Unbiasedness):* $\mathbb{E}[q_h \cdot \widetilde{X}_h] = \mathbb{E}[q_h \cdot X_h]$

2. (Boundedness): $\|\widetilde{X}_h\|_2 \leq \frac{4H^{k-1}d^3\sqrt{m}}{\gamma}$

In the following theorem, we show that Algorithm 2 satisfies the no-external-regret property.

Theorem 3.9 (No-External-Regret)**.** *The external regret of* Lazy-ComBCP$_{k,l}$ *is at most* $3H^{k-1}H^{2/3}d^3m^{3/2}\log d$.

We note that the above bound implies no regret, because the H^{k-1} factor is the maximum reward per meta-day of the Lazy-ComBCP$_{k,l}$ learner.

3.3 Proof Sketch of Theorem 3.9

For simplicity, we assume that $H' = H$. In the case where $H' < H$, the derived upper bound still holds, since the actual horizon of play is smaller. Let $p^*_{k,l}$ be an optimal deterministic policy of the adversary, with $p^*_{k,l}(M^*_{k,l}) = 1$, of the learner $k \in [K]$ for the interval indexed by $l \in \left[\left\lceil \frac{T}{H^k} \right\rceil\right]$. First, we derive the following proposition.

Proposition 3.10 (Decomposition of optimal deterministic policy)**.** *Let* $q^*_{k,l} = \frac{M^*_{k,l}}{m}$*. It holds that* $mq^*_{k,l} = \sum_{M\in\mathcal{A}} p^*_{k,l}(M)M$.

Using the unbiasedness of the estimator (Lemma 3.8) and Proposition 3.10, we obtain the following proposition, which at first glance provides an upper bound on the external regret.

Proposition 3.11. *For all* $k \in \{1,\dots,K-1\}$ *and* $l \in \left[\left\lceil \frac{T}{H^k} \right\rceil\right]$*, the external regret of* Lazy-ComBCP$_{k,l}$ *can be bounded as follows:*

$$\mathrm{Reg}_H \leq m\mathbb{E}\left[\sum_{h=1}^{H} q^*_{k,l}\cdot\widetilde{X}_h - \sum_{h=1}^{H} q_h\cdot\widetilde{X}_h\right] + \gamma H^{k-1}.$$

Now, we need to bound the first term of the above expression. Using Lemma 3.8, we ensure that $\eta\|\widetilde{X}_h\|_\infty \leq 1$, so that Lazy-ComBCP satisfies the following Online Mirror Descent lemma.

Lemma 3.12 (Online Mirror Descent on $\mathcal{M}$)**.** *If* $\eta\|\widetilde{X}_h\|_\infty \leq 1$*, it holds that*

$$\mathbb{E}\left[\sum_{h=1}^{H}\left(q^*_{k,l} - q_h\right)\cdot\widetilde{X}_h\right] \leq \eta\mathbb{E}\left[\sum_{h=1}^{H} q_h\cdot\widetilde{X}_h^2\right] + \frac{\log d}{\eta},$$

where $\widetilde{X}_h^2$ *is the coordinate-wise square of* $\widetilde{X}_h$.

By decomposing q_h on the variance term of the above expression, we obtain the following:

$$\mathbb{E}_{k,l,h}\left[\sum_{M\in\mathcal{A}} q_h\cdot\widetilde{X}_h^2\right] = \frac{1}{m}\mathbb{E}_{k,l,h}\left[\sum_{M\in\mathcal{A}}\widetilde{p}_h(M)\left(\sum_{\tau=1}^{H^{k-1}}\widetilde{R}_\tau\right)^2\cdot M\right] \tag{4}$$

The challenge here is that the above variance term contains the squared sum of the reward estimates (which are due to the laziness of the algorithm), all of which are sampled by the *master's policy* $\widehat{p}_{k,l,h,\tau}$, and not by the ScaleLearner's policy $\widetilde{p}_h$, the external regret of which we aim to bound. In contrast to the analysis of no-external-regret combinatorial bandit algorithms (e.g., [DKH07, BDH+08, CBL12, CTMSP+15]), where bounding the variance involves controlling $\mathbb{E}_{k,l,h}\left[M_\tau^T\Sigma_\tau^+M_\tau\right]$, it turns out that we need to bound $\mathbb{E}_{k,l,h}\left[M_\tau^T\Sigma_\tau^{+2}M_\tau\right]$ instead. The following lemma provides a bound for these expectations.

Lemma 3.13. *For all $k \in \{1, ..., K-1\}$, $l \in \left[\left\lceil \frac{T}{H^k} \right\rceil\right]$ and $h \in [H]$, it holds that:*

$$\mathbb{E}_{k,l,h}\left[M_\tau^T \Sigma_\tau^{+2} M_\tau\right] \leq d\lambda_{min}^{-1}(\Sigma_\tau),$$

where $\lambda_{min}(\Sigma_\tau)$ is the minimum nonzero eigenvalue of $\Sigma_\tau = \mathbb{E}_{\hat{p}_{k,l,h,\tau}}[MM^T]$.

Now, using Lemmas 3.13 and 3.5 and a careful analysis, we derive the following final lemma, which concludes the proof.

Lemma 3.14 (Bounded variance)**.** *For all $k \in \{1, ..., K-1\}$, $l \in \left[\left\lceil \frac{T}{H^k} \right\rceil\right]$ and $h \in [H]$, Lazy-CombCP$_{k,l}$ satisfies the following:*

$$\mathbb{E}[q_h \cdot \widetilde{X}_h^2] \leq H^{k-1} \sum_{\tau=1}^{H^{k-1}} \mathbb{E}_{k,l,h}\left[M_\tau^T \Sigma_\tau^{-2} M_\tau\right] \leq \frac{4H^{2k-2}d^4}{\gamma}$$

4 Main Result

4.1 Swap Regret Bounds

Combining the no-external-regret guarantees of Algorithm 2 (Theorem 3.9) with Lemma 3.2, we derive our main result, which provides the no-swap-regret guarantees of Swap-ComBCP.

Theorem 4.1 (No-Swap-Regret)**.** *Setting $H = 27\lfloor \log(T) \rfloor^3 d^9 m^{9/2} \log^3 d$ and $K = \lfloor \log_H(T) \rfloor$, Swap-CombCP satisfies the following swap-regret guarantee:*

$$\text{SwapReg} \leq \frac{45 \log(d \log T) T}{\log T}.$$

To provide anytime guarantees, typically required for online learning, we next show that the Doubling Trick [ACBG02, BK18] can be applied to our algorithm.

Lemma 4.2 (Anytime Guarantees)**.** *Assuming that, for given T, Algorithm 1 achieves*

$$\text{SwapReg}_T(\phi) \leq \mathcal{O}\left(\frac{T \log(d \log T)}{\log T}\right),$$

then it achieves the above bound with anytime guarantees, if implemented with the Doubling Trick.

By applying the lower bound of [DFG+24] in combinatorial bandits (see also Appendix B), we obtain the following corollary, which states that in the regime of interest, that is $T < \exp(d)$, our swap regret upper bound is tight, in the sense that there exists no T^p-no-swap-regret scheme, where p is a constant less than 1.

Corollary 4.3 (Tightness)**.** *For $T \leq \exp(\Omega(d^{1/14}))$ there exists a combinatorial bandit setting with action set $\mathcal{A} \subset \{0,1\}^d$ and an oblivious adversary running for T rounds against whom any learner suffers swap regret at least $\Omega\left(T/\log^6 T\right)$.*

4.2 Per-Iteration Complexity in Well-Studied Combinatorial Settings

We present some important combinatorial bandit settings where Swap-ComBCP is applicable (i.e., the L_1-norm of the action vectors equals a fixed value m) and efficiently implementable (i.e., we can efficiently perform decomposition, projection and approximate barycentric spanner calculation). The settings we consider are *m-sets*, *Spanning Trees and Graph Matroid Bandits (k-forests), Paths, Permutations* and *Truncated Permutations*. For all the above settings, except for paths, efficient decomposition and projection algorithms are proposed in [SHK+12]. For paths in DAGs, the path polytope has a succinct description which allows efficient decomposition and projection with interior point methods (see [BV04], page 545). A careful reader might observe that paths do not necessarily satisfy the condition that $\|M\|_1$ can vary across $\mathcal{A}$. Fortunately, it turns out that one can always transform the DAG, so that this condition is satisfied (see [GLLO07]). Moreover, for all the above settings, we can efficiently compute an approximate barycentric spanner as there exists an efficient linear minimization oracle over $\mathcal{A}$ (see Proposition F.2). More details are also provided in Appendix H.1.

Regarding the more general setting, where $\|M\|_1$ varies across actions $M \in \mathcal{A}$, we provide an alternative implementation for Lazy-CombAlg, namely Lazy-ComBand, which is a lazy version of the well-known ComBand algorithm [CBL12]. Due to space constraints, we only include this algorithm in Appendix G. Both Swap-ComBCP and Swap-ComBand achieve the same asymptotic swap regret guarantees. Although Swap-ComBand is applicable to the more general setting, the applications, where efficient implementations of it are known, are more limited compared to Swap-ComBCP. In particular, the main applications of Swap-ComBand are paths, as well as problems that can be modelled as paths, such as m-sets, both of which are also covered by Swap-ComBCP.

5 Conclusion

Our paper focused on the problem of no-swap-regret learning in combinatorial bandits, i.e., structured bandit settings where the number of actions is exponentially large (in the settings description). Importantly, we provided the first online algorithms for combinatorial bandits which achieve vanishing swap regret and per-iteration complexity that scale polylogarithmically in the number of actions. One interesting open question is whether we can obtain similar results in combinatorial bandits for realized swap regret with high probability guarantees.

Acknowledgments

Most of this work was done while all authors were visiting Archimedes Research Unit, Athens, Greece. Ioannis Panageas was supported by NSF grant CCF-2454115. This work was also supported by the French National Research Agency (ANR) in the framework of the PEPR IA FOUNDRY project (ANR-23-PEIA-0003) and project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

References

[ABL14] Jean-Yves Audibert, Sébastien Bubeck, and Gábor Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39(1):31–45, 2014.

[ACBFS02] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The non-stochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[ACBG02] Peter Auer, Nicolo Cesa-Bianchi, and Claudio Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75, 2002.

[AK04] Baruch Awerbuch and Robert D Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 45–53, 2004.

[AK08] Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, 2008.

[BCBK12] Sébastien Bubeck, Nicolo Cesa-Bianchi, and Sham M Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Conference on Learning Theory*, pages 41–1. JMLR Workshop and Conference Proceedings, 2012.

[BDH+08] Peter Bartlett, Varsha Dani, Thomas Hayes, Sham Kakade, Alexander Rakhlin, and Ambuj Tewari. High-probability regret bounds for bandit online linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory-COLT 2008*, pages 335–342. Omnipress, 2008.

[BEDL06] Avrim Blum, Eyal Even-Dar, and Katrina Ligett. Routing without regret: on convergence to Nash equilibria of regret-minimizing in routing games. In *PODC '06: Proceedings of the 25th annual ACM SIGACT-SIGOPS symposium on Principles of Distributed Computing*, pages 45–52, 2006.

[BK18] Lilian Besson and Emilie Kaufmann. What doubling tricks can and can't do for multi-armed bandits. *arXiv preprint arXiv:1803.06971*, 2018.

[BM07] Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8(6), 2007.

[BP16] Gábor Braun and Sebastian Pokutta. An efficient high-probability algorithm for linear bandits. *arXiv preprint arXiv:1610.02072*, 2016.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[CBL03] Nicolo Cesa-Bianchi and Gábor Lugosi. Potential-based algorithms in on-line prediction and game theory. *Machine Learning*, 51:239–261, 2003.

[CBL12] Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

[CTMSP+15] Richard Combes, Mohammad Sadegh Talebi Mazraeh Shahi, Alexandre Proutiere, et al. Combinatorial bandits revisited. *Advances in neural information processing systems*, 28, 2015.

[DDFG24] Yuval Dagan, Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich. From external to swap regret 2.0: An efficient reduction for large action spaces. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1216–1222, 2024.

[DFG+24] Constantinos Daskalakis, Gabriele Farina, Noah Golowich, Tuomas Sandholm, and Brian Hu Zhang. A lower bound on swap regret in extensive-form games. *arXiv preprint arXiv:2406.13116*, 2024.

[DKH07] Varsha Dani, Sham M Kakade, and Thomas Hayes. The price of bandit information for online optimization. *Advances in Neural Information Processing Systems*, 20, 2007.

[FV97] Dean Foster and Rakesh V. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21(1):40–55, October 1997.

[FV99] Dean P Foster and Rakesh Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 29(1-2):7–35, 1999.

[GLLO07] András György, Tamás Linder, Gábor Lugosi, and György Ottucsák. The on-line shortest path problem under partial monitoring. *Journal of Machine Learning Research*, 8(10), 2007.

[HK16] Elad Hazan and Zohar Karnin. Volumetric spanners: an efficient exploration basis for learning. *Journal of Machine Learning Research*, 2016.

