One Arrow, Two Kills: An Unified Framework for Achieving Optimal Regret Guarantees in Sleeping Bandits

Pierre Gaillard, Aadirupa Saha, Soham Dan

One Arrow, Two Kills: An Unified Framework for Achieving Optimal Regret Guarantees in Sleeping Bandits

Pierre Gaillard * Aadirupa Saha† Soham Dan‡

Abstract

We address the problem of *'Internal Regret'* in *Sleeping Bandits* in the fully adversarial setup, as well as draw connections between different existing notions of sleeping regrets in the multiarmed bandits (MAB) literature and consequently analyze the implications: Our first contribution is to propose the new notion of *Internal Regret* for sleeping MAB. We then proposed an algorithm that yields sublinear regret in that measure, even for a completely adversarial sequence of losses and availabilities. We further show that a low sleeping internal regret always implies a low external regret, and as well as a low policy regret for iid sequence of losses. The main contribution of this work precisely lies in unifying different notions of existing regret in sleeping bandits and understand the implication of one to another. Finally, we also extend our results to the setting of *Dueling Bandits* (DB)–a preference feedback variant of MAB, and proposed a reduction to MAB idea to design a low regret algorithm for sleeping dueling bandits with stochastic preferences and adversarial availabilities. The efficacy of our algorithms is justified through empirical evaluations.

1 Introduction

The problem of online sequential decision-making in standard K-armed multiarmed bandit (MAB) is well studied in machine learning [4, 45] and used to model online decision-making problems

under uncertainty. Due to their implicit exploration-vs-exploitation tradeoff, bandits are able to model clinical trials, movie recommendations, retail management job scheduling etc., where the goal is to keep pulling the 'best-item' in hindsight through sequentially querying one item at a time and subsequently observing a noisy reward feedback of the queried arm [14, 5, 4, 1, 10]. However, from a practical viewpoint, the decision space (or arm space $\mathcal{A} = \{1, \dots, K\}$) often changes over time due to unavailability of some items: For example, some items might go out of stock in a retail store, some websites could be down, some restaurants might be closed etc. This setting is studied in the multiarmed bandit (MAB) literature as *sleeping bandits* [22, 26, 20, 19], where at any round the set $S_t \subseteq \mathcal{A}$ of available actions could vary stochastically [26, 13] or adversarially [19, 23, 20]. Over the years, several lines of research have been conducted for sleeping multi-armed bandits (MAB) with different notions of regret performance, e.g. policy, ordering, or sleeping external regret [8, 26, 39].

In this paper, we introduce a new notion of sleeping regret, called *Sleeping Internal Regret*, that helps to bridge the gaps between different existing notions of sleeping regret in MAB. We show

*Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France. pierre.gaillard@inria.fr

†Toyota Technological Institute at Chicago (TTIC), US; aadirupa@ttic.edu

‡IBM Research, US; soham.dan@ibm.com (Major part of the work was done while the author was at the University of Pennsylvania)

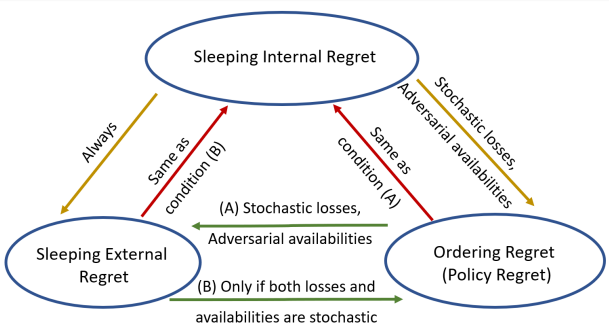

Figure 1: One Arrow, Two Kills: The connections between our proposed notion of Sleeping Internal Regret and different existing notions of regret for sleeping MAB and their implications

that our regret notion can be applied to the fully adversarial setup, which implies sleeping external regret in the fully adversarial setup (i.e. when both losses and item availabilities are adversarial), as well as policy regret in the stochastic setting (i.e. when losses are stochastic). We further propose an efficient $O(\sqrt{T})$ worst-case regret algorithm for sleeping internal regret. Finally we also motivate the implication of our results for the *Dueling Bandits* (DB) framework, which is an online learning framework that generalizes the standard multiarmed bandit (MAB) [5] setting for identifying a set of 'good' arms from a fixed decision-space (set of items) by querying preference feedback of actively chosen item-pairs [49, 2, 52, 35, 36]. The main contributions can be listed as follows:

• **Connecting Existing Sleeping Regret.** The first contribution (Sec. 2) lies in relating the existing notions of sleeping regret given as:

– The first one, *sleeping external regret*, is mostly used in prediction with expert advice [8, 16]. If the learner had played j instead of k_t at all rounds where j was available, we want the learner to not incur large regret. It is well-used to design dynamic regret algorithms [28, 51, 11, 50, 46]. It has the advantage that efficient no-regret algorithms can be designed even when both S_t and losses ℓ_t are adversarial.

– The second one, called *ordering regret*, is mostly used in the bandit literature [23, 39, 21, 27]. It compares the cumulative loss of the learner, with the one of the best ordering σ^* that selects the best available action according to σ^* at every round. No efficient algorithm exists when both ℓ_t and S_t are adversarial: either S_t or ℓ_t should be i.i.d [23].

– We also note that in some works, policies π^* (i.e., functions from subsets of $[K]$ to $[K]$) are considered instead of orderings σ^*, termed as *policy regret* [26, 39]. The latter two are equivalent when the losses are i.i.d., or come from an oblivious adversary with stochastic sleeping.

• **General Notion of Sleeping Regret.** Our second and one of the primary contribution lies in introducing a new notion of sleeping regret, called *Internal Sleeping Regret* (Definition 1), which we show actually unifies the different notions of sleeping regret under a general umbrella (see Fig. 1): We show that (i) Low sleeping internal regret always implies a low sleeping external regret, even under fully adversarial setup. (ii) For stochastic losses is also implies a low ordering regret (equivalently policy regret), even under adversarial availabilities. *Thus we now have a tool,* Sleeping Internal Regret*, optimizing which can simultaneously optimize all the existing notions of sleeping regret (and justifies the title of this work too!)* (Sec. 2.3).

• **Algorithm Design and Regret Implications.** The main contribution of this works is to

propose an efficient algorithm (SI-EXP3, Alg. 1) w.r.t. *Sleeping Internal Regret*, and design an $O(\sqrt{T})$ regret algorithm (Thm. 4). As motivated, the generalizability of our regret further implies $O(\sqrt{T})$ external regret at any setting and also ordering regret for i.i.d losses Rem. 3. We are the first to achieve this regret unification with only a single algorithm (Sec. 3).

• **Extensions: Generalized Regret for *Dueling-Bandits* (DB) and Algorithm.** Another versatility of *Internal Sleeping Regret* is it can be made useful for designing no-regret algorithms for the sleeping dueling bandits (DB) setup, which is a relative feedback based variant of standard MAB [52, 2, 7] (Sec. 4).

– **General Sleeping DB.** Towards this, we propose a new and more unifying notion of sleeping dueling bandits setup that allows the environment to play from different subsets of available dueling pairs ($A_t \subseteq [K]^2$) at each round t. This generalizes standard notion of DB setting where $A_t = [K]^2$ without sleeping, but also the setup of Sleeping DB for $A_t = S_t \times S_t$, [31].
– **Unifying Sleeping DB Regret.** Next, taking ques from our notion of *Sleeping Internal Regret* for MAB, we propose a generalized dueling bandit regret, *Internal Sleeping DB Regret* (Eq. (10)), which unifies the classical dueling bandit regret [52] as well as sleeping DB regret [31] (Rem. 4).
– **Optimal Algorithm Design.** Having established this new notion of sleeping regret in dueling bandits, we propose an efficient and order optimal $O(\sqrt{T})$ sleeping DB algorithm, using a reduction to MAB setup [32] (Thm. 5). This improves the regret bound of [31] that only get $O(T^{2/3})$ worst-case regret even in the simpler $A_t = S_t \times S_t$ setting.

• **Experiments.** Finally, in Sec. 5, we corroborate our theoretical results with extensive empirical evaluation (see Sec. 5). In particular, our algorithm significantly outperforms baselines as soon as there is dependency between S_t and ℓ_t. Experiments also seem to show that our algorithm can be used efficiently to converge to Nash equilibria of two-player zero-sum games with sleeping actions (see Rem. 5).

Related Works. The problem of regret minimization for stochastic multiarmed bandits (MAB) is widely studied in the online learning literature [5, 1, 25, 3], and as motivated above, the problem of item non-availability in the MAB setting is a practical one, which is studied as the problem of *sleeping MAB* [22, 26, 20, 19], for both stochastic rewards and adversarial availabilities [19, 23, 20] as well as adversarial rewards and stochastic availabilities [22, 26, 13]. In case of stochastic rewards and adversarial availabilities the achievable regret lower bound is known to be $\Omega(\sqrt{KT})$, K being the number of actions in the decision space $\mathcal{A} = [K]$. The well studied EXP4 algorithm does achieve the above optimal regret bound, although it is computationally inefficient [23, 19]. The optimal and efficient algorithm for this case is by [39], which is known to yield $\tilde{O}(\sqrt{T})$ regret,[1].

On the other hand over the last decade, the relative feedback variants of stochastic MAB problem has seen a widespread resurgence in the form of the Dueling Bandit problem, where, instead of getting noisy feedback of the reward of the chosen arm, the learner only gets to see a noisy feedback on the pairwise preference of two arms selected by the learner [52, 53, 24, 47, 40, 38, 30, 41], or even extending the pairwise preference to subsetwise preferences [44, 9, 33, 36, 37, 17, 29].

Surprisingly, there has been almost no work on dueling bandits in sleeping setup, despite the huge practicality of the problem framework. In a very recent work, [31] attempted the problem of Sleeping DB for the setup of stochastic preferences and adversarial availabilities, however there

[1] $\tilde{O}(\cdot)$ notation hides the logarithmic dependencies.

proposed algorithms can only yield a suboptimal regret guarantee of $O(T^{2/3})$. Our work is the first to achieve $\tilde{O}(\sqrt{T})$ regret for Sleeping Dueling Bandits (see Thm. 5).

