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

Pierre Gaillard
Univ. Grenoble Alpes, Inria,
CNRS, Grenoble INP

Aadirupa Saha*
TTI, Chicago

Soham Dan#
IBM Research

Abstract

We address the problem of Internal Regret in adversarial Sleeping Bandits and the relationship between different notions of sleeping regrets in multi-armed bandits. We propose a new concept called Internal Regret for sleeping multi-armed bandits (MAB) and present an algorithm that achieves sublinear Internal Regret, even when losses and availabilities are both adversarial. We demonstrate that a low internal regret leads to both low external regret and low policy regret for i.i.d. losses. Our contribution is unifying existing notions of regret in sleeping bandits and exploring their implications for each other. In addition, we extend our results to Dueling Bandits (DB), a preference feedback version of multi-armed bandits, and design a low-regret algorithm for sleeping dueling bandits with stochastic preferences and adversarial availabilities. We validate the effectiveness of our algorithms through empirical evaluations.

1 INTRODUCTION

The problem of online sequential decision-making in standard multi-armed bandits (MAB) is well studied in machine learning (Auer, 2000; Vermorel and Mohri, 2005) and used to model online decision-making problems under uncertainty. Due to their implicit exploration-vs-exploitation trade-off, 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 (Even-Dar et al., 2006; Auer et al., 2002; Auer, 2000; Agrawal and Goyal, 2012; Bubeck et al., 2012).

From a practical perspective, the decision space (also called the arm space, denoted as $\mathcal{A} = 1, \ldots, K$) frequently changes over time due to the unavailability of some items. For example, in a retail store, some items may go out of stock, certain websites may go down, or some restaurants may be closed. This situation is known as sleeping bandits in the multi-armed bandit literature (Kanade et al., 2009; Neu and Valko, 2014a; Kanade and Steinke, 2014a; Kale et al., 2016). In this scenario, the set $S_t \subseteq \mathcal{A}$ of available actions can vary stochastically (Neu and Valko, 2014a; Cortes et al., 2019) or adversarially (Kale et al., 2016; Kleinberg et al., 2010; Kanade and Steinke, 2014a) at any round $t \geq 1$. Over the years, several lines of research have been conducted for sleeping multi-armed bandits with different notions of regret performance, e.g. policy, ordering, or sleeping external regret (Blum and Mansour, 2007; Neu and Valko, 2014a; Saha et al., 2020).

This paper introduces a new notion of sleeping regret, called *Sleeping Internal Regret*, which helps to connect different existing concepts of sleeping regret in multi-armed bandit problems. We show 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). Additionally, an efficient worst-case regret algorithm for sleeping internal regret is proposed, which achieves $O(\sqrt{T})$ performance, where T is the number of rounds. The implications of these results for the Dueling Bandit (DB) framework (Yue et al., 2012; Ailon et al., 2014; Zoghi et al., 2014; Saha and Gopalan, 2019a,b) are also

Corresponding email: aadirupa@ttic.edu
$*$Author is currently with Apple ML Research, US.
# Major part of the work was done while the author was at the University of Pennsylvania

Proceedings of the 26$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206. Copyright 2023 by the author(s).

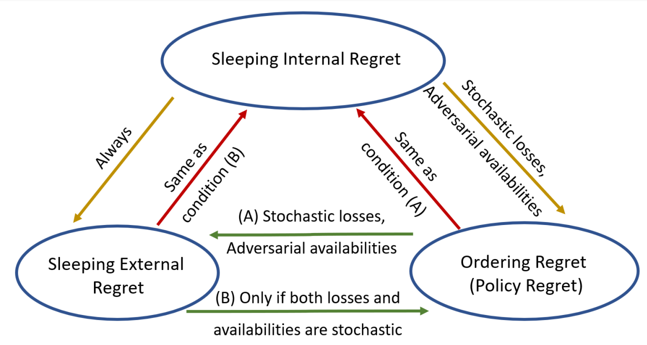
Sleeping Internal Regret
Always
Same as condition (B)
Same as condition (A)
Stochastic losses, Adversarial availabilities
(A) Stochastic losses, Adversarial availabilities
Sleeping External Regret
Ordering Regret (Policy Regret)
(B) Only if both losses and availabilities are stochastic

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

discussed. The main contributions of this paper can be summarized 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 (Blum and Mansour, 2007; Gaillard et al., 2014). 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 (Raj et al., 2020; Zhao et al., 2020; Campolongo and Orabona, 2021; Zhang et al., 2019; Wei et al., 2016). It has the advantage that efficient no-regret algorithms can be designed even when both availabilities $S_t \subseteq \mathcal{A}$ and losses $\ell_t \in [0,1]^K$ are adversarial.

– The second one, called *ordering regret*, is mostly used in the bandit literature (Kleinberg et al., 2010; Saha et al., 2020; Kanade and Steinke, 2014b; Neu and Valko, 2014b). It compares the cumulative loss of the learner, with the one of the best ordering σ^* (i.e., a permutation of $[K]$) 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 (Kleinberg et al., 2010).