[Ito20] Shinji Ito. A tight lower bound and efficient reduction for swap regret. *Advances in Neural Information Processing Systems*, 33:18550–18559, 2020.

[KPF+25] Andreas Kontogiannis, Vasilis Pollatos, Gabriele Farina, Panayotis Mertikopoulos, and Ioannis Panageas. Efficient kernelized learning in polyhedral games beyond full information: From colonel blotto to congestion games. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[KRS10] Satyen Kale, Lev Reyzin, and Robert E Schapire. Non-stochastic bandit slate problems. *Advances in Neural Information Processing Systems*, 23, 2010.

[KV05] Adam Tauman Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, October 2005.

[KWA+14] Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson. Matroid bandits: Fast combinatorial optimization with learning. *arXiv preprint arXiv:1403.5045*, 2014.

[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, Cambridge, UK, 2020.

[MY14] Mehryar Mohri and Scott Yang. Conditional swap regret and conditional correlated equilibrium. *Advances in Neural Information Processing Systems*, 27, 2014.

[MY17] Mehryar Mohri and Scott Yang. Online learning with transductive regret. *Advances in Neural Information Processing Systems*, 30, 2017.

[PR24] Binghui Peng and Aviad Rubinstein. Fast swap regret minimization and applications to approximate correlated equilibria. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1223–1234, 2024.

[Sha24] Dravyansh Sharma. No internal regret with non-convex loss functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14919–14927, 2024.

[SHK+12] Daiki Suehiro, Kohei Hatano, Shuji Kijima, Eiji Takimoto, and Kiyohito Nagano. Online prediction under submodular constraints. In *Algorithmic Learning Theory: 23rd International Conference, ALT 2012, Lyon, France, October 29-31, 2012. Proceedings 23*, pages 260–274. Springer, 2012.

[SL05] Gilles Stoltz and Gábor Lugosi. Internal regret in on-line portfolio selection. *Machine Learning*, 59:125–159, 2005.

[Sto05] Gilles Stoltz. *Incomplete information and internal regret in prediction of individual sequences*. PhD thesis, Université Paris Sud-Paris XI, 2005.

[TW03] Eiji Takimoto and Manfred K Warmuth. Path kernels and multiplicative updates. *The Journal of Machine Learning Research*, 4:773–818, 2003.

[VZ13] Yannick Viossat and Andriy Zapechelnyuk. No-regret dynamics and fictitious play. *Journal of Economic Theory*, 148(2):825–842, March 2013.

[WAEK15] Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Branislav Kveton. Efficient learning in large-scale combinatorial semi-bandits. In *ICML '15: Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Contents

1 Introduction **2**

2 Preliminaries **5**

3 Swap-ComBCP: A No-Swap-Regret Combinatorial Bandit Algorithm **5**
- 3.1 The algorithmic framework . . . 6
- 3.2 Instantiating Lazy-CombAlg . . . 8
- 3.3 Proof Sketch of Theorem 3.9 . . . 10

4 Main Result **11**
- 4.1 Swap Regret Bounds . . . 11
- 4.2 Per-Iteration Complexity in Well-Studied Combinatorial Settings . . . 12

5 Conclusion **12**

A Further Related Work **17**

B Lower Bound **17**

C Proof of Theorem 3.2 **17**

D Anytime Guarantees of the Multi-Scale Algorithm **21**

E Proof of Lemma 3.13 **23**

F Swap-ComBCP **24**
- F.1 Barycentric Spanners . . . 24
- F.2 Analysis of Swap-ComBCP . . . 25
- F.3 Issue with the exploration scheme of CombExp . . . 32

G Swap-ComBand **34**

H Per-Iteration Complexity in Well-Studied Combinatorial Settings **39**
- H.1 Applications of Swap-ComBCP . . . 39
- H.2 Applications of Swap-ComBand . . . 40

A Further Related Work

Other Strong Notions of Regret Another well-studied strong notion of regret is the internal regret, which was initially introduced in [FV99] in the context of calibrated forecasting. Internal regret is a notion of regret, weaker than swap and stronger than external regret, which measures the change in performance by substituting every occurrence of a given action a by an alternative action a'. Several algorithms minimizing internal regret have been developed, e.g., [SL05, BM07, CBL03, Sha24] to name a few. Moreover, Mohri and Yang [MY14, MY17] have introduced alternative strong notions of regret, namely conditional swap regret and transductive regret, which account for all possible modifications of a learner's action sequence that depend on some fixed bounded history.

Further related work on adversarial combinatorial bandits. [BCBK12] proposed the use of John's ellipsoid method for optimal experiment design, achieving a regret bound of $\widetilde{\mathcal{O}}(\sqrt{mdT})$ for any finite action set, against a lower bound of $\widetilde{\mathcal{O}}(\sqrt{dT})$ [ABL14]; in general however, John's ellipsoid method requires time polynomial in N, so it is not always efficient. An alternative – and, in many cases, more efficient – exploration method is via approximate barycentric spanners [AK04, AK08, DKH07, BDH+08, BP16], or the more sophisticated approximate volumetric spanners [HK16].

B Lower Bound

In [DFG+24] a swap regret lower bound is derived for adversarial online learning in the context of extensive-form games (EFGs) with full information. The authors study EFGs from the perspective of a single player as an online learning setting and show that achieving average swap regret ϵ against an oblivious adversary in EFGs with d decision nodes requires a number of rounds T that is exponential either in d or ϵ. Moreover, as shown in [DFG+24] the actions of a single player in EFGs can be written as binary vectors of dimensionality d and the rewards are linear in the action vectors. Therefore, the above setting is captured by the combinatorial setting of our paper, and the above lower bound is applicable to combinatorial bandits. In particular, Theorem 4.1 in [DFG+24] states that there exists an action set $\mathcal{A} \subset \{0,1\}^d$, such that for any $\epsilon > 0$ and $T \leq \exp(\Omega(\min\{d^{1/14}, \epsilon^{-1/6}\}))$, there exists an oblivious adversary running for T rounds against whom no learner can achieve average swap regret less than ϵ. Since the lower bound holds for full information feedback, it also carries over to the bandit case.

In the following corollary we restate the result for the regime of interest, that is $T << \exp(d)$, in terms of the regret as a function of T.

Corollary. *For $T \leq \exp(\Omega(d^{1/14}))$ there exists a combinatorial bandit setting with action set $\mathcal{A} \subset \{0,1\}^d$ and an oblivious adversary running for T rounds against whom any learner suffers swap regret at least $\Omega\left(\frac{T}{\log^6 T}\right)$.*

Therefore, in the regime of interest (that is, $T << \exp(d)$), our swap regret upper bound is tight, in the sense that there exists no T^p-no-swap-regret scheme, where p is a constant less than 1.

C Proof of Theorem 3.2

We first prove Proposition 3.1 which describes the useful property of laziness in swap regret.

Proposition 3.1 (Swap Regret is upper bounded by External Regret for time invariant policies): *For any policy $p \in \Delta(\mathcal{A})$ that is fixed in an interval $t = 1...T$, any sequence of rewards R_t and any swap function $\phi : \mathcal{A} \to \mathcal{A}$ it holds that*

$$\sum_{t=1}^{T} \sum_{M \in \mathcal{A}} p(M) R_t \cdot \phi(M) \leq \max_{M \in \mathcal{A}} \left\{ \sum_{t=1}^{T} R_t \cdot M \right\}$$

Proof.

$$\begin{aligned}
\sum_{t=1}^{T} \sum_{M \in \mathcal{A}} p(M) R_t \cdot \phi(M) &= \sum_{t=1}^{T} R_t \cdot \sum_{M \in \mathcal{A}} p(M) \phi(M) \\
&= \left(\sum_{t=1}^{T} R_t \right) \cdot \left(\sum_{M \in \mathcal{A}} p(M) \phi(M) \right) \\
&= \left(\sum_{t=1}^{T} R_t \right) \cdot V, \qquad \text{for some } V \in Conv(\mathcal{A}) \\
&\leq \max_{V \in Conv(\mathcal{A})} \left\{ \left(\sum_{t=1}^{T} R_t \right) \cdot V \right\} \\
&\leq \max_{M \in \mathcal{A}} \left\{ \sum_{t=1}^{T} R_t \cdot M \right\}
\end{aligned}$$

In the last step we used the fact that linear functions achieve their maximum at extreme points of the feasible region. In the case of the convex hull, the extreme points are precisely the original set of vectors. □

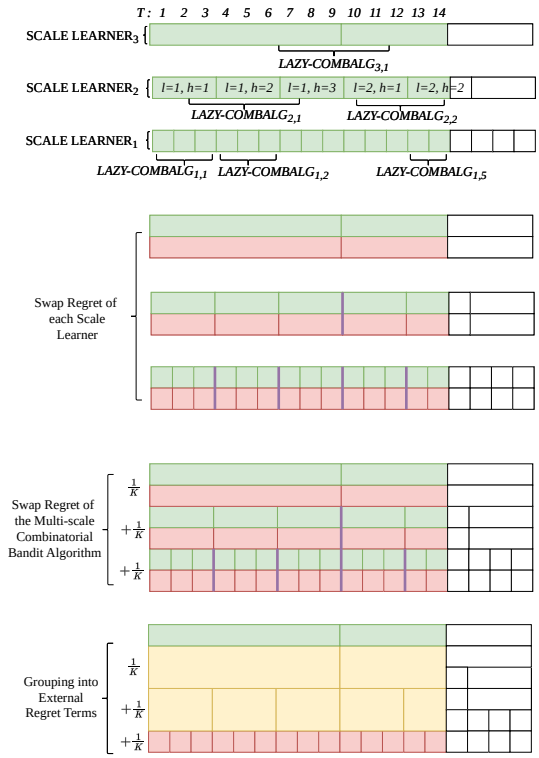


Figure 1: Swap Regret Analysis for the Multi-scale Combinatorial Bandit Algorithm for $T = 14, K = 3$ and $H = 3$. Purple lines represent restarts of the ScaleLearners. Green terms correspond to expected cumulative rewards collected by the learners, while red terms correspond to expected cumulative rewards of a swap policy that is optimal in hindsight. In each interval, where a Lazy-CombAlg's policy remains fixed, the optimal swap policy is playing a fixed pure action (see Proposition 3.1). Thus, red terms actually correspond to expected cumulative rewards of fixed actions that are optimal in hindsight. Orange terms represent the external regrets of the lazy no-external regret algorithms.

Proof of Theorem 3.2

Proof. Due to space constraints, here we use the notation SR to denote SwapReg.

$$\mathrm{SR}_T(\phi) = \mathbb{E}\left[\sum_{t=1} R_t \cdot (\phi(M_t) - M_t)\right] \tag{5}$$

$$= \mathbb{E}\left[\sum_{t=1} \mathbb{E}_{\widehat{p}_t}\left[R_t \cdot (\phi(M_t) - M_t) \mid \mathcal{F}_t\right]\right] \tag{6}$$

$$= \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\sum_{t=1}^{T}\sum_{M\in\mathcal{A}} p_{k,t}(M) R_t \cdot (\phi(M) - M)\right] \tag{7}$$

$$\leq \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\sum_{l=1}^{\left\lfloor\frac{T}{H^k}\right\rfloor}\sum_{h=1}^{H}\sum_{M\in\mathcal{A}} p_{k,l,h}(M)\sum_{\tau=1}^{H^{k-1}} R_{k,l,h,\tau} \cdot (\phi(M) - M)\right] +$$

$$+\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\sum_{h=1}^{H'_k}\sum_{M\in\mathcal{A}}p_{k,l,h}(M)\sum_{\tau=1}^{D_{h,k}}R_{k,l,h,\tau}\cdot(\phi(M)-M)\Big|_{l=\left\lfloor\frac{T}{H^k}\right\rfloor+1}\right] \tag{8}$$

$$\leq\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\sum_{l=1}^{\left\lfloor\frac{T}{H^k}\right\rfloor}\sum_{h=1}^{H}\left[\max_{M\in\mathcal{A}}\left\{\sum_{\tau=1}^{H^{k-1}}R_{k,l,h,\tau}\cdot M\right\}-\sum_{M\in\mathcal{A}}p_{k,l,h}(M)\sum_{\tau=1}^{H^{k-1}}R_{k,l,h,\tau}\cdot M\right]\right]+$$

$$+\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\sum_{h=1}^{H'_k}\left[\max_{M\in\mathcal{A}}\left\{\sum_{\tau=1}^{D_{h,k}}R_{k,l,h,\tau}\cdot M\right\}-\sum_{M\in\mathcal{A}}p_{k,l,h}(M)\sum_{\tau=1}^{D_{h,k}}R_{k,l,h,\tau}\cdot M\right]\Big|_{l=\left\lfloor\frac{T}{H^k}\right\rfloor+1}\right] \tag{9}$$