2 Problem Formulation

In this section, we introduce problem of sleeping multiarmed bandit formally, followed by the definition of *Internal Sleeping Regret* – a new notion of learner's performance in sleeping MAB (Sec. 2.3). The last part of this section discusses the different notions of existing regret bounds in Sleeping MAB (Sec. 2.1) and their connections (Sec. 2.2, summarized in Fig. 1).

Problem Setting: Sleeping MAB. Let $[K] = \{1, \dots, K\}$ be a set of arms. At each round $t \geq 1$, a set of available arms $S_t \subseteq [K]$ is revealed to a learner, that is asked to select an arm $k_t \in S_t$, upon which the learner gets to observe the loss $\ell_t(k_t)$ of the selected arm. Note the sequence of item-availabilities $\{S_t\}_{t=1}^T$ as well as the loss sequence $\{\ell_t\}_{t=1}^T$ can be stochastic or adversarial (oblivious) in nature. We consider the hardest setting of adversarial losses and availabilities, which clearly subsumes the other settings as special cases (see Sec. 2.3 for details).

The next thing to understand is how should we evaluate the learner or what is the final objective? Before proceeding to our unifying notion of *Sleeping MAB regret, let us do a quick overview of existing notions of sleeping MAB regret studied in the prior bandit literature.*

2.1 Existing Objectives for Sleeping MAB

1. External Sleeping Regret. The first notion was introduced by [8]. Here, the learner is compared with each arm, only on the rounds in which the arm is available:

$$R_T^{\text{ext}}(k) := \sum_{t=1}^{T} \left(\ell_t(k_t) - \ell_t(k)\right) \mathbb{1}\{k \in S_t\}\,. \tag{1}$$

The learner is asked to control $\max_{k\in[K]} R_T(k) = o(T)$ as $T \to \infty$. In [8], the authors provide an algorithm which achieves $R_T(k) \leq O(\sqrt{T})$ for all k.

2. Ordering Regret. This second notion compares the performance of the learner on all rounds, with any fixed ordering $\sigma = (\sigma_1, \dots, \sigma_K) \in \Sigma$ of the arms, where Σ denotes the set of all possible orderings of $[K]$:

$$R_T^{\text{ordering}}(\sigma) := \sum_{t=1}^{T} \ell_t(k_t) - \ell_t(\sigma(S_t))\,, \tag{2}$$

where $\sigma(S_t) = \{\sigma_k \text{ s.t. } k = \text{argmin}\{i : \sigma_i \in S_t\}\}$ denotes the best arm available in S_t. Consequently, in this case, the learner's regret is evaluated is evaluated against the best ordering $\max_{\sigma\in\Sigma} R_T^{\text{ordering}}(\sigma)$.

It is known that no polynomial time algorithm can achieve a sublinear regret without stochastic assumptions on the losses ℓ_t or the availabilities S_t, as the problem is known to be NP-hard when both rewards and availabilities are adversarial [23, 20, 19]. For adversarial losses and i.i.d. S_t (where each arm is independently available according to a Bernoulli distribution), [39] proposed an algorithm with $O(\sqrt{T})$ regret. For i.i.d. losses and adversarial availabilities, a UCB based algorithm with logarithmic regret was proposed in [23].

3. Policy Regret A policy $\pi : 2^{[K]} \mapsto [K]$ denotes here a mapping from a set of available actions/experts to an item. Let $\Pi := \{\pi \mid 2^{[K]} \mapsto [K]\}$ be the class of all policies. In this case, the regret of the learner is measured against a fixed policy π is defined as:

$$R_T^{\text{policy}}(\pi) = \mathbb{E}\Big[\sum_{t=1}^{T} \ell_t(i_t) - \sum_{t=1}^{T} \ell_t(\pi(S_t))\Big], \tag{3}$$

where the expectation is taken w.r.t. the availabilities and the randomness of the player's strategy [39]. As usual, in this case, the learner's regret is evaluated is evaluated against the best policy $\max_{\pi\in\Pi} R_T^{\text{policy}}(\pi)$.

2.2 Relations across Different Notions of Existing Sleeping MAB Regret

One may wonder how these above notions are related. Is one stronger than the other? Or does optimizing one implies optimizing the other? Under what assumptions on the sequence of losses $\{\ell_t\}_{t\in[T]}$ and availabilities $\{S_t\}_{t\in[T]}$? We answer all these questions in this section and also summarized them in Fig. 1.

1. Relationship between (ii) Ordering Regret and (iii) Policy Regret. These two notions are very close, in principle they are equivalent in all practical contexts where they can be controlled. Note for stochastic losses and availabilities, it is easy to see both are equivalent, i.e. $\max_{\sigma\in\Sigma} R_T^{\text{ordering}}(\sigma) = \max_{\pi\in\Pi} R_T^{\text{policy}}(\pi)$. In fact, even when either losses or the availabilities are stochastic, and losses are independent of the availabilities (which are the only settings in which algorithms exist for these notions), we can claim the same equivalence! See App. A.3 for a proof. Thus, *for the rest of this paper, we will only work with Ordering Regret* (R_T^{ordering}).

2. Relationship between (i) External Sleeping Regret and (ii) Ordering Regret.

• **Does Ordering Regret** (2) **Implies External Regret** (1)**?**

– **Case (i): Stochastic losses, Adversarial** S_t**:** Yes, in this case it does. Since losses are stochastic, let at any round t, $\mathbb{E}[\ell_t(i)] = \mu_i$ for all $i \in [K]$. Then,

$$\begin{aligned}
\mathbb{E}[R_T^{\text{ext}}(k)] &= \sum_{t=1}^{T} (\mu_{k_t} - \mu_k)\mathbb{1}\{k \in S_t\} \\
&\leq \sum_{t=1}^{T} (\mu_{k_t} - \mu_{k_t^*})\mathbb{1}\{k \in S_t\} \\
&\leq \sum_{t=1}^{T} (\mu_{k_t} - \mu_{k^*}) = \mathbb{E}[R_T^{\text{ordering}}(\sigma)]
\end{aligned}$$

where the first inequality simply follows by the definition of $k_t^* = \sigma^*(S_t)$, σ^* being the best ordering in the hindsight, and by noting that for i.i.d. losses for any $i \in S_t$, $\mu_i - \mu_{k_t^*} \geq 0$.

– **Case (ii): Adversarial losses, Stochastic** S_t**:** The implication does not hold in this case. We can construct examples to show that it is possible to have $\mathbb{E}[R_T^{ordering}(\sigma)] = 0$ but $\mathbb{E}[R_T^{ext}(k)] = O(T)$ for some $k \in [K]$ (see App. A). The key observation lies in making the losses ℓ_t dependent of availability S_t.

• **Does External Regret** (1) **Implies Ordering Regret** (2)**?** Clearly, this direction is not true in general, as indeed, it would otherwise contradict the hardness result for ordering regret: This is since minimizing ordering regret is known to be NP-Hard for adversarial ℓ_t and S_t [23], while one

can easily construct efficient $\tilde{O}(\sqrt{T})$ regret algorithms for external regret in the fully adversarial setup, e.g. even our proposed algorithm SI-EXP3 achieves that (see Rem. 3). Let us analyze in a more case by case basis:

– **Case (i) Stochastic losses, Adversarial S_t:** No! Even for i.i.d. losses the implication does not hold for adversarial sleeping. To see this, we consider the following counter example with three arms ($K = 3$). Assume that the arms incur constant losses $\ell_t(k) = k$ when they are available. During the first $T/2$ rounds, we set $S_t = \{1, 2\}$ so that the worst arm is unavailable and during the last $T/2$ rounds, the best arm is the one that is sleeping, i.e., $S_t = \{2, 3\}$. Then, an algorithm that selects the first arm for $t = 1, \ldots, T/2$ and the worst arm for $t = T/2 + 1, \ldots, T$ satisfies $R_T^{\text{ext}}(k) = 0$ for any $k \in [3]$. Yet, $R_T^{\text{ordering}}((1, 2, 3)) = T/2$ because the algorithm chooses the worst arm 3 instead of 2 when $S_t = \{2, 3\}$.
– **Case (ii) Adversarial losses, Stochastic S_t:** The implication is not true in this case as well. We can simply do the same counter-example by taking i.i.d. availability sets: $S_t = \{1, 2\}$ with probability $1/2$ and $S_t = \{2, 3\}$ otherwise.

This essentially shows ordering regret is a stronger notion of regret compared to external regret.

To summarize the above relations, precisely, we present them pictorially in Fig. 1. However, it is already hard to keep track of the relations of so any different notions of regret! An even more daunting task is, which one to work with? Does optimizing one, necessarily guarantee a low regret in another? we are seeking for a more general sleeping notion that would imply both. To solve this, we introduce thereafter a new notion of sleeping MAB regret that unifies the above notions of regret under a general umbrella.

2.3 Internal Sleeping Regret: A New Performance Objective for Sleeping MAB

The notion of *Internal Regret* was introduced in the theory of repeated games [15], and largely studied in online learning since then, see among other [12, 42, 43, 8]. Roughly, a small internal regret for some pair (i, j) implies at any round t, learner would not have regretted playing j, where she actually played arm-i instead. Drawing motivation, for our sleeping MAB setup, we define the notion as follows:

Definition 1 (Internal Sleeping Regret)**.** *For any pair of arms $(i, j) \in [K]^2$, the* internal sleeping regret *is*

$$R_T^{int}(i \to j) := \sum_{t=1}^{T} \big(\ell_t(k_t) - \ell_t(j)\big) \mathbb{1}\{i = k_t, j \in S_t\}. \tag{4}$$

Typically, optimizing $R_T^{\text{int}}(i \to j)$ implies, we want that if the learner had played j on all the rounds where he played i and j was available, he does not incur large regret. The strength of this notion is that it can be minimized efficiently (as detailed in Sec. 3) for general adversarial losses and availabilities which is the key behind our main results (see Rem. 3 and 4).

2.4 Generalizing Power of Internal Sleeping Regret

In this section, we discuss, how R_T^{int} generalizes the other existing notions of sleeping regret as discussed in Sec. 2.1.

1. Internal Regret vs External Regret We start by noting that following is a well-known result in the classical online learning setting [43] (without sleeping).