– 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* (Neu and Valko, 2014a; Saha et al., 2020). 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.** We propose an efficient algorithm (SI-EXP3, Alg. 1) w.r.t. *Sleeping Internal Regret*, and design a $O(\sqrt{T})$ regret algorithm (Thm. 4). As motivated, the generalizability of our regret further implies $O(\sqrt{T})$ external regret 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 bandit setup, which is a relative feedback based variant of standard MAB (Zoghi et al., 2014; Ailon et al., 2014; Bengs et al., 2021) (Sec. 4).

– General Sleeping DB. We propose a new and more unifying notion of sleeping dueling bandit 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 the standard notion of DB setting where $A_t = [K]^2$ without sleeping, but also the setup of Sleeping DB for $A_t = S_t \times S_t$, (Saha and Gaillard, 2021).

– Unifying Sleeping DB Regret. Next, based on 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 (Zoghi et al., 2014) as well as sleeping DB regret (Saha and Gaillard, 2021) (Rem. 4).

– Optimal Algorithm Design. We propose an efficient and order optimal $O(\sqrt{T})$ sleeping DB algorithm (Thm. 5). This improves the regret bound of Saha and Gaillard (2021) that only get $O(T^{2/3})$ worst-case regret even in the more restrictive $A_t = S_t \times S_t$ setting.

- **Experiments.** In Sec. 5, we provide empirical evidence to support our theoretical findings. Our algorithm performs better than baselines when there is a correlation between S_t and ℓ_t. The experimental results also suggest that our algorithm can effectively converge to Nash equilibria in two-player zero-sum games that include sleeping actions (as discussed in Rem. 5).

Related Work. The problem of regret minimization for stochastic multi-armed bandits is widely studied in the online learning literature (Auer et al., 2002; Agrawal and Goyal, 2012; Lattimore and Szepesvári, 2018; Audibert and Bubeck, 2010), 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* (Kanade et al., 2009; Neu and Valko, 2014a; Kanade and Steinke, 2014a; Kale et al., 2016), for both stochastic rewards and adversarial availabilities (Kale et al., 2016; Kleinberg et al., 2010; Kanade and Steinke, 2014a) as well as adversarial rewards and stochastic availabilities (Kanade et al., 2009; Neu and Valko, 2014a; Cortes et al., 2019). 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 (Kleinberg et al., 2010; Kale et al., 2016). The optimal and efficient algorithm for this case is by Saha et al. (2020), 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 (Zoghi et al., 2014, 2015; Komiyama et al., 2015; Wu and Liu, 2016; Saha et al., 2021a; Saha and Krishnamurthy, 2022; Saha, 2021; Saha et al., 2021b), or even extending the pairwise preference to subsetwise preferences (Sui et al., 2017; Brost et al., 2016; Saha and Gopalan, 2018a, 2019b, 2020; Ghoshal and Saha, 2022; Ren et al., 2018).

Little research has been done on dueling bandits in a sleeping setup, despite its practical usefulness. In a recent study, Saha and Gaillard (2021) addressed the Sleeping Dueling Bandit problem in the case of stochastic preferences and adversarial availabilities. However, their proposed algorithms only provide a suboptimal regret guarantee of $O(T^{2/3})$. Our work is the first to achieve $O(\sqrt{T})$ regret for Sleeping Dueling Bandits, as detailed in Thm. 5.

2 SETTING

This section introduces the formal problem of sleeping multi-armed bandit and a new notion of learner's performance called "Internal Sleeping Regret" (as described in Section Sec. 2.3). It also discusses the different existing regret bounds in Sleeping MAB (as outlined in Section Sec. 2.1) and their relationships, which are summarized in Fig. 1.

Problem Setting: Sleeping MAB. Let $[K] = \{1, \ldots, K\}$ 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 $\ell_t(k_t) \in [0, 1]$ of the selected arm. The availability sequence $\{S_t\}_{t=1}^T$ and the loss sequence $\{\ell_t\}_{t=1}^T$ can be either stochastic or adversarial (oblivious) in nature. We consider the hardest setting of adversarial losses and availabilities, which 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 Blum and Mansour (2007). Here, the learner is compared with each arm, only on the rounds in which the arm is available:

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

The learner is asked to control $\max_{k \in [K]} R_T(k) = o(T)$ as $T \to \infty$. In Blum and Mansour (2007), 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 $\sigma = (\sigma_1, \ldots, \sigma_K) \in \Sigma$ of the arms, where Σ denotes the set of all possible orderings of $[K]$:

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

where $\sigma(S_t) = \big\{\sigma_k \text{ s.t. } k = \text{argmin}\{i : \sigma_i \in S_t\}\big\}$ denotes the best arm available in S_t. Consequently, in this case, the learner's regret 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 Kleinberg et al. (2010); Kanade and Steinke (2014a); Kale et al. (2016). For adversarial losses and i.i.d. S_t (where each arm is independently available according to a Bernoulli distribution), Saha et al. (2020) 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 Kleinberg et al. (2010).