$$=\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\sum_{l=1}^{\left\lfloor\frac{T}{H^k}\right\rfloor}\sum_{h=1}^{H}\left[\max_{M\in\mathcal{A}}\left\{X_{k,l,h}\cdot M\right\}-\sum_{M\in\mathcal{A}}p_{k,l,h}(M)X_{k,l,h}\cdot M\right]\right]+$$

$$+\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\sum_{h=1}^{H'_k}\left[\max_{M\in\mathcal{A}}\left\{X_{k,l,h}\cdot M\right\}-\sum_{M\in\mathcal{A}}p_{k,l,h}(M)X_{k,l,h}\cdot M\right]\Big|_{l=\left\lfloor\frac{T}{H^k}\right\rfloor+1}\right] \tag{10}$$

$$=\mathbb{E}\left[\frac{1}{K}\sum_{k=2}^{K}\sum_{l=1}^{\left\lfloor\frac{T}{H^k}\right\rfloor}\sum_{h=1}^{H}\max_{M\in\mathcal{A}}\left\{X_{k,l,h}\cdot M\right\}+\frac{1}{K}\sum_{l=1}^{\left\lfloor\frac{T}{H}\right\rfloor}\sum_{h=1}^{H}\max_{M\in\mathcal{A}}\left\{X_{1,l,h}\cdot M\right\}\right]+$$

$$+\mathbb{E}\left[\frac{1}{K}\sum_{k=2}^{K}\sum_{h=1}^{H'_k}\left[\max_{M\in\mathcal{A}}\left\{X_{k,l,h}\cdot M\right\}\right]\Big|_{l=\left\lfloor\frac{T}{H^k}\right\rfloor+1}\right]+\mathbb{E}\left[\frac{1}{K}\sum_{h=1}^{H'_1}\left[\max_{M\in\mathcal{A}}\left\{X_{1,l,h}\cdot M\right\}\right]\Big|_{l=\left\lfloor\frac{T}{H}\right\rfloor+1}\right]-$$

$$-\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K-1}\sum_{l=1}^{\left\lfloor\frac{T}{H^k}\right\rfloor}\sum_{h=1}^{H}\sum_{M\in\mathcal{A}}p_{k,l,h}(M)X_{k,l,h}\cdot M+\frac{1}{K}\sum_{l=1}^{\left\lfloor\frac{T}{HK}\right\rfloor}\sum_{h=1}^{H}\sum_{M\in\mathcal{A}}p_{K,l,h}(M)X_{K,l,h}\cdot M\right]-$$

$$-\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K-1}\sum_{h=1}^{H'_k}\left[\sum_{M\in\mathcal{A}}p_{k,l,h}(M)X_{k,l,h}\cdot M\right]\Big|_{l=\left\lfloor\frac{T}{H^k}\right\rfloor+1}\right]-\mathbb{E}\left[\frac{1}{K}\sum_{h=1}^{H'_K}\left[\sum_{M\in\mathcal{A}}p_{K,l,h}(M)X_{K,l,h}\cdot M\right]\Big|_{l=\left\lfloor\frac{T}{HK}\right\rfloor+1}\right] \tag{11}$$

$$\leq\mathbb{E}\left[\frac{1}{K}\sum_{k=2}^{K}\sum_{l=1}^{\left\lfloor\frac{T}{H^k}\right\rfloor}\sum_{h=1}^{H}\max_{M\in\mathcal{A}}\left\{X_{k,l,h}\cdot M\right\}\right]+\mathbb{E}\left[\frac{1}{K}\sum_{k=2}^{K}\sum_{h=1}^{H'_k}\left[\max_{M\in\mathcal{A}}\left\{X_{k,l,h}\cdot M\right\}\right]\Big|_{l=\left\lfloor\frac{T}{H^k}\right\rfloor+1}\right]+\frac{T}{K}-$$

$$-\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K-1}\sum_{l=1}^{\left\lfloor\frac{T}{H^k}\right\rfloor}\sum_{h=1}^{H}\sum_{M\in\mathcal{A}}p_{k,l,h}(M)X_{k,l,h}\cdot M\right]-\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K-1}\sum_{h=1}^{H'_k}\left[\sum_{M\in\mathcal{A}}p_{k,l,h}(M)X_{k,l,h}\cdot M\right]\Big|_{l=\left\lfloor\frac{T}{H^k}\right\rfloor+1}\right] \tag{12}$$

$$=\frac{1}{K}\sum_{k=1}^{K-1}\sum_{l=1}^{\left\lceil\frac{T}{H^k}\right\rceil}\mathrm{Reg}(\textsc{Lazy-CombAlg}_{k,l})+\frac{T}{K} \tag{13}$$

In 6 we use the tower property, and in 7 we use the definition of the conditional expectation and the definition of the master's distribution $\widehat{p}_t$. When we write $R_{k,l,h,\tau}$ we mean R_t at $t=(l-1)H^k+(h-1)H^{k-1}+\tau$.

One crucial part is step 8, where for each $\textsc{ScaleLearner}_k$ we split T into two parts; the first part contains the segments in which the $\textsc{Lazy-CombAlg}$ instance fully runs the horizon, and one

final part in which the Lazy-CombAlg instance may not fully run the horizon, due to the fact that H^K may not perfectly divide T. In particular, we consider the following: (a) in the first part, for each ScaleLearner$_k$ we have $\left\lfloor \frac{T}{H^k} \right\rfloor$ segments of size $H \cdot H^{k-1}$ days, where the Lazy-CombAlg instance fully runs a horizon of H meta-days, (b) in the second part, for each learner, there remain H'_k meta-days, where $0 \le H'_k \le H$ (that is at most H^k days), where the Lazy-CombAlg instance may terminate its run before a horizon of H meta-days is completed. Moreover, in this second part the last meta day of each learner may last less than H^k days and we denote this duration by $D_{h,k}$.

In step 9 we use a key property of laziness that allows us to upper bound the performance of the reference swap function ϕ by the performance of the best fixed action in an interval where the learner has fixed policy, as we show n Proposition 3.1.

In step 10 we define

$$X_{k,l,h} := \begin{cases} \sum_{\tau=1}^{H^{k-1}} R_{k,l,h,\tau} & 1 \le k \le K, \quad 1 \le l \le \left\lfloor \frac{T}{H^k} \right\rfloor, \quad 1 \le h \le H \\ \sum_{\tau=1}^{D_{h,k}} R_{k,l,h,\tau} & 1 \le k \le K, \quad l = \left\lfloor \frac{T}{H^k} \right\rfloor + 1, \quad 1 \le h \le H'_k \end{cases}$$

In steps 11, 12 and 13 we group the terms corresponding to the cumulative reward of the optimal fixed action in each meta-day and the terms corresponding to the expected cumulative reward of the lazy learners so that the external regrets of the lazy learners are formed. For the swap regret upper bound we ignore the expected cumulative reward of the layer $k = K$, since its contribution only decreases the upper bound. Moreover, we isolate the cumulative reward of the optimal fixed actions in the meta-days at layer $k = 1$ (these meta-days are actually days since they have duration $H^{k-1} = 1$). These terms can not be paired into any external regret term and they increase the swap regret upper bound, but we can bound their total contribution by $\frac{T}{K}$, which is vanishing on K. This analysis can be visualized in Figure 1.

□

D Anytime Guarantees of the Multi-Scale Algorithm

Lemma D.1 (Lemma 4.2 restated)**.** *Assuming that Algorithm 1 achieves swap regret*

$$\mathrm{SwapReg}_T(\phi) \le \mathcal{O}\left(\frac{T \log(d \log T)}{\log T}\right),$$

Then, Algorithm 1 achieves the above swap regret with anytime guarantees, if it is implemented with the Doubling Trick.

Proof. We apply the Doubling Trick [ACBG02, BK18] to obtain anytime guarantees for Algorithm 1. For convenience, the analysis below ignores the terms that do not depend on T.

First, we split the sum of the regrets (induced by the Doubling Trick) into two parts:

$$S = \sum_{i=1}^{\log T} \frac{2^i \log(di)}{i} \le \log(d \log T) \sum_{i=1}^{\log T} \frac{2^i}{i} = \log(d \log T) \underbrace{\sum_{i=1}^{\lfloor \log T/2 \rfloor} \frac{2^i}{i}}_{\text{Part 1}} + \log(d \log T) \underbrace{\sum_{i=\lfloor \log T/2 \rfloor + 1}^{\log T} \frac{2^i}{i}}_{\text{Part 2}}.$$

- For $i \le \log T/2$, note that $2^i \le \sqrt{T}$, since $i \le \log T/2$ implies $2^i = \sqrt{2^{\log T}} = \sqrt{T}$. Therefore:

$$\sum_{i=1}^{\lfloor \log T/2 \rfloor} \frac{2^i}{i} \le \sqrt{T} \cdot \sum_{i=1}^{\lfloor \log T/2 \rfloor} \frac{1}{i}.$$

The summation $\sum_{i=1}^{n} \frac{1}{i}$ is the n-th harmonic number, H_n, which satisfies:

$$H_n \le \ln n + 1.$$

Substituting $n = \lfloor \log T/2 \rfloor$, we get:

$$\sum_{i=1}^{\lfloor \log T/2 \rfloor} \frac{2^i}{i} \le \sqrt{T} \cdot (\ln(\log T/2) + 1).$$

Thus, Part 1 is:

$$\mathcal{O}\left(\sqrt{T} \cdot \log \log T\right),$$

which is asymptotically smaller than $\mathcal{O}\left(\frac{T}{\log T}\right)$.

- For $i > \log T/2$, note that $i \ge \log T/2$ implies $\frac{1}{i} \le \frac{2}{\log T}$. Therefore:

$$\sum_{i=\lfloor \log T/2 \rfloor + 1}^{\log T} \frac{2^i}{i} \le \frac{2}{\log T} \cdot \sum_{i=\lfloor \log T/2 \rfloor + 1}^{\log T} 2^i.$$

The summation $\sum_{i=\lfloor \log T/2 \rfloor + 1}^{\log T} 2^i$ is a geometric sequence, which simplifies to:

$$\sum_{i=\lfloor \log T/2 \rfloor + 1}^{\log T} 2^i = 2^{\log T + 1} - 2^{\lfloor \log T/2 \rfloor + 1}.$$

Using $2^{\log T} = T$, this becomes:

$$\sum_{i=\lfloor \log T/2 \rfloor + 1}^{\log T} 2^i = 2T - 2 \cdot \sqrt{T} \le 2T.$$

Substituting this back, we have:

$$\sum_{i=\lfloor \log T/2 \rfloor + 1}^{\log T} \frac{2^i}{i} \le \frac{2}{\log T} \cdot 2T = \frac{4T}{\log T}.$$

Combining the above, we have that the total summation is the sum of Part 1 and Part 2. Since Part 1 is $\mathcal{O}\left(\sqrt{T} \cdot \log \log T\right)$ and Part 2 is $\mathcal{O}\left(\frac{T}{\log T}\right)$, the dominant term is from Part 2. Therefore:

$$S = \sum_{i=1}^{\log T} \frac{2^i}{i} = \mathcal{O}\left(\frac{T}{\log T}\right).$$

□

E Proof of Lemma 3.13

Lemma E.1 (Lemma 3.13 restated)**.** *For all $k \in \{1, ..., K-1\}$, $l \in \left[\left\lceil \frac{T}{H^k} \right\rceil\right]$ and $h \in [H]$, it holds that,*

$$\mathbb{E}_{k,l,h}\left[M^T_{k,l,h,\tau}\Sigma^{+2}_{k,l,h,\tau}M_{k,l,h,\tau}\right] \leq d\lambda^{-1}_{min}(\Sigma_{k,l,h,\tau}),$$

where $\lambda^{-1}_{min}(\Sigma_{k,l,h,\tau})$ is the minimum eigenvalue of the co-occurrence matrix $\Sigma_{k,l,h,\tau}$ under the law of $\hat{p}_{k,l,h,\tau}$.