Lemma 2 (Internal Regret Implies External Regret Always[43]). *For any sequences (ℓ_t), (S_t), and any algorithm, $R_T^{ext}(k) = \sum_{i=1}^K R_T^{int}(i \to k)$ for all $k \in [K]$.*

The proof follows from the regret definitions. Thus any uniform upper-bound on the internal regret, implies the same bound for the external regret up to a factor K. The other direction is not true though!

2. Internal Sleeping Regret vs Ordering Regret

Lemma 3 (Internal Regret Implies Ordering (for stochastic Losses)). *Assume that the losses $(\ell_t)_{t\geq 1}$ are i.i.d.. Then, for any ordering σ, we have*

$$\mathbb{E}\big[R_T^{ordering}(\sigma)\big] \leq \sum_{i=1}^{K} \sum_{j \in D_i} \mathbb{E}\big[R_T^{int}(i \to j)\big] \,.$$

where D_i is the set of arms such that $\mathbb{E}[\ell_t(j)] \leq \mathbb{E}[\ell_t(i)]$.

Therefore, any algorithm that satisfies $\mathbb{E}\big[R_T^{\text{int}}(i \to j)\big] \leq O(\sqrt{T})$, also satisfies $\mathbb{E}\big[R_T^{\text{ordering}}(\sigma)\big] \leq O(\sqrt{T})$. The proof is deferred to the App. A.4.

Remark 1. *An interesting research direction for the future would be to see if the sleeping internal regret also implies the ordering regret for adversarial losses and stochastic availabilities? Our experiments seem to point into this direction (see App. D), but despite our efforts, we could not prove it. Such a result would in particular imply that any algorithm that can achieve sublinear regret w.r.t. sleeping internal regret R_T^{int} (we in fact proposed such an algorithm in Sec. 3, see SI-EXP3), would actually satisfy a best-of-both worlds guarantee! That is if either the losses or the availabilities are stochastic, the algorithm it will in turn incur a sublinear regret w.r.t. ordering regret $R_T^{ordering}$ as well.*

Remark 2. *On the other hand, it is well known that for adversarial losses, a small external regret does not imply a small internal regret [43] even when $S_t = [K]$. But, when losses are stochastic and availabilities are adversarial, minimizing the ordering regret does control the internal regret (see App. A.4.*

3 SI-EXP3: An Algorithm for Minimizing Internal Sleeping Regret

We now consider the problem setting of Sleeping MAB Sec. 2 and propose an EXP3 based algorithm that is shown to yield sublinear sleeping regret guarantee. It is worth noting that, our algorithm applies to the hardest setting of adversarial losses and availabilities, which clearly subsumes the stochastic settings (losses or availabilities) as a special case: As proved in Thm. 4, our proposed algorithm SI-EXP3 achieves $\tilde{O}(\sqrt{KT})$ internal regret for any arbitrary sequence of losses $\{\ell_t\}_{t\in[T]}$ and availabilities $\{S_t\}_{t\in[T]}$. Further the generalizability of our internal regret (see Fig. 1) implies $O(\sqrt{T})$ external regret at any setting and also ordering regret for i.i.d losses, as detailed in Rem. 3.

Our regret analysis is inspired from the construction of [42] (see Section 3, Thm. 3.2) for the internal regret although the 'sleeping component' or item non-avilabilities is not considered. Another relevant work is [8], which designs an algorithm minimizing a variant of internal sleeping regret with

a subtle difference: it considers time selection functions $I \in \mathcal{I} \subseteq \{0,1\}^T$ instead of sleeping actions. More precisely, the regret considered by [8] is of the form

$$\max_{I \in \mathcal{I}} \sum_{t=1}^{T} \left(\ell_t(k_t) - \ell_t(j)\right) \mathbb{1}\{k_t = i, j \in I(t)\}\,.$$

Thus $R_T^{\mathrm{int}}(i \to j)$ would correspond to the choice $I(t) = \mathbb{1}\{j \in S_t\}$, but the dependence on the arm j is not possible in their definition and makes the adaptation of their algorithm challenging. Furthermore, their regret bound (Thm. 18) only holds in the full information setting. It may be adapted to bandit feedback, but would yield a suboptimal regret $O(K\sqrt{TK\log K})$ (which is the internal regret upper-bound they obtain in Thm. 11 without the sleeping component) in comparison with Thm. 4. We now describe our main algorithm, SI-EXP3, for optimizing Internal Sleeping Regret.

3.1 Algorithm: SI-EXP3

The Sleeping-Internal-EXP3 (SI-EXP3) procedure is a two-level algorithm. At round $t \geq 1$, the master algorithm forms a probability vector $p_t \in \Delta_K$ over the arms, which is used to sample the played action $k_t \sim p_t$. The vector p_t is such that $p_t(i) = 0$ for any $i \notin S_t$. A subroutine, based on EXP3 [5], combines $K(K-1)$ sleeping experts indexed by $i \to j$, for $i \neq j$. Each expert aims to minimize the internal sleeping regret $R_T^{\mathrm{int}}(i \to j)$. We detail below how to construct p_t.

Algorithm 1 SI-EXP3: Sleeping Internal Regret Algorithm for MAB

input: Arm set: $[K]$, learning rate $\eta > 0$
init: $E := \{(i,j) \in [K]^2, i \neq j\}$

$\tilde{q}_1 \in \Delta_E$ uniform distribution on E
for $t = 1, 2, \ldots, T$ **do**
  Observe $S_t \subseteq [K]$ and define $q_t \in \Delta_E$ as in (8)
  Define $p_t \in \Delta_K$ by solving (9)
  Predict $k_t \sim p_t$ and observe $\ell_t(k_t)$
  Define $\widehat{\ell}_t(k) = \frac{\ell_t(k)}{p_t(k)} \mathbb{1}\{k = k_t\}$ for all $k \in [K]$
  for $(i,j) \in E$ **do**
    Define $p_t^{i \to j} \in \Delta_K$ as in (5)
    Define $\widehat{\ell}_t(i \to j)$ as in (6)
  end for
  Update $\tilde{q}_{t+1}(i \to j) \propto \tilde{q}_t(i \to j) e^{-\eta \widehat{\ell}_t(i \to j)}$
end for

For any $i \neq j$, we denote by $p_t^{i \to j} \in \Delta_K$ the probability vector that moves the weight of p_t from i to j,

$$p_t^{i \to j}(k) = \begin{cases} 0 & \text{if } k = i \\ p_t(i) + p_t(j) & \text{if } k = j \\ p_t(k) & \text{otherwise} \end{cases}\,. \tag{5}$$

As usually considered in adversarial multi-armed bandits, for any active arm $i \in S_t$, we define the associated estimated loss $\widehat{\ell}_t(i) := \ell_t(i)\mathbb{1}\{i = k_t\}/p_t(i)$. Furthermore, by abuse of notation, we also

define for any $i, j \in [K]$, $i \neq j$ the loss

$$\widehat{\ell}_t(i \to j) := \begin{cases} \sum_{k=1}^K p_t^{i \to j}(k) \widehat{\ell}_t(k) & \text{if } j \in S_t \\ \ell_t(k_t) & \text{otherwise} \end{cases} \tag{6}$$

The subroutine then computes the exponential weighted average of the experts $i \to j$, by forming the weights

$$\tilde{q}_t(i \to j) := \frac{\exp\Big(-\eta \sum_{s=1}^{t-1} \widehat{\ell}_s(i \to j)\Big)}{\sum_{i' \neq j'} \exp\Big(-\eta \sum_{s=1}^{t-1} \widehat{\ell}_s(i' \to j')\Big)} \,. \tag{7}$$

To avoid assigning mass from an active item i to an inactive $j \notin S_t$, the subroutine then normalizes those weights so that sleeping experts get 0 mass

$$q_t(i \to j) := \frac{\tilde{q}_t(i \to j) \mathbb{1}\{j \in S_t\}}{\sum_{i' \neq j'} \tilde{q}_t(i' \to j') \mathbb{1}\{j' \in S_t\}} \,. \tag{8}$$

Finally, the master algorithms forms $p_t \in \Delta_K$ by solving the linear system

$$p_t = \sum_{i \neq j} p_t^{i \to j} q_t(i \to j) \,. \tag{9}$$

The existence and the practical computation of such a p_t is an application of Lemma 3.1 of [42].

3.2 Regret Analysis of SI-EXP3

We now analyze the sleeping internal regret guarantee of Alg. 1 (Thm. 4), and also the implications to other notions of sleeping regret (Rem. 3).

Theorem 4. *Consider the problem of Sleeping MAB for arbitrary (adversarial) sequences of losses $\{\ell_t\}$ and availabilities $\{S_t\}$. Let $T \geq 1$ and $\eta^2 = (\log K)/(2\sum_{t=1}^T |S_t|)$. Assume that $0 \leq \ell_t(i) \leq 1$ for any $i \in S_t$ and $t \in [T]$. Then,*

$$\mathbb{E}\big[R_T^{int}(i \to j)\big] \leq 2\sqrt{2 \log K \sum_{t=1}^T |S_t|} \leq 2\sqrt{2TK \log K} \,,$$

for all $i \neq j$ in $[K]$.

The proof is postponed to App. B. The learning rate η can be calibrated online at the price of a small multiplicative constant by choosing a time-varying learning rate $\eta_t^2 = (\log K)/(2\sum_{s=1}^t |S_s|)$ or by using the doubling-trick technique [12].

Remark 3. *The above theorem also implies a bound of order $O(K\sqrt{KT \log K})$ for the external sleeping regret by Lem. 2 and $O(K^2\sqrt{KT \log K})$ for the ordering regret when the losses are i.i.d. by Lem. 3. SI-EXP3 is the first to simultaneously achieve order-optimal* sleeping external regret *for fully adversarial setup as well as* ordering regret *for stochastic losses and this was possible owning to the versatility of* Sleeping Internal Regret *as summarized in Fig. 1.*

4 Implications: Better Regret for Sleeping Dueling Bandits

In this section, we show the implication of our result to sleeping dueling bandits.