Proof. Let $\lambda_i(\Sigma_{k,l,h,\tau})$ be the eigenvalue of $\Sigma_{k,l,h,\tau}$ corresponding to the eigenvector $u_i(\Sigma_{k,l,h,\tau})$. For brevity, we abuse notation and instead simply use $\lambda_{\tau,i}$ and $u_{\tau,i}$. Moreover, let $\lambda_{\tau,1}, ..., \lambda_{\tau,r}$ be the r nonzero eigenvalues with corresponding eigenvectors $u_{\tau,1}, ..., u_{\tau,r}$. Based on the above, we have the following:

$$\begin{aligned}
\mathbb{E}_{k,l,h}\left[M^T_{k,l,h,\tau}\Sigma^{+2}_{k,l,h,\tau}M_{k,l,h,\tau}\right] &= \mathbb{E}_{k,l,h}\left[\mathbb{E}_{k,l,h,\tau}\left[M^T_{k,l,h,\tau}\Sigma^{+2}_{k,l,h,\tau}M_{k,l,h,\tau}\right]\right] && (14)\\
&= \mathbb{E}_{k,l,h}\left[\sum_{M\in\mathcal{A}} \widehat{p}_{k,l,h,\tau}(M) M^T\Sigma^{+2}_{k,l,h,\tau}M\right] && (15)\\
&\leq \mathbb{E}_{k,l,h}\left[\sum_{M\in\mathcal{A}} \widehat{p}_{k,l,h,\tau}(M) \sum_{i\in[r]} \lambda^{-2}_{\tau,i}\cdot(u_{\tau,i}\cdot M)^2\right] && (16)\\
&= \mathbb{E}_{k,l,h}\left[\sum_{i\in[r]} \lambda^{-2}_{\tau,i} \sum_{M\in\mathcal{A}} \widehat{p}_{k,l,h,\tau}(M)(u_{\tau,i}\cdot M)^2\right] && (17)\\
&= \mathbb{E}_{k,l,h}\left[\sum_{i\in[r]} \lambda^{-1}_{\tau,i}\right] && (18)\\
&\leq d\lambda^{-1}_{\min}(\Sigma_{k,l,h,\tau}) && (19)
\end{aligned}$$

where in 14 we used the tower property, in 16 we used: (a) the inequality $x^TAx \leq \sum_{i\in[r]}\lambda_i(u_i\cdot x)$ for any vertex $x \in \mathbb{R}^d$ and any symmetric positive semi-definite matrix $A \in \mathbb{R}^{d\times d}$, and (b) the fact that the eigenvalues of the squared inverse matrix A^{+2} equal the squared inverse of the eigenvalues of A.

$$\lambda_{\tau,i} = u_i^T\Sigma_{k,l,h,\tau}u_i = \sum_{M\in\mathcal{A}} \widehat{p}_{k,l,h,\tau}(M)(u_i\cdot M)^2. \tag{20}$$

Moreover, in 19 we used the fact that $r \leq d$. □

F Swap-ComBCP

F.1 Barycentric Spanners

This section introduces the notion of barycentric spanners [AK04]. In the Algorithm 2, we leverage barycentric spanners to ensure adequate exploration of each coordinate $i \in [d]$ that allows us to guarantee low variance of the reward estimators.

Definition F.1 (C-approximate barycentric spanner)**.** A subset of vectors $S = \{b_1, \ldots, b_d\} \subseteq \mathcal{A}$ is said to be C-approximate barycentric spanner of $\mathcal{A}$, with $C > 1$, if, for all $M \in \mathcal{A}$, there exists $a \in \mathbb{R}^d$ such that

$$M = Ba \quad \text{and} \quad \|a\|_\infty \leq C.$$

where B is a matrix whose columns are the barycentric spanners $\{b_1, \ldots, b_d\}$.

The following proposition ensures that, if specific conditions hold, there exists an efficient algorithm for computing a C-approximate barycentric spanner.

Proposition F.2 ([AK08], Proposition 2.5)**.** *Suppose $\mathcal{A} \subseteq \mathbb{R}^d$ is a compact set not contained in any proper linear subspace. Given an oracle for optimizing linear functions over $\mathcal{A}$, for any $C > 1$ there exists an algorithm that computes a C-approximate barycentric spanner for $\mathcal{A}$ in polynomial time, using $\widetilde{\mathcal{O}}\left(d^2\right)$ calls to the optimization oracle.*

In order to control the minimum eigenvalue of the co-occurrence matrix under the law of μ (uniform distribution over the barycentric spanner), we will make use of the following lemma.

Lemma F.3 (Lemma 3.5 restated)**.** *Let μ be the uniform distribution over the C-approximate barycentric spanner, S, of $\mathcal{A} \subset \mathbb{R}^d$, Also, let $\lambda_{min}(\mu)$ be the minimum eigenvalue of the co-occurrence matrix under the law of μ. Then, it holds that,*

$$\lambda_{min}(\mu) \geq \frac{1}{C^2 d^3}.$$

Proof. Let v_i, for $i \in [d]$, be the spanner vectors of S, and let B be the matrix with v_i as columns. Using the barycentric spanner property, we know that the coefficients $\lambda_{M,i}$ for all $M \in A$ satisfy the following:

$$M = \sum_i \lambda_{M,i} v_i = B\boldsymbol{\lambda}, \quad |\lambda_{M,i}| \leq C$$

where $\boldsymbol{\lambda}$ being the vector containing the coefficients. For all $M \in span(\mathcal{A})$, we will show that $MM^T = B\lambda\lambda^T B^T \leq nC^2 BB^T$. We have that,

$$\begin{aligned}
x^T \left(B\boldsymbol{\lambda}\boldsymbol{\lambda}^T B^T - dC^2 BB^T\right) x &= x^T B \left(\boldsymbol{\lambda}\boldsymbol{\lambda}^T - dC^2 I\right) B^T x \\
&= y^T \left(\boldsymbol{\lambda}\boldsymbol{\lambda}^T - dC^2 I\right) y, \quad \text{where } y = B^T x \\
&= y^T \boldsymbol{\lambda}\boldsymbol{\lambda}^T y - dC^2 y^T y \\
&= (y \cdot \boldsymbol{\lambda}\boldsymbol{\lambda})^2 - dC^2 \|y\|^2 \\
&\leq \|\boldsymbol{\lambda}\|^2 \|y\|^2 - dC^2 \|y\|^2 \\
&= \left(\|\boldsymbol{\lambda}\|^2 - dC^2\right) \|y\|^2
\end{aligned}$$

$$\leq 0, \quad \text{because } \|\boldsymbol{\lambda}\|^2 \leq dC^2.$$

Therefore, since MM^T and BB^T are symmetric matrices, we have that,

$$MM^T \preceq dC^2 BB^T \tag{21}$$
$$= d^2C^2\widetilde{B} \tag{22}$$

where we denote by $\widetilde{B}$ the co-occurrence matrix under the law of μ.

Now, let μ' be any distribution on $\mathcal{A}$. Then, we have that $\mathbb{E}_{M\sim\mu'}\left[MM^T\right] \preceq C^2d^2\widetilde{B}$, which implies that,

$$\lambda_{\min}(\mu) \geq \frac{\lambda_{\min}(\mu')}{C^2d^2}.$$

However, from [CBL12, BCBK12], we know that there exists a distribution with a minimum eigenvalue of the induced co-occurrence matrix being at least $1/d$. This distribution is the John's exploration distribution, which always exists for the action sets of combinatorial bandits and provides a minimum eigenvalue of the co-occurrence matrix being at least $1/d$ (see [BCBK12], last sentence of the proof of Theorem 4, and in [CBL03] page 9: "$\lambda_{min} = \Omega(1/d)$ is achievable for all classes S"). □

F.2 Analysis of Swap-ComBCP

Lemma F.4 (Lemma F.4 and Lemma 3.14 combined)**.** *For all $k \in \{1, ..., K-1\}$, $l \in \left[\left\lceil \frac{T}{H^k} \right\rceil\right]$ and $h \in [H]$, Lazy-ComBCP satisfies the following properties:*

1. *(Unbiasedness):* $\mathbb{E}[mq_{k,l,h} \cdot \widetilde{X}_{k,l,h}] = \mathbb{E}[mq_{k,l,h} \cdot X_{k,l,h}]$
2. *(Boundedness):* $\|\widetilde{R}_{k,l,h,\tau}\|_2 \leq \frac{C^2d^3\sqrt{m}}{\gamma}$
3. *(Bounded variance):* $\mathbb{E}[q_{k,l,h} \cdot \widetilde{X}^2_{k,l,h}] \leq \frac{H^{2k-2}d^4C^2}{\gamma}$

Proof. • 1.

For each (k, l, h), the following hold:

$$\sum_{\tau=1}^{H^{k-1}} \mathbb{E}\left[\widetilde{R}_{k,l,h,\tau} \,\middle|\, \mathcal{F}_{k,l,h,\tau}\right] = \sum_{\tau=1}^{H^{k-1}} \mathbb{E}\left[r_{k,l,h,\tau}\Sigma^{-1}_{k,l,h,\tau}M_{k,l,h,\tau} \,\middle|\, \mathcal{F}_{k,l,h,\tau}\right] \tag{23}$$
$$= \sum_{\tau=1}^{H^{k-1}} \mathbb{E}\left[\Sigma^{-1}_{k,l,h,\tau}M_{k,l,h,\tau}M_{k,l,h,\tau} \cdot R_{k,l,h,\tau} \,\middle|\, \mathcal{F}_{k,l,h,\tau}\right] \tag{24}$$
$$= \sum_{\tau=1}^{H^{k-1}} \left(\Sigma^{-1}_{k,l,h,\tau}\Sigma_{k,l,h,\tau}R_{k,l,h,\tau}\right) \tag{25}$$
$$= \sum_{\tau=1}^{H^{k-1}} R_{k,l,h,\tau} \tag{26}$$

Using the above, we have

$$mq_{k,l,h} \cdot \sum_{\tau=1}^{H^{k-1}} \mathbb{E}\left[\widetilde{R}_{k,l,h,\tau} \,\middle|\, \mathcal{F}_{k,l,h,\tau}\right] = mq_{k,l,h} \cdot \sum_{\tau=1}^{H^{k-1}} R_{k,l,h,\tau} \tag{27}$$

$$= mq_{k,l,h} \cdot X_{k,l,h} \tag{28}$$

Taking expectations and using the tower property in eq. (28) we get

$$\mathbb{E}[mq_{k,l,h} \cdot \widetilde{X}_{k,l,h}] = \mathbb{E}[mq_{k,l,h} \cdot X_{k,l,h}] \tag{29}$$

- 2.

For all $k \in \{2, ..., K+1\}$, $l \in \left[\left\lceil \frac{T}{H^k} \right\rceil\right]$ and $h \in [H]$, we have:

$$\|\widetilde{R}_{k,l,h,\tau}\|_\infty \le \|\widetilde{R}_{k,l,h,\tau}\|_2 \tag{30}$$

$$= \left\|r_{k,l,h,\tau}\Sigma_{k,l,h}^{-1} M_{k,l,h,\tau}\right\|_2 \tag{31}$$

$$\le \left\|\Sigma_{k,l,h,\tau}^{-1} M_{k,l,h,\tau}\right\|_2 \tag{32}$$

$$= \sqrt{M_{k,l,h,\tau}^\top \Sigma_{k,l,h,\tau}^{-1} \Sigma_{k,l,h}^{-1} M_{k,l,h,\tau}} \tag{33}$$

$$\le \|M_{k,l,h,\tau}\|_2 \sqrt{\lambda_{\max}\left(\Sigma_{k,l,h,\tau}^{-1} \Sigma_{k,l,h,\tau}^{-1}\right)} \tag{34}$$

$$= \sqrt{m}\sqrt{\lambda_{\max}\left(\Sigma_{k,l,h,\tau}^{-1} \Sigma_{k,l,h,\tau}^{-1}\right)} \tag{35}$$

$$= \sqrt{m}\lambda_{\max}\left(\Sigma_{k,l,h,\tau}^{-1}\right) \tag{36}$$

$$= \frac{\sqrt{m}}{\lambda_{\min}\left(\Sigma_{k,l,h,\tau}\right)} \tag{37}$$

$$\le \frac{\sqrt{m}}{\gamma\lambda_{\min}\left(\widetilde{B}\right)} \tag{38}$$

$$\le \frac{C^2 d^3 \sqrt{m}}{\gamma} \tag{39}$$

where in 38 we defined $\widetilde{B}$ to be the co-occurrence matrix under the law of the uniform distribution over the barycentric spanner, and also we made use of the Weyl's inequality. We also define B to be a matrix whose columns are the vectors of the approximate barycentric spanner S. In 39 we used Lemma F.3.

We also have that,

$$\sup_{M \in \mathcal{A}} M^T \widetilde{B}^{-1} M \le \sup_{\alpha \in [-C,C]^d} \alpha^T B^T \widetilde{B}^{-1} B\alpha \tag{40}$$

$$\le \sup_{\|\alpha\| = \|\beta\| = C\sqrt{d}} \alpha^T B^T \widetilde{B}^{-1} B\alpha \tag{41}$$

$$\leq C^2 d \cdot \|B^T \widetilde{B}^{-1} B\|_2 \tag{42}$$
$$= C^2 d \cdot \lambda_{\max}\left(B^T \widetilde{B}^{-1} B\right) \tag{43}$$
$$= C^2 d \cdot \frac{1}{\lambda_{\min}\left(B^{-1} \widetilde{B} B^{-T}\right)} \tag{44}$$
$$= C^2 \frac{d}{\lambda_{\min}\left(B^{-1}\left(\frac{1}{d} B B^T\right) B^{-T}\right)} \tag{45}$$
$$\leq C^2 d^2 \tag{46}$$

- 3.

For all $k \in \{1, ..., K-1\}$, $l \in \left[\left\lceil \frac{T}{H^k} \right\rceil\right]$ and $h \in [H]$, we have:

$$\mathbb{E}_{k,l,h}\left[q_{k,l,h} \cdot \widetilde{X}^2_{k,l,h}\right] = \frac{1}{m}\mathbb{E}_{k,l,h}\left[m q_{k,l,h} \cdot \widetilde{X}^2_{k,l,h}\right] \tag{47}$$
$$= \frac{1}{m}\mathbb{E}_{k,l,h}\left[\sum_{M \in \mathcal{A}} \widetilde{p}_{k,l,h}(M) M \cdot \widetilde{X}^2_{k,l,h}\right] \tag{48}$$
$$= \frac{1}{m}\mathbb{E}_{k,l,h}\left[\widehat{M}_{k,l,h} \cdot \widetilde{X}^2_{k,l,h}\right] \tag{49}$$
$$= \frac{1}{m}\mathbb{E}_{k,l,h}\left[\widehat{M}_{k,l,h} \cdot \left(\sum_{\tau=1}^{H^{k-1}} \widetilde{R}_{k,l,h,\tau}\right)^2\right] \tag{50}$$
$$\leq \frac{H^{k-1}}{m}\mathbb{E}_{k,l,h}\left[\widehat{M}_{k,l,h} \cdot \sum_{\tau=1}^{H^{k-1}} \widetilde{R}^2_{k,l,h,\tau}\right] \tag{51}$$
$$= \frac{H^{k-1}}{m}\sum_{\tau=1}^{H^{k-1}} \mathbb{E}_{k,l,h}\left[\widehat{M}^2_{k,l,h} \cdot \widetilde{R}^2_{k,l,h,\tau}\right] \tag{52}$$
$$\leq \frac{H^{k-1}}{m}\sum_{\tau=1}^{H^{k-1}} \mathbb{E}_{k,l,h}\left[M^T_{k,l,h,\tau} \Sigma^{-1}_{k,l,h,\tau} \widehat{M}_{k,l,h} \widehat{M}^T_{k,l,h} \Sigma^{-1}_{k,l,h,\tau} M_{k,l,h,\tau}\right] \tag{53}$$
$$\leq \frac{H^{k-1}}{m}\sum_{\tau=1}^{H^{k-1}} \mathbb{E}_{k,l,h}\left[\lambda_{\max}(\widehat{M}_{k,l,h} \widehat{M}^T_{k,l,h}) M^T_{k,l,h,\tau} \Sigma^{-2}_{k,l,h,\tau} M_{k,l,h,\tau}\right] \tag{54}$$
$$\leq \frac{H^{k-1}}{m} \cdot \max_{M \in \mathcal{A}} \lambda_{\max}(M M^T) \cdot \sum_{\tau=1}^{H^{k-1}} \mathbb{E}_{k,l,h}\left[M^T_{k,l,h,\tau} \Sigma^{-2}_{k,l,h,\tau} M_{k,l,h,\tau}\right] \tag{55}$$
$$\leq H^{k-1}\sum_{\tau=1}^{H^{k-1}} \mathbb{E}_{k,l,h}\left[M^T_{k,l,h,\tau} \Sigma^{-2}_{k,l,h,\tau} M_{k,l,h,\tau}\right] \tag{56}$$

where in 48 we used the Carathéodory decomposition step of the algorithm, in 49 we defined $\widehat{M}_{k,l,h}$ to be a random pure action under the law of $\widetilde{p}_{k,l,h}$, independent of $M_{k,l,h,\tau}$ which is under the law of $\widehat{p}_{k,l,h}$, in 51 we used the inequality $(a_1 + ... + a_N)^2 \leq N \sum_{i=1}^N a_i^2$, in 52 we used the fact $\widehat{M}_{k,l,h} = \widehat{M}^2_{k,l,h}$ (squares are in an element-wise manner), in 53 we used the the fact that the bandit feedback reward $r_{k,l,h,\tau} \leq 1$, and also we expand the inner product, in 54 we used the inequality

$x^T ABAx^T \le \lambda_{\max}(B) x^T A^2 x$ for any vector $x \in \mathbb{R}^d$ and symmetric matrices $A, B \in \mathbb{R}^{d\times d}$, in 56 we used the fact that $\max_{M\in\mathcal{A}} \lambda_{\max}(MM^T) \le m$.

The next thing to do is to bound each summand of the above sum. Before doing so, we note that each of these summands is an expectation under the law of the master's distribution $\widehat{p}_{k,l,h,\tau}$. Using Lemma E.1, we have the following:

$$\mathbb{E}_{k,l,h}\left[M_{k,l,h,\tau}^T \Sigma_{k,l,h,\tau}^{-2} M_{k,l,h,\tau}\right] \le d\lambda_{\min}^{-1}(\Sigma_{k,l,h,\tau}) \tag{57}$$

$$\le \frac{d}{\gamma \lambda_{\min}(\widetilde{B})} \tag{58}$$

$$\le \frac{C^2 d^4}{\gamma} \tag{59}$$

where in 58 we used Weyl's inequality applying to $\lambda_{\min}(\Sigma_{k,l,h,\tau}) \ge \gamma\lambda_{\min}(\widetilde{B})$, and in 59 we used lemma F.3.

Putting the above all together, we have the following:

$$\mathbb{E}_{k,l,h}\left[q_{k,l,h} \cdot \widetilde{X}_{k,l,h}^2\right] \le \frac{H^{2k-2} d^4 C^2}{\gamma} \tag{60}$$

Finally, taking expectations in both sides we get the desired result:

$$\mathbb{E}\left[q_{k,l,h} \cdot \widetilde{X}_{k,l,h}^2\right] \le \frac{H^{2k-2} d^4 C^2}{\gamma} \tag{61}$$

□

Let $p_{k,l}^*$ be an optimal (deterministic) policy of the adversary, with $p_{k,l}^*(M_{k,l}^*) = 1$.

Proposition F.5 (Proposition 3.10 restated)**.** *Let $q_{k,l}^* = \frac{M_{k,l}^*}{m}$. It holds that*

$$mq_{k,l}^* = \sum_{M\in\mathcal{A}} p_{k,l}^*(M) M.$$

Proof. Using the fact that $mq_{k,l}^* \in co(\mathcal{M})$, from the Carathéodory theorem there exists a distribution (policy), $\widehat{p}_{k,l}$, over $\mathcal{M}$ such that

$$\sum_{M\in\mathcal{A}} \widehat{p}_{k,l}(M) M = mq_{k,l}^* \tag{62}$$

$$\implies \sum_{M\in\mathcal{A}} \widehat{p}_{k,l}(M) M = M_{k,l}^* \tag{63}$$

$$\implies \widehat{p}_{k,l} = p_{k,l}^* \tag{64}$$

□

Requiring $\eta \le \frac{\gamma}{H^{k-1}C^2d^3\sqrt{m}}$, by Lemma F.4, we have that $\eta\|\widetilde{X}_{k,l,h}\|_\infty \le 1$. This condition will be shown to be satisfied later in the analysis. We have the following lemma.

Lemma F.6 (Lemma 3.12 restated)**.** *If $\eta\|\widetilde{X}_{k,l,h}\|_\infty \le 1$, it holds that*

$$\begin{aligned}\sum_{h=1}^{H} q^*_{k,l}\cdot\widetilde{X}_{k,l,h} - \sum_{h=1}^{H} q_{k,l,h}\cdot\widetilde{X}_{k,l,h} &\le \eta\sum_{h=1}^{H} q_{k,l,h}\cdot\widetilde{X}^2_{k,l,h} + \frac{KL(q^*_{k,l}, q_{k,l,1})}{\eta}\\ &\le \eta\sum_{h=1}^{H} q_{k,l,h}\cdot\widetilde{X}^2_{k,l,h} + \frac{\log d}{\eta},\end{aligned}$$

where $\widetilde{X}^2_t$ is the vector that is the coordinate-wise square of $\widetilde{X}_t$.

Proof. The proof directly follows the proof of Lemma A1 [BP16], and also by using the fact that

$$\begin{aligned}\mathrm{KL}(q^*_{k,l}, q_{k,l,1}) &= \sum_{i=1}^{d} q^*_{k,l}(i)\log\frac{q^*_{k,l}(i)}{q_{k,l,1}(i)} &(65)\\ &= -\frac{1}{m}\sum_{i:M^*_{k,l}(i)=1}\log\frac{m}{d} &(66)\\ &= \log d - \log m &(67)\\ &\le \log d. &(68)\end{aligned}$$

where in 67 we used the fact that $\|M\|_1 = m$.

□

Theorem F.7 (Theorem 3.9 restated)**.** *The external regret of Lazy-ComBCP$_{k,l}$ is at most $3H^{k-1}H^{2/3}d^3m^{3/2}\log d$.*

Proof. To upper bound Reg of the learner k,l, we have:

$$\begin{aligned}\mathrm{Reg} &= \max_{M\in\mathcal{A}}\left\{\sum_{h=1}^{H} X_{k,l,h}\cdot M\right\} - \sum_{h=1}^{H}\mathbb{E}\left[\sum_{M\in\mathcal{A}} p_{k,l,h}(M)X_{k,l,h}\cdot M\right] &(69)\\ &= \sum_{h=1}^{H} X_{k,l,h}\cdot M^*_{k,l} - \sum_{h=1}^{H}\mathbb{E}\left[\sum_{M\in\mathcal{A}}\left[(1-\gamma)\widetilde{p}_{k,l,h}(M) + \gamma\mu(M)\right]X_{k,l,h}\cdot M\right] &(70)\\ &= \sum_{h=1}^{H} mq^*_{k,l}\cdot X_{k,l,h} - \sum_{h=1}^{H}\mathbb{E}\left[(1-\gamma)mq_{k,l,h}\cdot X_{k,l,h} + \gamma\sum_{M\in\mathcal{A}}\mu(M)X_{k,l,h}\cdot M\right] &(71)\\ &\le \sum_{h=1}^{H} mq^*_{k,l}\cdot X_{k,l,h} - \sum_{h=1}^{H}\mathbb{E}\left[(1-\gamma)mq_{k,l,h}\cdot X_{k,l,h}\right] &(72)\\ &= \sum_{h=1}^{H}\mathbb{E}\left[mq^*_{k,l}\cdot X_{k,l,h} - mq_{k,l,h}\cdot X_{k,l,h}\right] + \gamma\sum_{h=1}^{H}\mathbb{E}\left[mq_{k,l,h}\cdot X_{k,l,h}\right] &(73)\\ &\le \sum_{h=1}^{H}\mathbb{E}\left[mq^*_{k,l}\cdot X_{k,l,h} - mq_{k,l,h}\cdot X_{k,l,h}\right] + \gamma H^k &(74)\end{aligned}$$

$$= \sum_{h=1}^{H} \mathbb{E}\left[m q^*_{k,l} \cdot \widetilde{X}_{k,l,h} - m q_{k,l,h} \cdot \widetilde{X}_{k,l,h}\right] + \gamma H^k \tag{75}$$

$$\leq m\eta \sum_{h=1}^{H} \mathbb{E}\left[q_{k,l,h} \cdot \widetilde{X}^2_{k,l,h}\right] + \frac{m \log d}{\eta} + \gamma H^k \tag{76}$$

$$\leq m\eta \sum_{h=1}^{H} \frac{H^{2k-2} d^4 C^2}{\gamma} + \frac{m \log d}{\eta} + \gamma H^k \tag{77}$$

$$= \frac{\eta H^{2k-1} m d^4 C^2}{\gamma} + \frac{m \log d}{\eta} + \gamma H^k \tag{78}$$

where: in 71 we used Proposition F.5, and also the Carathéodory decomposition of $\widetilde{p}_{k,l,h}$, in 74 we used the Holder's inequality utilizing the fact that $\|X_{k,l,h}\|_\infty \leq H^{k-1}$, in 75 we used Lemma F.4 (1), in 76 we used Lemma F.6, and in 77 we used Lemma F.4 (3).

Now, recall that the maximum reward per time step (i.e., a meta-day) is H^{k-1}, while the horizon of our online learning setting for each learner is H. Therefore, we need each learner to achieve external regret sublinear to H^k, in order our algorithm to achieve no-swap-regret learning.