Motivation behind a generalized DB objective. We show interesting use-cases of our generalization such as when the user is asked at each round to choose two different actions $i \neq j$. Note that in the dueling bandit literature, the user may choose replicated arms (i, i), and is expected to converge to an optimal pair (i^*, i^*). However, in many applications, this does not make sense to show users the same pair of items (i, i), rather it might be preferred to see their top-two choices, i.e we would expect the algorithm to converge to the best pair (i, j), $i \neq j$ (more motivating examples are provided after Rem. 4). Classical dueling bandit algorithms do not easily allow for such a restriction, whereas this can be easily achieved with our sleeping procedure.

4.1 Our Problem Setting: Sleeping Dueling Bandits (Sleeping DB)

We generalize the setting of for dueling bandits with adversarial sleeping of [31]. We consider the stochastic dueling bandit setting with a preference matrix $P \in [0,1]^{K\times K}$ such that $P(i,j) = 1-P(i,j)$ is the probability of item i to beat item j in some round. Furthermore, we assume that their exists a total ordering of the arms σ, such that $P(\sigma(i), j) \geq P(\sigma(i'), j)$ for all $\sigma(i) \leq \sigma(i')$ and all $j \in [K]$. This is for instance satisfied for utility-based preference matrices [2], or for Plackett-Luce model [6], where it is assumed that the K items are associated to positive score parameters $\theta_1, \ldots, \theta_K$ and $P(i,j) = \theta_i/(\theta_i + \theta_j)$ for all $i, j \in [K]$.

Before each round $t \geq 1$, an adversary reveals a set of possible dueling pairs $A_t \subseteq [K]^2$. We assume that if $(i,j) \in A_t$ then $(j,i) \in A_t$. Furthermore, we denote for any $i \in [K]$ by $A_t(i) := \{j \in [K], (i,j) \in A_t\}$, the set of possible adversaries for $i \in [K]$, and as usual by $S_t := \{i \in [K], A_t(i) \neq \emptyset\}$ the set of available arms at time t. After observing A_t, the learner selects a pair of items $(i_t, j_t) \in A_t$ and observes the result of the duel $o_t(i_t, j_t)$ which follows a Bernoulli distribution with mean $P(i_t, j_t)$.

Performance: Internal Sleeping DB regret. We measure the learner's regret w.r.t. the following regret measure:

$$R_T^{\texttt{SI-DB}} = \frac{1}{2}\sum_{t=1}^{T}\Big(\max_{j^*\in A_t(i_t)} P(j^*, i_t) + \max_{i^*\in A_t(j_t)} P(i^*, j_t) - 1\Big). \tag{10}$$

Since the definition is inspired from internal regret, we term it as Internal Sleeping DB regret (or `SI-DB` in short) — the measure essentially evaluates the dueling choices of the learner (i_t, j_t) against their best 'available competitor' according to $A_t(\cdot)$.

Remark 4 (Generalizability of $R_T^{\texttt{SI-DB}}$)**.** *It is noteworthy that if all pairs are available $A_t = [K]^2$, then $i_t^* = j_t^*$ is the Condorcet Winner (CW) (see [52] for definition) for all rounds since $\mathbb{P}$ respects total ordering. Thus, in this case, $R_T^{SI\text{-}DB}$ reduces to the standard CW-regret studied in DB [48, 52, 7]. Moreover, $R_T^{SI\text{-}DB}$ also generalizes the notion of Sleeping Dueling Bandit of [31] for the special case $A_t = S_t \times S_t$ (i.e. when all pairs of the available items are feasible): This is since for this case we again have $i_t^* = j_t^*$ (owing to the total ordering assumption of $\mathbb{P}$), and hence we can recover their notion of sleeping regret (see Eqn. (1) of [31]). Nevertheless, our new notion offers more flexibility as we now show with some application examples.*

Motivating Examples: Practicability of $R_T^{\texttt{SI-DB}}$.

(i.) Dueling bandits with non-repeating arms. A first example consists in choosing $A_t = \{(i,j) \in [K]^2, i \neq j\}$. Our algorithm will be forced to play two different items at every round. Such a restriction is new to dueling bandits although it makes sense in many applications, such as recommendation systems in which we may want to suggest pairs of different items. Our algorithm will converge to the best pair. An interesting question for future work is to generalize our strategy to any size M (possibly larger than 2) of subsets of unique battling items. A similar setting was considered by [34] but they allow choosing replicate items.

(ii). Preference learning with categories. Another example comes from an application in which the arms may be grouped into different categories (or teams). For example, one may thing of a recommendation system for movies. The latter could be action movies, documentaries, TV series, or romantic movies. The system could be asked to suggest at every round movies from different categories to provide diversity into the suggestion. Our algorithm would simultaneously learn the best movies but also the best two categories. In addition, the possibility of sleeping allows the film collection to vary over time.

4.2 Sparring SI-EXP3: An Algorithm for Sleeping DB and Regret Analysis

Following the generic reduction from multi-armed bandit to dueling bandit from [32] (see Section 4), we consider the following algorithm. We run two versions $\mathcal{A}^{\text{left}}$ and $\mathcal{A}^{\text{right}}$ of the internal sleeping regret algorithm of Thm. 4 in parallel. At each round t, i_t is chosen by $\mathcal{A}^{\text{left}}$, which is run on the availability sets S_t and losses $\ell_t^{\text{left}}(k) = o_t(j_t, k)$, $k \in S_t$. After i_t is chosen, $\mathcal{A}^{\text{right}}$ chooses j_t, by using the availability sets $A_t(i_t)$ and losses $\ell_t^{\text{right}}(k) = o_t(i_t, k)$. We call the algorithm as *Sparring SI-EXP3*, following the classical nomenclature from [2] which first invented the idea of designing a DB algorithm by making two MAB algorithms competing against another, and famously termed it as '*Sparring*'.

Theorem 5. *Consider the problem setting of Sleeping DB defined above (Sec. 4) and let $T \geq 1$. Then, Sparring SI-EXP3 satisfies*

$$\mathbb{E}[R_T^{\texttt{SI-DB}}] \leq 2K^2\sqrt{2TK\log K}\,.$$

The proof is postponed to App. C. As explained in Rem. 4, by choosing A_t of the form $S_t \times S_t$ for some subset $S_t \subseteq [K]$, we retrieve the setting of [31]. Note that they provide distribution dependent upper-bounds while we present worst-case upper-bound. They show a high-probability regret bound of order $O(K^2\log(1/\delta)/\Delta^2)$ for an UCB based algorithm, and a $O(K^3/\Delta^2 + K^2\log(T)/\Delta)$ upper-bound on the expected regret of an algorithm based on empirical divergences. Their analysis are quite technical and non-trivial to adapt to general sets A_t as our result. Furthermore , both their algorithms yield a worst-case regret of order $O(T^{2/3})$ while we only suffer $O(\sqrt{T})$.

5 Experiments

We provide synthetic experiments in sleeping multi-armed bandits. In all the experiments, we plot the policy regret R_T^{policy} for MAB (3). All experiments are averaged across 50 runs. Further experiments (including some in the dueling setup) are provided in App. D. We compare the results of the following algorithms:

- SI-EXP3: Alg. 1;
- S-UCB: A sleeping UCB procedure [23] for ordering regret with stochastic losses;

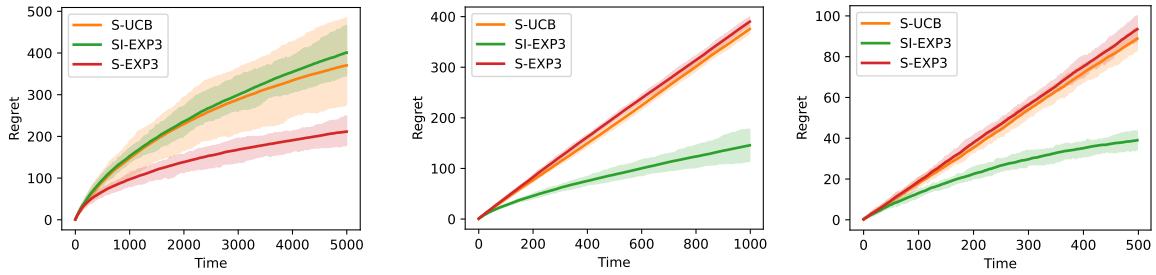

Figure 2: [Left] Stochastic environment [Middle] Dependent environment [Right] Rock Paper Scissors

• S-EXP3: Sleeping-EXP3G [39] for ordering regret with adversarial losses and i.i.d. sleeping.

We compare to these two algorithms because they achieve state-of-the-art performance in their respective settings. The hyper-parameters η of SI-EXP3 and (η, λ) of S-EXP3 are considered as time-varying hyper-parameters and set to $t^{-1/2}$.

Stochastic environment. The losses and availabilies for $K = 10$ arms are i.i.d. and respectively follow Bernoulli distributions with mean μ_k and a_k. The latter are uniformly sampled at the start of each run on $(0, 1)$. Rounds with no available arms are skipped.

Dependent environment. We consider the following semi-stochastic environment with $K = 3$. The pairs (S_t, ℓ_t) are still i.i.d. but the losses ℓ_t depend on the availabilities. The sets S_t are first uniformly sampled among $\{1, 2\}, \{1, 2, 3\}, \{1, 3\}$ and $\{2, 3\}$. According to the values of S_t, the loss vectors are then respectively $(0, .5, x), (0, .5, 1), (1, x, 0)$, and $(x, 0, 1)$, where x means that the arm is sleeping.

Rock-Paper-Scissors. We consider a repeated two-player zero-sum game with payoff matrix

$$P = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix}.$$

We assume that at the start of each round some action may be unavailable (S_t is uniformly sampled as in the previous environment). The game is then played on S_t only. We consider an opponent that is playing the Nash equilibrium of the sub-games (i.e., the game with the payoff matrix P restricted to S_t) and run each algorithm against that opponent.

Results. The cumulative regrets are provided in Fig. 2. We find that as soon as there are dependencies between the loss vectors and availabilities, SI-EXP3 significantly outperforms the other two algorithms. This is not surprising: S-EXP3 and S-UCB were indeed designed to perform well with respect to the best fixed ordering. Typically, they first order the actions with respect to their average performance on all rounds, and play the best action that is available in S_t. In case of dependency between S_t and ℓ_t, the best ordering may vary over the rounds and S-UCB and S-EXP3 incur linear regret. Note that this situation can happen often in real life. An example is an internet sales site that wants to offer products. Some products are useless if others are out of stock. For instance, it is less interesting to offer the products of a cooking recipe if some of the ingredients are not available.