Setting $\eta' = H^{k-1}\eta$, we have:

$$\text{Reg} \leq \frac{\eta' H^k m d^4 C^2}{\gamma} + \frac{H^{k-1} m \log d}{\eta'} + \gamma H^k \tag{79}$$

$$= H^{k-1}\left(\frac{\eta' H m d^4 C^2}{\gamma} + \frac{m \log d}{\eta'} + \gamma H\right) \tag{80}$$

Now, the following must hold (required for the Online Mirror Descent Lemma):

$$\eta \leq \frac{\gamma}{H^{k-1} C^2 d^3 \sqrt{m}} \Rightarrow \eta' \leq \frac{\gamma}{C^2 d^3 \sqrt{m}}.$$

Setting $\eta' = \frac{1}{d^3 \sqrt{m} H^{2/3}}$, $\gamma = H^{-1/3}$, and $C = 2$ we have:

$$\text{Reg} \leq 3 H^{k-1} H^{2/3} d^3 m^{3/2} \log d.$$

□

Theorem F.8 (Theorem 4.1 restated)**.** *Setting* $H = 27\lfloor \log(T) \rfloor^3 d^9 m^{9/2} \log^3 d$ *and* $K = \lfloor \log_H(T) \rfloor$, *Swap-ComBCP satisfies the following swap-regret guarantee:*

$$\text{SwapReg} \leq \frac{45 \log(d \log T) T}{\log T}.$$

Proof. Setting $g(m, d) = 3 d^3 m^{3/2} \log d$ and putting everything together in 13, we have:

$$\text{SwapReg}_T(\phi) \leq \frac{1}{K} \sum_{k=1}^{K-1} \sum_{l=1}^{\lceil \frac{T}{H^k} \rceil} \text{Reg}(\text{Lazy-ComBCP}_{k,l}) + \frac{T}{K} \tag{81}$$

$$\leq \frac{1}{K}\sum_{k=1}^{K-1}\sum_{l=1}^{\left\lceil \frac{T}{H^k}\right\rceil} g(m,d)H^{k-1}H^{2/3} + \frac{T}{K} \tag{82}$$

$$\leq \frac{1}{K}\sum_{k=1}^{K-1}\left(\frac{T}{H^k}+1\right) g(m,d)H^{k-1}H^{2/3} + \frac{T}{K} \tag{83}$$

$$= \frac{Tg(m,d)}{H^{1/3}} + \frac{1}{K}H^{2/3}g(m,d)\sum_{j=0}^{K-2} H^j + \frac{T}{K} \tag{84}$$

$$= \frac{Tg(m,d)}{H^{1/3}} + \frac{1}{K}H^{2/3}g(m,d)\frac{H^{K-1}-1}{H-1} + \frac{T}{K} \tag{85}$$

$$\leq \frac{Tg(m,d)}{H^{1/3}} + \frac{2}{K}g(m,d)\frac{H^{K-1}}{H^{1/3}} + \frac{T}{K} \tag{86}$$

$$\tag{87}$$

We set $H = \lfloor \log(T) \rfloor^3 (g(m,d))^3$ and $K = \lfloor \log_H(T) \rfloor$ and we obtain:

$$\text{SwapReg}_T(\phi) \leq \frac{T}{\lfloor \log(T) \rfloor} + 2\frac{T}{KH\lfloor \log(T) \rfloor} + \frac{T}{K} \tag{88}$$

$$\leq 2\frac{T}{\lfloor \log(T) \rfloor} + \frac{T}{\lfloor \log_H(T) \rfloor} \tag{89}$$

$$\leq 2\frac{T}{\log(T)-1} + \frac{T}{\log_H(T)-1} \tag{90}$$

$$\leq 4\frac{T}{\log(T)} + 2\frac{T}{\log_H(T)} \tag{91}$$

$$= 4\frac{T}{\log(T)} + 2\frac{T\log(H)}{\log(T)} \tag{92}$$

$$\leq \frac{(4 + 6\log(\log(T)) + 6\log(g(m,d)))\,T}{\log(T)} \tag{93}$$

$$\leq \frac{45\log(d\log T)T}{\log T} \tag{94}$$

□

F.3 Issue with the exploration scheme of CombExp

In settings where the components of the action set appear with highly uneven frequencies—some occurring in far more actions than others—CombExp faces a challenge in its exploration strategy. Infrequently occurring components might be insufficiently explored, which becomes problematic if these components yield high rewards. As a result, the learner risks overlooking them, potentially leading to significantly suboptimal performance and increased regret. Next we construct one such example in the setting of online path planning to demonstrate the issue.

The idea is to construct a DAG where all edges but one participate in an exponential number of paths and one edge in a small number of paths. The latter edge gives high reward while all the other edges give small reward. For simplicity consider that the rewards are time invariant, so the no-regret learning objective reduces to correctly identifying the high-reward edge and selecting paths that include it. We show that, in such a construction, CombExp requires, with high probability, an exponential number of rounds to identify the high-reward edge. Consequently, for any horizon that is only polynomially large, the algorithm incurs regret that grows linearly with the horizon. Next we provide the formal construction of the counterexample.

Fix an integer $n \geq 1$. Define the directed acyclic graph $G = (V, E)$ where the set of vertices is

$$V = \{S, D\} \cup \{A_i, B_i \mid i = 1, 2, \ldots, n\}.$$

and the set of edges is defined as:

$$\begin{aligned} E \;=\; & \{(S, D), (S, A_1), (S, B_1)\} \\ & \cup\; \{(A_i, A_{i+1}), (A_i, B_{i+1}), (B_i, A_{i+1}), (B_i, B_{i+1}) \mid i = 1, \ldots, n-1\} \\ & \cup\; \{(A_n, t), (B_n, t)\}. \end{aligned}$$

In this DAG the number of nodes is $|V| = 2n + 2$, the number of edges $|E| = 4n + 1$ and the maximal path length is $n + 1$. In particular, there are 2^n paths of length $n + 1$ and one path of length 1. Thus, in the binary representation of paths in this setting we have $d = 4n + 1$, $m = n + 1$ and $|\mathcal{A}| = 2^n + 1$.

Regarding the usage of edges in the DAG we have the following:

- (S, D) is used by exactly 1 path.
- Edges (S, A_1), (S, B_1), (A_n, D), (B_n, D) are each used by 2^{n-1} paths.
- Internal edges $(X_i \to Y_{i+1})$ for $X, Y \in \{A, B\}$, $1 \leq i \leq n - 1$ are each used by 2^{n-2} paths.

We consider an online learning setting where actions are paths from the starting node S to the destination node D in the graph G we described above. In the edge incidence vectors $M \in \mathcal{A}$ of $S - D$ paths in G suppose that the first coordinate $i = 1$ corresponds to the (shortcut) edge (S, D). That is, $M(1) = 1$ if the path represented by binary vector M contains the edge (S, D), otherwise $M(1) = 0$.

Consider the reward sequence $R_t(i) = \mathbb{1}[i = 1]$, $\forall t \in [T]$. We will study how CombEXP performs in this sequence. Obviously, the best fixed action M^* in hindsight is selecting the path that uses the shortcut edge (S, D). M^* gives reward 1 at each timestep, while all other actions give zero reward. The initial distribution assigns exponentially small probability to the optimal edge : $\mu_{\min} = m\mu^0(1) = \frac{1}{2^n+1}$. Moreover, we observe that in the decomposition p_0 of $q_0' = \mu^0$ the optimal action M^* appears with weight $\mu_{\min}$. This means that at $t = 1$ CombEXP plays M^* with probability

Algorithm 3 CombEXP [CTMSP+15]

Initialization: Let $\mu^0(i) = \frac{1}{m|\mathcal{A}|}\sum_{M\in\mathcal{A}} M(i)$, $\forall i \in [d]$, $\mu_{\min} = \min_{i\in[d]} m\mu^0(i)$ and $\underline{\lambda} = \lambda_{>0,\min}(\mathbb{E}_{M\sim\mathcal{U}(\mathcal{A})}[MM^\top])$

$$\text{Set } q_0 = \mu^0,\ \gamma = \frac{\sqrt{m\log\mu_{\min}^{-1}}}{\sqrt{m\log\mu_{\min}^{-1}}+\sqrt{C(Cm^2d+m)T}} \text{ and } \eta = \gamma C, \text{ with } C = \frac{\underline{\lambda}}{m^{3/2}}.$$

for $t = 1, \ldots T$ **do**

Mixing: Let $q'_{t-1} = (1-\gamma)q_{t-1} + \gamma\mu^0$.

Decomposition: Select a distribution p_{t-1} over $\mathcal{A}$ such that $\sum_M p_{t-1}(M)M = mq'_{t-1}$.

Sampling: Select a random arm M_t with distribution p_{t-1} and incur a reward $r_t = R_t \cdot M_t$.

Estimation: Let $\Sigma_{t-1} = \mathop{\mathbb{E}}_{M\sim p_{t-1}}[MM^\top]$. Set $\tilde{R}_t = r_t\Sigma_{t-1}^+ M_t$, where Σ_{t-1}^+ is the pseudo-inverse of Σ_{t-1}.

Update: Set $\tilde{q}_t(i) \propto q_{t-1}(i)\exp(\eta\tilde{R}_t(i)),\ \forall i \in [d]$.

Projection: Set q_t to be the projection of $\tilde{q}_t$ onto the set $\mathcal{P}$ using the KL divergence.

$\frac{1}{2^n+1}$. With probability $1 - \frac{1}{2^n+1}$ an action $M \neq M^*$ is played and reward $r_1 = 0$ is received. In this case, $\tilde{R}_t$ is the zero vector and no update is performed on q, that is $q_1 = q_0 = \mu^0$. This pattern is repeated in the next timesteps (i.e. $q_t = q_{t-1}, r_t = 0$) and with probability at least $1 - \frac{T}{2^n+1}$ CombEXP has realized regret $\sum_{t=1}^T R_t \cdot M^* - r_t = T$.

Apart from paths other combinatorial structures can create this issue as well. For example, consider partitions of n indistinguishable elements to k labeled groups under the binary representation of [KPF+25] and suppose that the learner receives non zero reward only if it assigns all n elements to one particular group. Then, the realized regret of CombEXP is $\Theta(T)$ with probability at least $1 - \frac{T}{\binom{n+k-1}{k-1}}$.

G SWAP-COMBAND

Algorithm 4 LAZY-COMBAND$_{k,l}$

Initialize $\widetilde{p}_{k,l,1} = [1/|\mathcal{A}|, ..., 1/|\mathcal{A}|] \in \Delta^{|\mathcal{A}|}$, $\mu \in \Delta^{|\mathcal{A}|}$, $\gamma = \frac{1}{H^{1/3}}$, $\eta = \frac{\lambda_{\min}}{H^{k-1/3}m}$

$H' = \begin{cases} H & \text{if } l \le \frac{T}{H^k} \\ \left\lceil \frac{T - \left\lfloor \frac{T}{H^k} \right\rfloor H^k}{H^{k-1}} \right\rceil & \text{othw.} \end{cases}$ /*Play for H meta-days or until the time limit T is reached*/

for $h = 1, 2, \ldots, H'$ **do**

  Mixing: Let $p_{k,l,h} = (1-\gamma)\widetilde{p}_{k,l,h} + \gamma\mu$

  Meta-day: Fix $p_{k,l,h}$ for H^{k-1} days and aggregate the rewards of these days:

$$\widetilde{X}_{k,l,h}(i) = \sum_{\tau=(\ell-1)H^k+(h-1)H^{k-1}+1}^{\min((\ell-1)H^k+hH^{k-1},T)} \widetilde{R}_\tau(i), \quad \forall i \in [d]$$

  Update: $\widetilde{p}_{k,l,h+1}(M) = \widetilde{p}_{k,l,h}(M)\exp(\eta\widetilde{X}_{k,l,h})$, $\forall M \in \mathcal{A}$

Lemma G.1. *(Corollary 12, in [CBL12]) For all t and $x \in \mathbb{R}^d$, it holds that $\Sigma_t\Sigma_t^+ x = x^{\|}$, where $x^{\|}$ is the orthogonal projection of x on the linear span of $\mathcal{A}$.*

Lemma G.2. *Let $\lambda_{\min}$ be the smallest nonzero eigenvalue of the co-occurrence matrix Σ under the law of μ. For all $k \in \{2, ..., K+1\}$, $l \in \left[\left\lceil \frac{T}{H^k} \right\rceil\right]$ and $h \in [H]$, LAZY-COMBAND satisfies the following properties:*

1. *(Unbiasedness):* $\mathbb{E}\left[\sum_{M\in\mathcal{A}} \widetilde{p}_{k,l,h}(M)\widetilde{X}_{k,l,h} \cdot M\right] = \mathbb{E}\left[\sum_{M\in\mathcal{A}} \widetilde{p}_{k,l,h}(M)X_{k,l,h} \cdot M\right]$
2. *(Boundedness):* $\left|\widetilde{X}_{k,l,h} \cdot M\right| \le \frac{H^{k-1}m}{\gamma\lambda_{min}}$
3. *(Bounded variance):* $\mathbb{E}\left[\sum_{M\in\mathcal{A}} \widetilde{p}_{k,l,h}(M)\widetilde{X}_{k,l,h}^2 \cdot M\right] \le \frac{H^{2k-2}m^2d}{\gamma\lambda_{min}}$

Proof. • 1.