Remark 5 (Application to game theory with sleeping actions)**.** *For sleeping two-player zero-sum games, the best policy to play depends on the actions available to the opponent: if Scissors is not available, then Paper is the best action, although when all actions are available the optimal policy is* $(1/3, 1/3, 1/3)$*. For instance, the Nash equilibrium of* P *restricted to* $S_t = \{1, 2\}$ *(Rock,*

Paper), is $(0, 1, 0)$ *(i.e., play Paper). Yet, here, all actions are on average equally good (i.e., taking the expectation over* S_t*); and S-UCB and S-EXP3 will converge to* $(1/2, 1/2, 0)$ *when Scissor is unavailable and incur linear regret. On the other hand, SI-EXP3 is able to leverage the dependence between* S_t *and* ℓ_t *and choose the right action. In App. D, we provide an additional experiment with two-player zero-sum randomized games (with a random payoff matrix* P*). An intriguing question for future work is whether SI-EXP3 converges to the Nash equilibrium of each subgame (or whether it obtains sublinear regret against an adversary that plays the Nash of* P *restricted to actions of* S_t*).*

Perspectives On a high level, the general theme of this work–to unify different notions of performance measure under a common umbrella and designing efficient algorithms for the general measure–can be applied to several other bandits/online learning/learning theory settings, which opens plethora of new directions.

Specifically as an extension to this work, some of the interesting open challenges could be: **(i).** to understand if sleeping internal regret also implies the ordering regret for adversarial losses but stochastic availabilities (see Rem. 1). **(ii).** to derive gap dependent bounds sleeping dueling bandit regret for stochastic preferences and adversarial sleeping, same as derived for its MAB counterpart in [23] or in a recent work [31] which though only gave suboptimal regret guarantees? **(iii).** to understand if our results can be extended to the subsetwise generalization of dueling bandits, studied as the *Battling Bandits* [35, 37, 29]; amongst many.

References

[1] Shipra Agrawal and Navin Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–1, 2012.

[2] Nir Ailon, Zohar Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *International Conference on Machine Learning*, pages 856–864. PMLR, 2014.

[3] Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. In *COLT-23th Conference on Learning Theory-2010*, pages 13–p, 2010.

[4] Peter Auer. Using upper confidence bounds for online learning. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 270–279. IEEE, 2000.

[5] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[6] Hossein Azari, David Parks, and Lirong Xia. Random utility theory for social choice. *Advances in Neural Information Processing Systems*, 25, 2012.

[7] Viktor Bengs, Róbert Busa-Fekete, Adil El Mesaoudi-Paul, and Eyke Hüllermeier. Preference-based online learning with dueling bandits: A survey. *Journal of Machine Learning Research*, 2021.

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

[9] Brian Brost, Yevgeny Seldin, Ingemar J. Cox, and Christina Lioma. Multi-dueling bandits and their application to online ranker evaluation. *CoRR*, abs/1608.06253, 2016.

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

[11] Nicolo Campolongo and Francesco Orabona. A closer look at temporal variability in dynamic online learning. *arXiv preprint arXiv:2102.07666*, 2021.

[12] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[13] Corinna Cortes, Giulia Desalvo, Claudio Gentile, Mehryar Mohri, and Scott Yang. Online learning with sleeping experts and feedback graphs. In *International Conference on Machine Learning*, pages 1370–1378, 2019.

[14] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(Jun):1079–1105, 2006.

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

[16] Pierre Gaillard, Gilles Stoltz, and Tim Van Erven. A second-order bound with excess losses. In *Conference on Learning Theory*, pages 176–196. PMLR, 2014.

[17] Suprovat Ghoshal and Aadirupa Saha. Exploiting correlation to achieve faster learning rates in low-rank preference bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 456–482. PMLR, 2022.

[18] Elad Hazan et al. Introduction to online convex optimization. arXiv:1909.05207, 2021.

[19] Satyen Kale, Chansoo Lee, and Dávid Pál. Hardness of online sleeping combinatorial optimization problems. In *Advances in Neural Information Processing Systems*, pages 2181–2189, 2016.

[20] Varun Kanade and Thomas Steinke. Learning hurdles for sleeping experts. *ACM Transactions on Computation Theory (TOCT)*, 6(3):11, 2014.

[21] Varun Kanade and Thomas Steinke. Learning hurdles for sleeping experts. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–16, 2014.

[22] Varun Kanade, H Brendan McMahan, and Brent Bryan. Sleeping experts and bandits with stochastic action availability and adversarial rewards. 2009.

[23] Robert Kleinberg, Alexandru Niculescu-Mizil, and Yogeshwer Sharma. Regret bounds for sleeping experts and bandits. *Machine learning*, 80(2-3):245–272, 2010.

[24] Junpei Komiyama, Junya Honda, Hisashi Kashima, and Hiroshi Nakagawa. Regret lower bound and optimal algorithm in dueling bandit problem. In *COLT*, pages 1141–1154, 2015.

[25] Tor Lattimore and Csaba Szepesvári. Bandit algorithms. *preprint*, 2018.

[26] Gergely Neu and Michal Valko. Online combinatorial optimization with stochastic decision sets and adversarial losses. In *Advances in Neural Information Processing Systems*, pages 2780–2788, 2014.

[27] Gergely Neu and Michal Valko. Online combinatorial optimization with stochastic decision sets and adversarial losses. *Advances in Neural Information Processing Systems*, 27, 2014.

[28] Anant Raj, Pierre Gaillard, and Christophe Saad. Non-stationary online regression. *arXiv preprint arXiv:2011.06957*, 2020.

[29] Wenbo Ren, Jia Liu, and Ness B Shroff. PAC ranking from pairwise and listwise queries: Lower bounds and upper bounds. *arXiv preprint arXiv:1806.02970*, 2018.

[30] Aadirupa Saha. Optimal algorithms for stochastic contextual preference bandits. *Advances in Neural Information Processing Systems*, 34:30050–30062, 2021.

[31] Aadirupa Saha and Pierre Gaillard. Dueling bandits with adversarial sleeping. *Advances in Neural Information Processing Systems*, 34, 2021.

[32] Aadirupa Saha and Pierre Gaillard. Versatile dueling bandits: Best-of-both-world analyses for online learning from preferences. *arXiv preprint arXiv:2202.06694*, 2022.

[33] Aadirupa Saha and Aditya Gopalan. Battle of bandits. In *Uncertainty in Artificial Intelligence*, 2018.

[34] Aadirupa Saha and Aditya Gopalan. Battle of bandits. In *UAI*, pages 805–814, 2018.

[35] Aadirupa Saha and Aditya Gopalan. Combinatorial bandits with relative feedback. In *Advances in Neural Information Processing Systems*, 2019.

[36] Aadirupa Saha and Aditya Gopalan. PAC Battling Bandits in the Plackett-Luce Model. In *Algorithmic Learning Theory*, pages 700–737, 2019.

[37] Aadirupa Saha and Aditya Gopalan. From pac to instance-optimal sample complexity in the plackett-luce model. In *International Conference on Machine Learning*, pages 8367–8376. PMLR, 2020.

[38] Aadirupa Saha and Akshay Krishnamurthy. Efficient and optimal algorithms for contextual dueling bandits under realizability. In *International Conference on Algorithmic Learning Theory*, pages 968–994. PMLR, 2022.

[39] Aadirupa Saha, Pierre Gaillard, and Michal Valko. Improved sleeping bandits with stochastic action sets and adversarial rewards. In *International Conference on Machine Learning*, pages 8357–8366. PMLR, 2020.

[40] Aadirupa Saha, Tomer Koren, and Yishay Mansour. Adversarial dueling bandits. In *International Conference on Machine Learning*, pages 9235–9244. PMLR, 2021.

[41] Aadirupa Saha, Tomer Koren, and Yishay Mansour. Dueling convex optimization. In *International Conference on Machine Learning*, pages 9245–9254. PMLR, 2021.

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

[43] Gilles Stoltz and Gábor Lugosi. Internal regret in on-line portfolio selection. *Machine Learning*, 59(1):125–159, 2005.

[44] Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Multi-dueling bandits with dependent arms. In *Conference on Uncertainty in Artificial Intelligence*, UAI'17, 2017.

[45] Joannes Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In *European conference on machine learning*, pages 437–448. Springer, 2005.

[46] Chen-Yu Wei, Yi-Te Hong, and Chi-Jen Lu. Tracking the best expert in non-stationary stochastic environments. *Advances in neural information processing systems*, 29, 2016.

[47] Huasen Wu and Xin Liu. Double Thompson sampling for dueling bandits. In *Advances in Neural Information Processing Systems*, pages 649–657, 2016.

[48] Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1201–1208. ACM, 2009.

[49] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

[50] Lijun Zhang, Tie-Yan Liu, and Zhi-Hua Zhou. Adaptive regret of convex and smooth functions. In *International Conference on Machine Learning*, pages 7414–7423. PMLR, 2019.

[51] Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Dynamic regret of convex and smooth functions. *Advances in Neural Information Processing Systems*, 33:12510–12520, 2020.

[52] Masrour Zoghi, Shimon Whiteson, Remi Munos, Maarten de Rijke, et al. Relative upper confidence bound for the k-armed dueling bandit problem. In *JMLR Workshop and Conference Proceedings*, number 32, pages 10–18. JMLR, 2014.

[53] Masrour Zoghi, Zohar S Karnin, Shimon Whiteson, and Maarten De Rijke. Copeland dueling bandits. In *Advances in Neural Information Processing Systems*, pages 307–315, 2015.

Supplementary: One Arrow, Two Kills: An Unified Framework for Achieving Optimal Regret Guarantees in Sleeping Bandits

A Appendix for Sec. 2

A.1 Low ordering regret $R_T^{\mathbf{ordering}}$ with Adversarial losses, Stochastic availabilities does not imply low external regret $R_T^{\mathbf{ext}}$

Lemma 6. *There exists a sequence of i.i.d. availabilities $(S_t)_{t\geq 1}$ and a sequence of losses $(\ell_t)_{t\geq 1}$ (possibly depending on S_t), such that, there exists an algorithm with*

$$\max_{\sigma} R_T^{ordering}(\sigma) = o(T) \qquad \textit{and} \qquad \max_{k} R_T^{ext}(k) = \Omega(T)\,,$$

as $T \to \infty$.

Proof. We provide the following example. Consider a MAB problem with 3 arms, $K = 3$. Suppose the problem encounters the following availability sets $\mathcal{A}_1 = \{1,2,3\}, \mathcal{A}_2 = \{1,2\},, \mathcal{A}_3 = \{1,3\}, \mathcal{A}_4 = \{2,3\}$ uniformly, i.e. $\mathbb{P}(\mathcal{S}_t = \mathcal{A}_i) = 1/4$ for all $i \in [4]$ and $t \in [T]$, where $\mathcal{S}_t$ being the availability set at time t. Further let us consider the adversarial (rather set dependent) loss sequence generated as follows:

	$\ell_t(1)$	$\ell_t(2)$	$\ell_t(3)$
if $\mathcal{S}_t = \mathcal{A}_1$	0	1	1
if $\mathcal{S}_t = \mathcal{A}_2$	0	1	x
if $\mathcal{S}_t = \mathcal{A}_3$	1	x	0
if $\mathcal{S}_t = \mathcal{A}_4$	x	1	1

where x can be any arbitrary loss value. For this example, the best orderings are $(1,2,3), (1,3,2)$ and $(3,1,2)$, that get a cumulative loss equals to $T/2$ in expectation. Indeed,

$$\frac{4}{T}\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\sigma(S_t))\right] = \begin{cases} 2 & \text{if } \sigma = (1,2,3) \\ 2 & \text{if } \sigma = (1,3,2) \\ 4 & \text{if } \sigma = (2,1,3) \\ 3 & \text{if } \sigma = (2,3,1) \\ 2 & \text{if } \sigma = (3,1,2) \\ 3 & \text{if } \sigma = (3,2,1) \end{cases}.$$

Consider and algorithm that plays $k_t = \sigma(S_t)$ according to the ordering $\sigma = (3,1,2)$. Then, $\mathbb{E}[\sum_{t=1}^{T} \ell_t(k_t)] = T/2$. It has thus no-regret $R_T^{\text{ordering}}(\sigma^*) \leq 0$ with respect to any ordering σ^*. Yet, its internal regret with respect to action 1 is

$$R_T^{\text{ext}}(1) = \mathbb{E}\left[\sum_{t=1}^{T} (\ell_t(k_t) - \ell_t(1)) \mathbb{1}\{1 \in S_t\}\right] = \frac{T}{4}.$$

This implies a 'no-regret ordering regret learner' does not imply a 'no-regret external regret learner' for any arbitrary sequence of adversarial losses, stochastic availabilities. □

A.2 Low ordering regret R_T^{ordering} with Stochastic losses, Adversarial availabilities does imply low internal regret R_T^{int}

Lemma 7. *Let $(\ell_t)_{t\geq 1}$ be an i.i.d. sequence of losses. Then, for any sequence of availability sets $(S_t)_{t\geq 1}$ such that S_t may only depend on $(\ell_s)_{s\leq t-1}$*

$$\max_{1\leq i,j\leq K} R_T^{int}(i\to j) \leq \max_{\sigma} R_T^{ordering}(\sigma)\,,$$

for any algorithm.

Proof. Consider stochastic losses such that $\ell_t(k)$ are i.i.d. with mean μ_k for all $k\in[K]$, and any sequence of availability sets $S_1,\dots,S_T\subseteq[K]$ (that can only depend on information up to $t-1$). Let σ^* be an optimal ordering

$$\sigma^* \in \operatorname*{argmin}_{\sigma} \mathbb{E}\left[\sum_{t=1}^{T}\ell_t(\sigma(S_t))\right].$$

Then, for all $S\subseteq[K]$, $\mu_{\sigma^*(S)} = \min_{i\in S}\mu_i$. Let k_t be the predictions of any algorithm. Let $(i,j)\in[K]^2$. Then,

$$\begin{aligned}
R_T^{\text{int}}(i\to j) &= \mathbb{E}\left[\sum_{t=1}^{T}(\ell_t(i)-\ell_t(j))\mathbb{1}\{i=k_t, j\in S_t\}\right]\\
&\leq \mathbb{E}\left[\sum_{t=1}^{T}(\mu_{k_t}-\mu_j)\mathbb{1}\{i=k_t, j\in S_t\} + (\mu_{k_t}-\mu_{\sigma^*(S_t)})\mathbb{1}\{k_t\neq i \text{ or } j\notin S_t\}\right]\\
&\leq \mathbb{E}\left[\sum_{t=1}^{T}\mu_{k_t}-\mu_{\sigma^*(S_t)}\right] = R_T^{\text{ordering}}(\sigma^*)\,,
\end{aligned}$$

where the inequalities are because $\mu_{\sigma^*(S_t)}\leq\mu_i$ for any $i\in S_t$. This concludes the proof. □

A.3 Equivalence of Policy and Ordering Regret

The policy regret is a stronger notion than ordering regret in general. From their definitions, we see

$$\max_{\sigma} R_T^{\text{ordering}}(\sigma) \leq \max_{\pi} R_T^{\text{policy}}(\pi)\,,$$

because for each ordering σ, one can associate a policy π, such that $\pi(S_t)=\sigma(S_t)$. But, the other direction is not true in general. Indeed, in the example of App. A.1, the inequality is strict. This is due to the dependence between losses and availabilities. Yet, no existing efficient algorithm can handle such dependence neither for policy regret nor for ordering regret. In this appendix, we prove that when either losses or availabilities are i.i.d. with no dependence, then the two notions are equivalent.

Lemma 8 (Stochastic losses and adversarial availabilities)**.** *Let $(\ell_t)_{t\geq 1}$ be an i.i.d. sequence of losses. Then, for any sequence of availability sets $(S_t)_{t\geq 1}$ such that S_t may only depend on $(\ell_s)_{s\leq t-1}$, then*

$$\max_{\pi} R_T^{policy}(\pi) = \max_{\sigma} R_T^{ordering}(\sigma)\,,$$

for any algorithm.

Proof. The proof follows from the observation that the best policy with i.i.d. losses is to play the available action with the smallest expected loss. Such a policy corresponds to the ordering $\mu_{\sigma_i} \leq \mu_{\sigma_j}$ for all $i \leq j$. Note that this would not be true if ℓ_t could depend on S_t. □

Lemma 9 (Adversarial oblivious losses and stochastic rewards)**.** *Let $(\ell_t)_{t\geq 1}$ be an arbitrary sequence of losses and $(S_t)_{t\geq 1}$ be a sequence of i.i.d. availability sets. Then,*

$$\max_{\pi} R_T^{policy}(\pi) = \max_{\sigma} R_T^{ordering}(\sigma)\,,$$

for any algorithm.

Proof. It is important to note here that we consider an oblivious adversary for the loss sequence (ℓ_t), which cannot depend on the randomness of (S_t). Let $\pi : 2^{[K]} \to [K]$ be a policy, then

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(\pi(S_t))\right] = \sum_{t=1}^{T} \sum_{S\in 2^{[K]}} \ell_t(\pi(S))\mathbb{P}(S = S_t) = \sum_{S\in 2^{[K]}} p(S) \sum_{t=1}^{T} \ell_t(\pi(S))$$

where $p(S) = \mathbb{P}(S_t = S)$. Thus, the best policy corresponds to the choice

$$\pi(S) \in \operatorname*{argmin}_{k\in S} \sum_{t=1}^{T} \ell_t(k)\,.$$

This policy corresponds to the ordering $\sum_{t=1}^{T} \ell_t(\sigma_i) \leq \sum_{t=1}^{T} \ell_t(\sigma_j)$ for $i \leq j$. □

A.4 Low internal regret R_T^{int} with Stochastic losses, Adversarial availabilities does imply low ordering regret R_T^{ordering}

Lemma 3 (Internal Regret Implies Ordering (for stochastic Losses))**.** *Assume that the losses $(\ell_t)_{t\geq 1}$ are i.i.d.. Then, for any sequence of availability sets $(S_t)_{t\geq 1}$ such that S_t may only depend on $(\ell_s)_{s\leq t-1}$, for any ordering σ, we have*

$$\mathbb{E}[R_T^{ordering}(\sigma)] \leq \sum_{i=1}^{K} \sum_{j\in D_i} \mathbb{E}[R_T^{int}(i \to j)]\,,$$

where D_i is the set of arms such that $\mathbb{E}[\ell_t(j)] \leq \mathbb{E}[\ell_t(i)]$.

Proof. Let $\mu_k = \mathbb{E}[\ell_t(k)]$ for all $k \in [K]$. Let σ^* be the best ordering such that $\mu_{\sigma_1^*} \leq \mu_{\sigma_2^*} \leq \cdots \leq \mu_{\sigma_K^*}$. Note that for any ordering σ, we have $\mathbb{E}[R_T^{\text{ordering}}(\sigma)] \leq \mathbb{E}[R_T^{\text{ordering}}(\sigma^*)]$. Thus, we can restrict ourselves to σ^*. Denote by $k_t^* := \sigma^*(S_t)$, the best available item in S_t. For any i, we also define by $D_i := \{j \in [K] : \mu_j \leq \mu_i\}$ the items that are better than i. Then,

$$\begin{aligned}
\mathbb{E}[R_T^{\text{ordering}}(\sigma)] &:= \mathbb{E}\Big[\sum_{t=1}^{T} \ell_t(k_t) - \ell_t(\sigma(S_t))\Big] = \mathbb{E}\Big[\sum_{t=1}^{T} \mu_{k_t} - \mu_{k_t^*}\Big] \\
&= \mathbb{E}\Big[\sum_{t=1}^{T}\sum_{i=1}^{K}\sum_{j\in D_i} (\mu_i - \mu_j)\mathbb{1}\{i = k_t, j = k_t^*\}\Big] \quad \leftarrow k_t^* \in D_i \text{ because it is the best item in } S_t
\end{aligned}$$

$$\leq \mathbb{E}\Big[\sum_{t=1}^{T}\sum_{i=1}^{K}\sum_{j\in D_i}(\mu_i-\mu_j)\mathbb{1}\{i=k_t, j\in S_t\}\Big] \quad \leftarrow \text{because } k_t^*\in S_t \text{ and } \mu_i-\mu_j\geq 0 \text{ for any } j\in D_i$$

$$\leq \sum_{i=1}^{K}\sum_{j\in D_i}\mathbb{E}\big[R_T^{\mathrm{int}}(i\to j)\big]\,.$$

□

B Proof of Thm. 4

Theorem 4. *Consider the problem of Sleeping MAB for arbitrary (adversarial) sequences of losses $\{\ell_t\}$ and availabilities $\{S_t\}$. Let $T\geq 1$ and $\eta^2=(\log K)/(2\sum_{t=1}^{T}|S_t|)$. Assume that $0\leq \ell_t(i)\leq 1$ for any $i\in S_t$ and $t\in[T]$. Then,*

$$\mathbb{E}\big[R_T^{int}(i\to j)\big]\leq 2\sqrt{2\log K\sum_{t=1}^{T}|S_t|}\leq 2\sqrt{2TK\log K}\,,$$

for all $i\neq j$ in $[K]$.