We have that,

$$\sum_{\tau=1}^{H^{k-1}} \mathbb{E}_{k,l,h,\tau}\left[\widetilde{R}_{k,l,h,\tau}\right] = \sum_{\tau=1}^{H^{k-1}} \mathbb{E}_{k,l,h,\tau}\left[r_{k,l,h,\tau}\Sigma_{k,l,h,\tau}^+ M_{k,l,h,\tau}\right] \tag{95}$$

$$= \sum_{\tau=1}^{H^{k-1}} \mathbb{E}_{k,l,h,\tau}\left[\Sigma_{k,l,h,\tau}^+ M_{k,l,h,\tau}M_{k,l,h,\tau} \cdot R_{k,l,h,\tau}\right] \tag{96}$$

$$= \sum_{\tau=1}^{H^{k-1}} \left(\Sigma_{k,l,h,\tau}^+\Sigma_{k,l,h,\tau}R_{k,l,h,\tau}\right) \tag{97}$$

$$= \sum_{\tau=1}^{H^{k-1}} R_{k,l,h,\tau}^{\|} \tag{98}$$

where in 98 we used Lemma G.1. Now, based on the above, for any $M \in \mathcal{A}$, we have that,

$$\sum_{\tau=1}^{H^{k-1}} \mathbb{E}_{k,l,h,\tau}\left[\widetilde{R}_{k,l,h,\tau}\right] \cdot M = \sum_{\tau=1}^{H^{k-1}} R^{\|}_{k,l,h,\tau} \cdot M \tag{99}$$

$$= \sum_{\tau=1}^{H^{k-1}} R_{k,l,h,\tau} \cdot M \tag{100}$$

$$= X_{k,l,h,\tau} \cdot M \tag{101}$$

where in 100 we used the fact that $R_{k,l,h,\tau} = R^{\|}_{k,l,h,\tau} + R^{\perp}_{k,l,h,\tau}$, and also that $R^{\perp}_{k,l,h,\tau} \cdot M = 0$. The above also implies that

$$\mathbb{E}_{k,l,h,\tau}\left[\sum_{M\in\mathcal{A}} \widetilde{p}_{k,l,h}(M)\widetilde{X}_{k,l,h} \cdot M\right] = \mathbb{E}_{k,l,h,\tau}\left[\sum_{M\in\mathcal{A}} \widetilde{p}_{k,l,h}(M) \sum_{\tau=1}^{H^{k-1}} \widetilde{R}_{k,l,h,\tau} \cdot M\right] \tag{102}$$

$$= \sum_{M\in\mathcal{A}} \widetilde{p}_{k,l,h}(M)\mathbb{E}_{k,l,h,\tau}\left[\sum_{\tau=1}^{H^{k-1}} \widetilde{R}_{k,l,h,\tau} \cdot M\right] \tag{103}$$

$$= \sum_{M\in\mathcal{A}} \widetilde{p}_{k,l,h}(M)\mathbb{E}_{k,l,h,\tau}\widetilde{X}_{k,l,h} \cdot M \tag{104}$$

where in 104 we used 101. Finally, taking expectations in both sides in 104, and using the tower property, we get the desired result.

- 2.

We have that,

$$\left|\widetilde{X}_{k,l,h} \cdot M\right| \leq \sum_{\tau=1}^{H^{k-1}} \left|r_{k,l,h,\tau}\right| \left|M^T \Sigma^{+}_{k,l,h,\tau} M_{k,l,h,\tau}\right| \tag{105}$$

$$\leq H^{k-1} \left\|\Sigma^{+}_{k,l,h,\tau}\right\|_2 \max_{M\in\mathcal{A}} \|M\|^2 \tag{106}$$

$$\leq \frac{H^{k-1}m}{\lambda_{\min}(\Sigma_{k,l,h,\tau})} \tag{107}$$

$$\leq \frac{H^{k-1}m}{\gamma\lambda_{\min}} \tag{108}$$

where in 106 we used the fact that $|r_t| \leq 1$, and in 108 we used the Weyl's inequality on $\lambda_{\min}\left(\Sigma_{k,l,h,\tau}\right) \geq \gamma\lambda_{\min}$, where we denote by $\lambda_{\min}\left(\Sigma_{k,l,h,\tau}\right)$ the smallest nonzero eigenvalue of the co-occurrence matrix under the law of $\widehat{p}_{k,l,h,\tau}$, and by $\lambda_{\min}$ the smallest nonzero eigenvalue of the co-occurrence matrix under the law of μ.

- 3.

For all $k \in \{1, ..., K-1\}$, $l \in \left[\left\lceil \frac{T}{H^k} \right\rceil\right]$ and $h \in [H]$, we have:

Let $Var = \mathbb{E}_{k,l,h}\left[\sum_{M\in\mathcal{A}} \widetilde{p}_{k,l,h}(M)\left(\widetilde{X}_{k,l,h} \cdot M\right)^2\right]$

$$Var \leq m\mathbb{E}_{k,l,h}\left[\sum_{M\in\mathcal{A}} \widetilde{p}_{k,l,h}(M)\widetilde{X}^2_{k,l,h} \cdot M\right] \tag{109}$$

$$= m\mathbb{E}_{k,l,h}\left[\widehat{M}_{k,l,h}\cdot\widetilde{X}^2_{k,l,h}\right] \tag{110}$$

$$= m\mathbb{E}_{k,l,h}\left[\widehat{M}_{k,l,h}\cdot\left(\sum_{\tau=1}^{H^{k-1}}\widetilde{R}_{k,l,h,\tau}\right)^2\right] \tag{111}$$

$$\leq H^{k-1}m\mathbb{E}_{k,l,h}\left[\widehat{M}_{k,l,h}\cdot\sum_{\tau=1}^{H^{k-1}}\widetilde{R}^2_{k,l,h,\tau}\right] \tag{112}$$

$$= H^{k-1}m\sum_{\tau=1}^{H^{k-1}}\mathbb{E}_{k,l,h}\left[\widehat{M}^2_{k,l,h}\cdot\widetilde{R}^2_{k,l,h,\tau}\right] \tag{113}$$

$$\leq H^{k-1}m\sum_{\tau=1}^{H^{k-1}}\mathbb{E}_{k,l,h}\left[M^T_{k,l,h,\tau}\Sigma^{+}_{k,l,h,\tau}\widehat{M}_{k,l,h}\widehat{M}^T_{k,l,h}\Sigma^{+}_{k,l,h,\tau}M_{k,l,h,\tau}\right] \tag{114}$$

$$\leq H^{k-1}m\sum_{\tau=1}^{H^{k-1}}\mathbb{E}_{k,l,h}\left[\lambda_{\max}(\widehat{M}_{k,l,h}\widehat{M}^T_{k,l,h})M^T_{k,l,h,\tau}\Sigma^{+2}_{k,l,h,\tau}M_{k,l,h,\tau}\right] \tag{115}$$

$$\leq H^{k-1}m\cdot\max_{M\in\mathcal{A}}\lambda_{\max}(MM^T)\cdot\sum_{\tau=1}^{H^{k-1}}\mathbb{E}_{k,l,h}\left[M^T_{k,l,h,\tau}\Sigma^{+2}_{k,l,h,\tau}M_{k,l,h,\tau}\right] \tag{116}$$

$$\leq H^{k-1}m^2\sum_{\tau=1}^{H^{k-1}}\mathbb{E}_{k,l,h}\left[M^T_{k,l,h,\tau}\Sigma^{+2}_{k,l,h,\tau}M_{k,l,h,\tau}\right] \tag{117}$$

where in 109 we used the Holder's inequality $(a_1+...+a_N)^2\leq N\sum_{i=1}^N a_i^2$, in 110 we defined $\widehat{M}_{k,l,h}$ to be a random pure action under the law of $\widetilde{p}_{k,l,h}$, independent of $M_{k,l,h,\tau}$ which is under the law of $\widehat{p}_{k,l,h}$, in 112 we used the inequality $(a_1+...+a_N)^2\leq N\sum_{i=1}^N a_i^2$, in 113 we used the fact $\widehat{M}_{k,l,h}=\widehat{M}^2_{k,l,h}$ (squares are in an element-wise manner), in 114 we used the the fact that the bandit feedback reward $r_{k,l,h,\tau}\leq 1$, in 115 we used the inequality $x^TABAx^T\leq\lambda_{\max}(B)x^TA^2x$ for any vector $x\in\mathbb{R}^d$ and symmetric matrices $A,B\in\mathbb{R}^{d\times d}$, in 117 we used the fact that $\max_{M\in\mathcal{A}}\lambda_{\max}(MM^T)\leq m$.

The next thing to do is to bound each summand of the above sum. Before doing so, we note that each of these summands is an expectation under the law of the master's distribution $\widehat{p}_{k,l,h,\tau}$. Let $\lambda_i(\Sigma_{k,l,h,\tau})$ be the eigenvalue of $\Sigma_{k,l,h,\tau}$ corresponding to the eigenvector $u_i(\Sigma_{k,l,h,\tau})$. For brevity, we abuse notation and instead simply use $\lambda_{\tau,i}$ and $u_{\tau,i}$. Moreover, let $\lambda_{\tau,1},...,\lambda_{\tau,r}$ be the r nonzero eigenvalues with corresponding eigenvectors $u_{\tau,1},...,u_{\tau,r}$. Based on the above, using Lemma E.1, we have the following:

$$\mathbb{E}_{k,l,h}\left[M^T_{k,l,h,\tau}\Sigma^{+2}_{k,l,h,\tau}M_{k,l,h,\tau}\right]\leq d\lambda^{-1}_{\min}(\Sigma_{k,l,h,\tau}) \tag{118}$$

$$\leq\frac{d}{\gamma\lambda_{\min}} \tag{119}$$

where in 119 we used the Weyl's inequality applying to $\lambda_{\min}(\Sigma_{k,l,h,\tau})\geq\gamma\lambda_{\min}$.

Putting the above all together, we have the following:

$$\mathbb{E}_{k,l,h}\left[\sum_{M\in\mathcal{A}}\widetilde{p}_{k,l,h}(M)\widetilde{X}^2_{k,l,h}\cdot M\right]\leq\frac{H^{2k-2}m^2d}{\gamma\lambda_{\min}} \tag{120}$$

Finally, taking expectations in both sides we get the desired result:

$$\mathbb{E}\left[\sum_{M\in\mathcal{A}}\widetilde{p}_{k,l,h}(M)\widetilde{X}^2_{k,l,h}\cdot M\right]\leq\frac{H^{2k-2}m^2d}{\gamma\lambda_{\min}} \tag{121}$$

□

Requiring $\eta\leq\frac{\gamma\lambda_{\min}}{H^{k-1}m}$ (this condition will be verified later), Lazy-ComBand satisfies the following Online Mirror Descent lemma:

Lemma G.3 (Online Mirror Descent on $\mathcal{A}$)**.** *If $\eta|\widetilde{X}_{k,l,h}\cdot M|\leq 1$, it holds that*

$$\mathbb{E}\left[\sum_{h=1}^{H}M^*\cdot\widetilde{X}_{k,l,h}-\sum_{h=1}^{H}\sum_{M\in\mathcal{A}}\widetilde{p}_{k,l,h}(M)\widetilde{X}_{k,l,h}\cdot M\right]\leq\eta\mathbb{E}\left[\sum_{h=1}^{H}\sum_{M\in\mathcal{A}}\widetilde{p}_{k,l,h}(M)\left(\widetilde{X}_{k,l,h}\cdot M\right)^2\right]+\frac{\log|\mathcal{A}|}{\eta}.$$

Theorem G.4 (No-External-Regret)**.** *The external regret of Lazy-ComBand$_{k,l}$ is at most $H^{k-1}H^{2/3}\Big(md+\frac{m^2\log d}{\lambda_{min}}+$*

Proof. Aiming to bound the external regret of Lazy-ComBand, we get,

$$\begin{aligned}
\mathrm{Reg}&=\max_{M\in\mathcal{A}}\left\{\sum_{h=1}^{H}X_{k,l,h}\cdot M\right\}-\sum_{h=1}^{H}\mathbb{E}\left[\sum_{M\in\mathcal{A}}p_{k,l,h}(M)X_{k,l,h}\cdot M\right] &(122)\\
&=\sum_{h=1}^{H}X_{k,l,h}\cdot M^*_{k,l}-\sum_{h=1}^{H}\mathbb{E}\left[\sum_{M\in\mathcal{A}}p_{k,l,h}(M)X_{k,l,h}\cdot M\right] &(123)\\
&=\mathbb{E}\left[\sum_{h=1}^{H}X_{k,l,h}\cdot M^*_{k,l}\right]-\sum_{h=1}^{H}\mathbb{E}\left[\sum_{M\in\mathcal{A}}p_{k,l,h}(M)X_{k,l,h}\cdot M\right] &(124)\\
&=\mathbb{E}\left[\sum_{h=1}^{H}X_{k,l,h}\cdot M^*_{k,l}\right]-\sum_{h=1}^{H}\mathbb{E}\left[\sum_{M\in\mathcal{A}}[(1-\gamma)\widetilde{p}_{k,l,h}(M)+\gamma\mu(M)]X_{k,l,h}\cdot M\right] &(125)\\
&=\mathbb{E}\left[\sum_{h=1}^{H}X_{k,l,h}\cdot M^*_{k,l}-\sum_{h=1}^{H}\sum_{M\in\mathcal{A}}\widetilde{p}_{k,l,h}(M)X_{k,l,h}\cdot M\right]+\gamma\sum_{h=1}^{H}\mathbb{E}\left[\sum_{M\in\mathcal{A}}\widetilde{p}_{k,l,h}(M)X_{k,l,h}\cdot M\right]\\
&\quad-\gamma\sum_{h=1}^{H}\mathbb{E}\left[\sum_{M\in\mathcal{A}}\mu(M)X_{k,l,h}\cdot M\right] &(126)\\
&\leq\mathbb{E}\left[\sum_{h=1}^{H}X_{k,l,h}\cdot M^*_{k,l}-\sum_{h=1}^{H}\sum_{M\in\mathcal{A}}\widetilde{p}_{k,l,h}(M)X_{k,l,h}\cdot M\right]+2\gamma H^k &(127)\\
&=\mathbb{E}\left[\sum_{h=1}^{H}\widetilde{X}_{k,l,h}\cdot M^*_{k,l}-\sum_{h=1}^{H}\sum_{M\in\mathcal{A}}\widetilde{p}_{k,l,h}(M)\widetilde{X}_{k,l,h}\cdot M\right]+2\gamma H^k &(128)\\
&\leq\eta\mathbb{E}\left[\sum_{h=1}^{H}\sum_{M\in\mathcal{A}}\widetilde{p}_{k,l,h}(M)\left(\widetilde{X}_{k,l,h}\cdot M\right)^2\cdot M\right]+\frac{\log|\mathcal{A}|}{\eta}+2\gamma H^k &(129)\\
&\leq\frac{\eta H^{2k-1}m^2d}{\gamma\lambda_{\min}}+\frac{m\log d}{\eta}+2\gamma H^k &(130)
\end{aligned}$$

where in 127 we used the fact that $|R_{k,l,h} \cdot M| \leq 1$, in 128 we used the unbiasedness of the estimator, in 129 we applied Lemma G.3, in 130 we used Lemma G.2 (2) and that $|\mathcal{A}| \leq d^m$.