Proof. Let $\mathcal{F}_t:=\sigma(S_1,\ell_1,k_1,\ell_1,\ldots,k_t,S_{t+1},\ell_{t+1})$ denotes the past randomness of the algorithm and the adversary at round $t+1$. We respectively denote by $\mathbb{E}_t[\,\cdot\,]:=\mathbb{E}[\,\cdot\,|\mathcal{F}_t]$ and $\mathbb{P}_t(\,\cdot\,):=\mathbb{P}(\,\cdot\,|\mathcal{F}_t)$ the conditional expectation and probability.

Note that $\tilde{q}_t(i\to j)$ follows the prediction of the exponentially weighted average forecaster on the losses $\widehat{\ell}_t(i\to j)$. Noting that $-\eta\widehat{\ell}_t(i\to j)\leq 1$ for all $i\neq j$ and $t\geq 1$, and applying the upper-bound on the exponentially weighted average forecaster yields for any $i\neq j$ (see Thm. 1.5 of [18])

$$\sum_{t=1}^{T}\sum_{i'\neq j'}\tilde{q}_t(i'\to j')\widehat{\ell}_t(i'\to j\,)-\sum_{t=1}^{T}\widehat{\ell}_t(i\to j)\leq\frac{\log(K(K-1))}{\eta}+\eta\sum_{t=1}^{T}\sum_{i'\neq j'}\tilde{q}_t(i'\to j')\widehat{\ell}_t(i'\to j')^2\,. \tag{11}$$

Now, we compute the expectations. Note that S_t, ℓ_t and p_t are $\mathcal{F}_{t-1}$-measurable by assumption. Since $k_t\in S_t$ almost surely, we have for all $j\in[K]$

$$\begin{aligned}
\mathbb{E}_{t-1}\big[\widehat{\ell}_t(i\to j)\big] &\overset{(6)}{=} \mathbb{E}_{t-1}\Big[\sum_{k\neq i}\ell_t(k)\mathbb{1}\{k=k_t,j\in S_t\}+\frac{p_t(i)\ell_t(j)}{p_t(j)}\mathbb{1}\{j=k_t\}+\ell_t(k_t)\mathbb{1}\{j\notin S_t\}\Big]\\
&=\mathbb{E}_{t-1}\Big[\ell_t(k_t)\mathbb{1}\{j\in S_t\}-\ell_t(i)\mathbb{1}\{i=k_t,j\in S_t\}+\frac{p_t(i)\ell_t(j)}{p_t(j)}\mathbb{1}\{j=k_t\}+\ell_t(k_t)\mathbb{1}\{j\notin S_t\}\Big]\\
&=\mathbb{E}_{t-1}\Big[\ell_t(k_t)-\ell_t(i)\mathbb{1}\{i=k_t,j\in S_t\}+\frac{p_t(i)\ell_t(j)}{p_t(j)}\mathbb{1}\{j=k_t\}\Big]\\
&=\mathbb{E}_{t-1}\Big[\ell_t(k_t)+p_t(i)(\ell_t(j)-\ell_t(i))\mathbb{1}\{j\in S_t\}\Big]\\
&=\mathbb{E}_{t-1}\Big[\ell_t(k_t)+(\ell_t(j)-\ell_t(i))\mathbb{1}\{i=k_t,j\in S_t\}\Big]\,.
\end{aligned}$$

Furthermore, by definitions of $\widehat{\ell}_t(i\to j), p_t$ and q_t, and denoting $\tilde{Q}_t=\sum_{i'\neq j'}\tilde{q}_t(i'\to j')\mathbb{1}\{j'\in S_t\}$, we have

$$\mathbb{E}_{t-1}\Big[\sum_{i\neq j}\tilde{q}_t(i\to j)\widehat{\ell}_t(i\to j)\Big]\overset{(6)}{=}\mathbb{E}_{t-1}\Big[\sum_{i\neq j}\tilde{q}_t(i\to j)\sum_{k=1}^{K}p_t^{i\to j}(k)\widehat{\ell}_t(k)\mathbb{1}\{j\in S_t\}+\sum_{i\neq j}\tilde{q}_t(i\to j)\ell_t(k_t)\mathbb{1}\{j\notin S_t\}\Big]$$

$$
\begin{aligned}
&\overset{(8)}{=} \mathbb{E}_{t-1}\Big[\tilde{Q}_t \sum_{i\neq j} q_t(i\to j) \sum_{k=1}^{K} p_t^{i\to j}(k)\widehat{\ell}_t(k) + (1-\tilde{Q}_t)\ell_t(k_t)\Big] \\
&\overset{(9)}{=} \mathbb{E}_{t-1}\Big[\tilde{Q}_t \sum_{k=1}^{K} p_t(k)\widehat{\ell}_t(k) + (1-\tilde{Q}_t)\ell_t(k_t)\Big] \\
&= \mathbb{E}_{t-1}\big[\tilde{Q}_t\ell_t(k_t) + (1-\tilde{Q}_t)\ell_t(k_t)\big] = \mathbb{E}_{t-1}\big[\ell_t(k_t)\big]\,.
\end{aligned}
$$

Therefore, the expectation of the left-hand side of (11) equals the internal sleeping regret:

$$
\mathbb{E}\Big[\sum_{t=1}^{T}\sum_{i'\neq j'} \tilde{q}_t(i'\to j')\widehat{\ell}_t(i'\to j\ ) - \sum_{t=1}^{T}\widehat{\ell}_t(i\to j)\Big] = R_T^{\mathrm{int}}(i\to j)\,. \tag{12}
$$

On the other hand,

$$
\begin{aligned}
&\mathbb{E}_{t-1}\Big[\sum_{i\neq j}\tilde{q}_t(i\to j)\widehat{\ell}_t(i\to j)^2\Big] \\
&= \mathbb{E}_{t-1}\Big[\sum_{i\neq j}\tilde{q}_t(i\to j)\Big(\sum_{k=1}^{K} p_t^{i\to j}(k)\widehat{\ell}_t(k)\Big)^2 \mathbb{1}\{j\in S_t\} + (1-\tilde{Q}_t)\ell_t(k_t)^2\Big] \\
&\leq \mathbb{E}_{t-1}\Big[\sum_{i\neq j}\tilde{q}_t(i\to j)\sum_{k=1}^{K} p_t^{i\to j}(k)\widehat{\ell}_t(k)^2\mathbb{1}\{j\in S_t\} + (1-\tilde{Q}_t)\ell_t(k_t)^2\Big] \\
&\overset{(8)\text{ and }(9)}{=} \mathbb{E}_{t-1}\Big[\tilde{Q}_t\sum_{k=1}^{K} p_t(k)\widehat{\ell}_t(k)^2 + (1-\tilde{Q}_t)\ell_t(k_t)^2\Big] \\
&= \mathbb{E}_{t-1}\Big[\tilde{Q}_t\frac{\ell_t(k_t)^2}{p_t(k_t)} + (1-\tilde{Q}_t)\ell_t(k_t)^2\Big] \\
&= \tilde{Q}_t\sum_{k\in S_t} p_t(k)\frac{\ell_t(k)^2}{p_t(k)} + (1-\tilde{Q}_t)\mathbb{E}_{t-1}\big[\ell_t(k_t)\big] \\
&\leq (|S_t|-1)\tilde{Q}_t + 1 \leq |S_t|\,.
\end{aligned}
$$

The expectation of the right-hand-side of (11) can thus be upper-bounded as

$$
\eta\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i\neq j}\tilde{q}_t(i\to j)\widehat{\ell}_t(i\to j)^2\right] \leq \eta\sum_{t=1}^{T}|S_t|\,.
$$

Therefore, substituting the above inequality and (12) into (11), and optimizing η concludes the proof. □

C Proof of Theorem 5

Theorem 5. *Consider the problem setting of Sleeping DB defined above (Sec. 4) and let $T \geq 1$. Then, Sparring SI-EXP3 satisfies*

$$
\mathbb{E}[R_T^{\texttt{SI-DB}}] \leq 2K^2\sqrt{2TK\log K}\,.
$$

Proof. Denote by $j_t^* = \operatorname{argmax}_{j \in A_t(i_t)} P(j, i_t)$ and by $i_t^* = \operatorname{argmax}_{i \in A_t(j_t)} P(i, j_t)$. Then, using that $P(i,j) = 1 - P(j,i)$, we have

$$\begin{aligned}\mathbb{E}[R_T^{\texttt{SI-DB}}] &:= \mathbb{E}\left[\sum_{t=1}^{T} \frac{P(j_t^*, i_t) + P(i_t^*, j_t) - 1}{2}\right] \\ &:= \mathbb{E}\left[\sum_{t=1}^{T} \frac{P(j_t, i_t) - P(j_t, i_t^*)}{2}\right] + \mathbb{E}\left[\sum_{t=1}^{T} \frac{P(i_t, j_t) - P(i_t, j_t^*)}{2}\right]. \end{aligned} \tag{13}$$

Let us focus on the first term of the r.h.s, the other one can be analysed similarly.