Setting $\eta' = H^{k-1}\eta$, we have:

$$\mathrm{Reg}_k \leq H^{k-1}\left(\frac{\eta' H m^2 d}{\gamma \lambda_{\min}} + \frac{m \log d}{\eta'} + 2\gamma H\right) \tag{131}$$

Now, setting $\gamma = \frac{1}{H^{1/3}}$ and $\eta' = \frac{\gamma\lambda_{\min}}{H^{1/3}m} = \frac{\lambda_{\min}}{H^{2/3}m} \Rightarrow \eta = \frac{\lambda_{\min}}{H^{k-1/3}m}$, we get,

$$\mathrm{Reg}_k \leq H^{k-1}\left(H^{2/3} m d + \frac{H^{2/3} m^2 \log d}{\lambda_{\min}} + 2H^{2/3}\right) \tag{132}$$

$$= H^{k-1}H^{2/3}\left(md + \frac{m^2 \log d}{\lambda_{\min}} + 2\right) \tag{133}$$

Notice that with the above configuration the condition $\eta \leq \frac{\gamma\lambda_{\min}}{H^{k-1}m}$ holds.

□

Theorem G.5 (No-Swap-Regret)**.** *If* $H = \lfloor \log(T) \rfloor^3 (g(m, d, \lambda_{min}))^3$ *and* $K = \lfloor \log_H(T) \rfloor$*,* SWAP-COMBAND *achieves swap-regret*

$$\mathrm{SwapReg}_T(\phi) \leq \mathcal{O}\left(\frac{T \log(d \log T)}{\log(T)}\right),$$

where $g(m, d, \lambda_{min}) = \left(m^2 d + \frac{m^2 \log d}{\lambda_{min}} + 2m\right)$.

Proof. Let $g = \left(md + \frac{m^2 [\log d]}{\lambda_{\min}} + 2\right)$. Putting the no-regret guarantees in 13, the following hold:

$$\mathrm{SwapReg}_T(\phi) \leq \frac{1}{K}\sum_{k=1}^{K-1}\sum_{l=1}^{\lceil \frac{T}{H^k} \rceil} \mathrm{Reg}(\textsc{Lazy-CombAlg}_{k,l}) + \frac{T}{K} \tag{134}$$

$$\leq \frac{1}{K}\sum_{k=1}^{K-1}\sum_{l=1}^{\lceil \frac{T}{H^k} \rceil} g(m, d, \lambda_{\min}) H^{k-1} H^{2/3} + \frac{T}{K} \tag{135}$$

$$\leq \frac{1}{K}\sum_{k=1}^{K-1}\left(\frac{T}{H^k} + 1\right) g(m, d, \lambda_{\min}) H^{k-1} H^{2/3} + \frac{T}{K} \tag{136}$$

$$= \frac{T g(m, d, \lambda_{\min})}{H^{1/3}} + \frac{1}{K} H^{2/3} g(m, d, \lambda_{\min}) \sum_{j=0}^{K-2} H^j + \frac{T}{K} \tag{137}$$

$$= \frac{T g(m, d, \lambda_{\min})}{H^{1/3}} + \frac{1}{K} H^{2/3} g(m, d, \lambda_{\min}) \frac{H^{K-1} - 1}{H - 1} + \frac{T}{K} \tag{138}$$

$$\leq \frac{T g(m, d, \lambda_{\min})}{H^{1/3}} + \frac{2}{K} g(m, d, \lambda_{\min}) \frac{H^{K-1}}{H^{1/3}} + \frac{T}{K} \tag{139}$$

We set $H = \lfloor \log(T) \rfloor^3 (g(m, d, \lambda_{\min}))^3$ and $K = \lfloor \log_H(T) \rfloor$ and we obtain:

$$\begin{aligned}
\mathrm{SwapReg}_T(\phi) &\leq \frac{T}{\lfloor \log(T) \rfloor} + 2\frac{T}{KH\lfloor \log(T) \rfloor} + \frac{T}{K} && (140)\\
&\leq 2\frac{T}{\lfloor \log(T) \rfloor} + \frac{T}{\lfloor \log_H(T) \rfloor} && (141)\\
&\leq 2\frac{T}{\log(T) - 1} + \frac{T}{\log_H(T) - 1} && (142)\\
&\leq 4\frac{T}{\log(T)} + 2\frac{T}{\log_H(T)} && (143)\\
&= 4\frac{T}{\log(T)} + 2\frac{T \log(H)}{\log(T)} && (144)\\
&\leq \frac{(4 + 6\log(\log(T)) + 6\log(g(m, d, \lambda_{\min})))\, T}{\log(T)} && (145)
\end{aligned}$$

□

Remark G.6. In Swap-ComBand, under the minimal assumption of an efficient linear minimization oracle for $\mathcal{A}$, μ can be efficiently constructed as the uniform distribution over a barycentric spanner (see Proposition F.2) and achieve $\lambda_{\min}(\mu) \geq \frac{1}{4d^3}$ (see Lemma F.3). As an alternative, μ could be set to the uniform distribution over $\mathcal{A}$, which in many cases has $1/\lambda_{\min}(\mu) = \mathcal{O}(\mathrm{poly}(d))$ (see [CBL12]).

H Per-Iteration Complexity in Well-Studied Combinatorial Settings

H.1 Applications of Swap-ComBCP

We present some important combinatorial bandit settings where Swap-ComBCP is applicable (i.e. the $L1$ norm of the action vectors is equal to a fixed value m) and efficiently implementable (i.e. there exist efficient algorithms for decomposition, projection and approximate barycentric spanner calculation). For all settings we consider here, except for paths, efficient decomposition and projection algorithms are proposed in [SHK+12]. For paths in DAGs the path polytope has a succinct description which allows efficient decomposition and projection with interior point methods (see [BV04], page 545]). Efficient calculation of an approximate barycentric spanner can be achieved if one has access to an efficient linear minimization oracle (LMO) over the action set $\mathcal{A}$ (see Proposition F.2), thus for the settings of interest it remains to show that such an LMO exists.

- **m-sets:** The most basic instance of combinatorial bandits are m-sets, where the decision maker at each timestep selects a subset of m out of d items. In binary representation the action space is defined as $\mathcal{A} = \{M \in \{0,1\}^d \mid \sum_i M_i = m\}$. In this setting it trivially holds that $|M|_1 = m$ and the LMO greedily chooses the top m arms with the lowest cost.

- **Spanning Trees and Graph Matroid Bandits (k-forests):** We examine an online decision-making problem where, at each timestep, the decision maker selects a spanning tree from an undirected graph $G = (V, E)$. This type of problem is relevant in certain mobile communication networks, where a minimum-cost subnetwork must be chosen in each timestep

to ensure the network remains connected. Formally, the action space is defined as follows: $\mathcal{A} = \{M \in [0,1]^{|E|} \mid \text{the set of edges} \{e \mid M_e = 1\} \text{ forms a spanning tree of } G\}$. We observe that for all $M \in \mathcal{A}$ it holds that $|M|_1 = |V| - 1$. As for the LMO, it can be efficiently implemented by minimum spanning tree algorithms, such as Kruskal's algorithm.

Generalizing the previous setting, let $\mathcal{A}$ denote the set of k-forests in a graph $G = (V, E)$, that is $\mathcal{A} = \{M \in [0,1]^{|E|} \mid \text{the set of edges} \{e \mid M_e = 1\} \text{ does not contain a cycle in } G \text{ and } |M|_1 = k\}$. It is known that $\mathcal{A}$ is a bases family of a truncation of a graphic matroid. Online learning in such action spaces has been studied in previous works (e.g. in [KWA+14]). For the LMO, Kruskal's algorithm with slight adaptations works.

- **Permutations and Truncated Permutations:** Consider the complete bipartite graph $K_{m,m}$, and let $\mathcal{A}$ contain all perfect matchings, that is $\mathcal{A} = \{M \in [0,1]^{m \times m} \mid \sum_{j=1}^{m} M_{i,j} = 1 \text{ for all } i = 1, \ldots, m \text{ and } \sum_{i=1}^{m} M_{i,j} = 1 \text{ for all } j = 1, \ldots, m\}$. Note that $\mathcal{A}$ is also equivalent to the set of all permutations of m items. Online optimization over permutations models problems such as ranking of search results or matching workers to tasks. The setting can be generalized to selecting truncated permutations of k out of m elements. Equivalently, a truncated permutation is a maximal matching in the complete bipartite graph between $[k]$ and $[m]$. Search query results typically take the form of a truncated permutation, where only the top k results, out of m available ones, are displayed in decreasing order of relevance. In binary representation, truncated permutations of k out of m items satisfy the condition $|M|_1 = k$ and for the LMO efficient algorithms for Maximum Bipartite Matching can be used. One simple option is the Ford-Fulkerson algorithm.

- **Paths:** Perhaps the most important setting in combinatorial bandits are path planning problems. We consider a directed acyclic graph $G = (V, E)$ with a source node s and a sink t where at each timestep the decision maker selects a path from s to t. In binary representation, each action vector M is the incidence vector of a path. To fit the requirements of ComBCP, all action vectors M should have the same $L1$ norm, that is all paths should have the same length. Of course this property does not hold in an arbitrary DAG, however, as shown in [GLLO07], it is possible to transform any DAG by adding at most $(K-2)(|V|-2)+1$ vertices and edges (with constant weight zero) and ensure that in the new DAG all paths from s to t have a length of K, where K is the maximal path length in the original graph. This way, learning paths in the transformed DAG is equivalent to learning paths in the original DAG, and the condition $|M|_1 = K$ is met. The LMO can be implemented with dynamic programming with running time linear in $|V| + |E|$.

H.2 Applications of Swap-ComBand

Next we present some applications where Swap-ComBand is efficiently implementable. The implementation of Swap-ComBand requires efficient sampling from the MWU distribution, the exploration policy μ and the uniform distribution over $\mathcal{A}$ and efficient calculation of the co-occurrence matrix. In many settings these routines can be efficiently implemented and outperform those used in Swap-ComBCP. Moreover, in contrast to Swap-ComBCP, Swap-ComBand does not require that all action vectors M have the same $L1$ norm, which in some applications can be restrictive.

Paths: The main application we consider is online path planning in DAGs. As we saw in the previous subsection Swap-ComBCP can be applied in this setting (after transforming the DAG to meet the requirement that all s-t paths have the same length). However, in each iteration Swap-ComBCP will need to perform the costly operation of projection on the path polytope. A standard approach for this step would be using interior point methods (see [BV04], page 545]), which would take $(|E|+|V|)^3$ time. In Swap-ComBand, the co-occurrence matrix calculation takes $\mathcal{O}(|E|^2)$ time using the techniques of [TW03] and sampling from MWU takes $\mathcal{O}(|E|)$ time using weight pushing [TW03]. Sampling from the uniform distribution over $\mathcal{A}$ takes $\mathcal{O}(|E|+|V|)$ time. For the exploration policy μ we can use a barycentric spanner, similarly to Swap-ComBCP. Thus, in path planning problems over DAGS Swap-ComBand can be efficiently implemented and its per iteration complexity is improved compared to Swap-ComBCP.

Path planning problems over DAGs can also model other classic combinatorial bandit settings, such as **m-sets** (see [CBL12]).