$$\begin{aligned}P(j_t, i_t) - P_t(j_t, i_t^*) &= \sum_{i=1}^{K}\sum_{i'=1}^{K} \left(P(j_t, i) - P(j_t, i')\right)\mathbb{1}\{i = i_t, i' = i_t^*\} \\ &\leq \sum_{i=1}^{K}\sum_{i' \in D_i} \left(P(j_t, i) - P(j_t, i')\right)\mathbb{1}\{i = i_t, i' \in S_t\},\end{aligned}$$

where $D_i := \{i' \in S_t : P(i', j_t) \geq P(i, j_t)\}$. The last inequality is because $i_t^* \in S_t \cap D_i$ and $P(j_t, i) - P(j_t, i') > 0$ for any $i' \in D_i$. Note that D_i does not depend on j_t because of the total ordering assumption. Then, taking the expectation and summing over t, we get

$$\begin{aligned}R_T^{\text{left}} &:= \mathbb{E}\left[\sum_{t=1}^{T} P(j_t, i_t) - P_t(j_t, i_t^*)\right] \\ &\leq \sum_{i=1}^{K}\sum_{i' \in D_i} \mathbb{E}\left[\sum_{t=1}^{T} \left(P(j_t, i) - P(j_t, i')\right)\mathbb{1}\{i = i_t, i' \in S_t\}\right] \\ &\leq \sum_{i=1}^{K}\sum_{i' \in D_i} \mathbb{E}\left[\sum_{t=1}^{T} \left(\ell_t^{\text{left}}(i) - \ell_t^{\text{left}}(i')\right)\mathbb{1}\{i = i_t, i' \in S_t\}\right] \\ &\leq \sum_{i=1}^{K}\sum_{i' \in D_i} 2\sqrt{2TK\log K} \leq 2K^2\sqrt{2TK\log K},\end{aligned}$$

where the second to last inequality is by Theorem 4 by construction of $\mathcal{A}^{\text{left}}$ which minimizes the internal regret. Similarly, we can show that

$$R_T^{\text{right}} := \mathbb{E}\left[\sum_{t=1}^{T} P(i_t, j_t) - P_t(i_t^*, j_t)\right] \leq 2K^2\sqrt{2TK\log K}.$$

Substituting both upper-bounds into (13) concludes the proof. □

D Experiments

D.1 Additional experiments on sleeping multi-armed bandits

In this section, we run some additional experiments to compare the 3 algorithms:
- SI-EXP3: Our proposed algorithm Internal Sleeping-EXP3 described in Sec. 3;

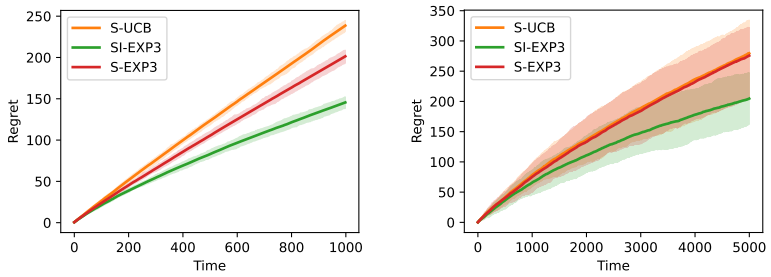

Figure 3: [Left] Random environment with dependence [Right] Random games

- S-UCB: The sleeping UCB procedure proposed by [23] for ordering regret with stochastic losses;
- S-EXP3: The algorithm Sleeping-EXP3G designed by [39] for ordering regret with adversarial losses and stochastic sleeping.

Again, each experiment is run 50 times and the policy regret is plotted in Figure 3.

Random environment with dependence This setup is similar to the dependent environment of Sec. 5 where the distribution of (S_t, ℓ_t) are uniformly sampled at the start of each run.

More precisely. We consider the following stochastic environment with $K = 5$. The pairs (S_t, ℓ_t) are i.i.d. and sampled as follows.

At the start of each run, five availability sets $\mathcal{A}_1, \dots, \mathcal{A}_5 \subseteq [K]$ are sampled by including independently each action with probability $1/2$. If a set contains no action, it is sampled again. Then, for each set $m = 1, \dots, 5$, a mean vector $\mu_m \in \mathbb{R}^K$ is uniformly sampled on $(0,1)^K$. Then, for $t = 1, \dots, T$, the availability set S_t is drawn uniformly from $\{\mathcal{A}_1, \dots, \mathcal{A}_5\}$ and the losses of each arm k is sample from a Bernoulli with parameter $\mu_{m_t}(k)$, where m_t is such that $S_t = \mathcal{A}_{m_t}$.

Random two-player zero-sum games This setup is similar to the Rock-Paper-Scissors environment of Sec. 5 but with $K = 10$ players and a random payoff matrix.

At the start of each run, a payoff matrix $G \in \mathbb{R}^{K \times K}$ is randomly sampled as follows. For each $1 \leq i < j \leq K$, $G_{ij} \overset{\text{i.i.d.}}{\sim} \text{Unif}((-1,1))$, $G_{ii} = 1/2$ and $G_{ji} = -G_{ij}$:

$$G = \begin{pmatrix} 0 & G_{12} & G_{13} & \dots \\ -G_{12} & 0 & G_{23} & \dots \\ -G_{13} & -G_{23} & 0 & \dots \\ \dots & \dots & \dots & 0 \end{pmatrix}.$$

Furthermore, 4 availability sets $(\mathcal{A}_m)_{1 \leq m \leq 4}$ are randomly sampled by including each action with probability $1/2$. For each $m \in [4]$, we compute $p_m \in \Delta_K$ the Nash equilibrium of the game G restricted to actions in $\mathcal{A}_m$. Note that $p_m(k) = 0$ for all $k \notin \mathcal{A}_m$. Then, for each $t = 1, \dots, T$, an availability set $S_t = \mathcal{A}_{m_t}$ is uniformly sampled in $\{\mathcal{A}_1, \dots, \mathcal{A}_4\}$. The algorithm is asked to choose an action $k_t \in S_t$ and receives the loss $\ell_t(k_t) \sim \mathcal{B}(G_{j_t k_t})$, where j_t is the action chosen by an optimal adversary that follows p_{m_t}.

The optimal strategy in this case should be too also follow $k_t \sim \mathcal{A}_{m_t}$ and would incur $\mathbb{E}[\ell_t(k_t)] = 1/2$. Figure 3 (right) plots the cumulative pseudo-regret $R_T = \sum_{t=1}^T G_{k_t, j_t} - T/2$. As we can see, SI-EXP3 significantly outperforms S-UCB and S-EXP3. It would be worth to investigate if SI-EXP3 could be used to compute Nash equilibria in repeated two-player zero-sum games with non-available actions.

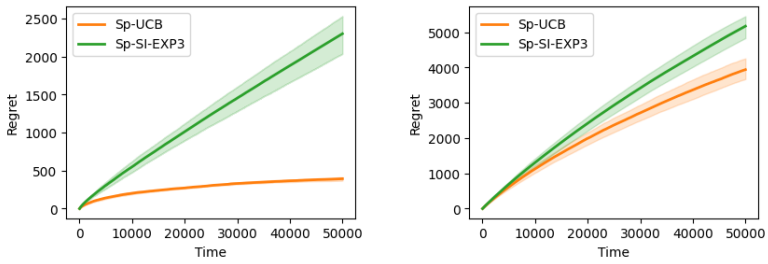

Figure 4: Dueling Bandits with non-repeating arms for $K = 4$ [Left] and $K = 30$ [Right] respectively. ($M = 5$).

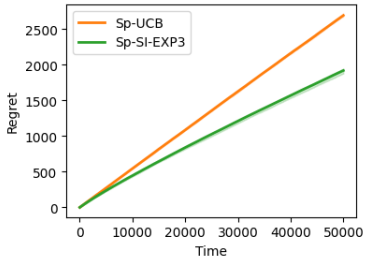

Figure 5: Preference Learning with Categories where Utilities depend on Availability.

D.2 Experiments on sleeping dueling bandits

Dueling Bandits with non-repeating arms This experimental setup is motivated by the first example in Sec. 4.1 where we want the algorithm to converge to the top 2 items (best pair). We consider utility scores $\{u_1, u_2, ..., u_K\}$ corresponding to the K arms, and the preference matrix P, with P_{ij} defined as $P_{ij} = \frac{u_i}{u_i+u_j}$ indicating the probability of arm i winning over arm j. We repeat this experiment for M independent runs, by sampling a random utility vector at the beginning of each run. We assume that all the arms are available for the first bandit. All the arms except the one chosen by the first bandit, is available to the second bandit. Each bandit runs its own custom algorithm (which can be UCB, SI-EXP3, etc.). Finally, the winning arm is decided according to P and the loss is 1 for the bandit that chose this arm and 0 for the other bandit. In the figures below, we plot the Internal Sleeping DB regret for choices of Sp-UCB and Sp-SIEXP3 (Sparring UCB, SI-EXP3 where both bandits internally use the UCB algorithm and the SI-EXP3 algorithm respectively). In Fig. 4 we plot Sp-SIEXP3 and Sp-UCB and we see that in this relatively simple setting Sp-UCB outperforms Sp-SIEXP3.

Note that despite its surprisingly good performance in Fig. 4, especially for small number of arms, Sp-UCB has no theoretical guarantees for dueling bandits. It would be interesting to study whether such guarantees are possible or whether it has a linear worst-case regret. Furthermore, Sp-UCB strongly assumes a total and fixed ordering of stock performance. As we can see in the following example, Sp-SI-EXP3 works better as soon as there is some dependence between the preference matrix and the availabilities. It is also worth to emphasize that we could not compare with classical dueling bandit algorithm that are not suited for this setting.

Preference Learning with Categories In this experimental setup we have availability dependent utility matrices. This is motivated by the following setting: if one item of a category is

unavailable, the overall utility values of all items in the category goes down. In the real world, this could be in a setting where I would want to watch a season of a show only if all the seasons are available. Concretely, we have K different availability sets, where $\mathcal{A}_i$ has all items available except i. We also have K utility vectors: $\{u_1, u_2, ..., u_K\}$. At each turn we randomly choose $r \in \{1, 2, ..., K\}$ and select $\mathcal{A}_r$ and u_r. Similar to the previous setting, the first bandit chooses an available item and the second bandit chooses an available item except the one chosen by the first bandit. In Fig. 5, we choose the utility vectors as $\{(1, 2, ..., K), (K, 1, 2, ..., K-1), ..., (2, 3, ..., K, 1)\}$ and we see that Sp-SIEXP3 significantly outperforms Sp-UCB.