3. Policy Regret A policy $\pi : 2^{[K]} \mapsto [K]$ denotes here a mapping from a set of available actions/experts

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

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:

$$R_T^{\text{policy}}(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(k_t) - \sum_{t=1}^{T} \ell_t(\pi(S_t))\right], \qquad (3)$$

where the expectation is taken w.r.t. the availabilities and the randomness of the player's strategy (Saha et al., 2020). 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 imply 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 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)?** This direction is not true in general, as indeed, it would otherwise contradict the hardness result for ordering regret. Minimizing ordering regret is known to be NP-Hard for adversarial ℓ_t and S_t Kleinberg et al. (2010), 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, $R_T^{\text{ordering}}\big((1, 2, 3)\big) = T/2$ 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 that ordering regret is a stronger notion of regret compared to external regret.

To precisely summarize the relationships between the various notions of regret mentioned above, we present them graphically in Figure 1. However, keeping track of so many different regret notions is already challenging. It is even more difficult to determine which one to work with and whether optimizing one will necessarily guarantee low regret in another. Thus, we aim to find a more general notion of sleeping regret that implies both. To address this challenge, we introduce a new sleeping MAB regret notion that unifies the existing 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 Foster and Vohra (1999), and largely studied in online learning since then, see among other Cesa-Bianchi and Lugosi (2006); Stoltz (2005); Stoltz and Lugosi (2005); Blum and Mansour (2007). 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*

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

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 (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 (Stoltz and Lugosi, 2005) (without sleeping).

Lemma 2 (Internal Regret Implies External Regret, Stoltz and Lugosi (2005)). *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). Such a result would 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 will 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 Stoltz and Lugosi (2005) 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 INTERNAL SLEEPING REGRET

We now propose an EXP3-based algorithm that guarantees sublinear sleeping internal regret. It is important to note that our algorithm works for the most challenging scenario of adversarial losses and availabilities, which includes stochastic settings (either losses or availabilities) as a special case. As proven 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 $\{S_t\}_{t \in [T]}$. In addition, the generality of our internal regret (as shown in Fig. 1) implies $O(\sqrt{T})$ external regret in any scenario, as well as ordering regret for i.i.d losses, as explained in Rem. 3.

Our regret analysis is inspired from the construction of Stoltz (2005) (see Section 3, Thm. 3.2) for the internal regret although the 'sleeping component' or item non-avilabilities is not considered by Stoltz (2005).

Another relevant work is Blum and Mansour (2007), 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 Blum and Mansour (2007) is of the form

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

Thus $R_T^{\text{int}}(i \to j)$ 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. Moreover, their algorithm differs from ours and the one of Stoltz (2005) because it combines algorithms. In short, Blum and Mansour (2007) provides a reduction that needs K

instances of a no-external regret algorithm, that are combined by a meta-algorithm. In contrast, we use a single instance of EXP3 to combine $K(K-1)$ fictitious experts. Each expert is in charge of controlling the internal regret $R_T^{int}(i \to j)$. No external regret algorithm is needed on the lower level. Because combining bandit algorithms is a difficult task (see e.g., Agarwal et al. (2017)), Blum and Mansour (2007) incurs suboptimal factors in $[K]$ (see their Thm. 11 which yields an internal regret of order $O(K\sqrt{TK\log K})$), need extra assumptions (see their Lem. 10), and the adaptation of their analysis to sleeping is sophisticated. Note in particular that, their regret bound (Thm. 18) only holds in the full information setting, while their result on bandit feedback (Thm. 11) does not have the sleeping component. 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 (Auer et al., 2002), combines $K(K-1)$ sleeping experts indexed by $i \to j$, for $i \neq j$. Each expert aims to minimize the internal sleeping regret $R_T^{\text{int}}(i \to j)$. 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 $t = 1, 2, \dots, T$ 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)
        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
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,

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

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

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

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

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

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

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

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

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

The existence and the practical computation of such a p_t is an application of Lemma 3.1 of Stoltz (2005). Similarly to Blum and Mansour (2007) and Stoltz (2005), the space and per-round time complexities of SI-EXP3 are $O(K^2)$ and $O(K^3)$ respectively. The bottleneck is the above fixed point resolution of p_t in (9).

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 $\eta^2 = (\log K)/\big(2\sum_{t=1}^T |S_t|\big)$. 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 constant factor by choosing a time-varying learning rate $\eta_t^2 = (\log K)/\big(2\sum_{s=1}^t |S_s|\big)$ or by using the doubling-trick technique (Cesa-Bianchi and Lugosi, 2006).

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 IN 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 Saha and Gaillard (2021) for dueling bandits with adversarial sleeping. We consider the stochastic dueling bandit setting with a preference matrix $P \in [0,1]^{K\times K}$ such that $P(i,j) = 1 - P(j,i)$ 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 satisfied for utility-based preference matrices (Ailon et al., 2014), or for Plackett-Luce model (Azari et al., 2012), 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 $(j,i) \in A_t$. 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 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 Zoghi et al. (2014) for definition) for all rounds since $\mathbb{P}$ respects total ordering. Thus, in this case, $R_T^{\texttt{SI-DB}}$ reduces to the standard CW-regret studied in DB (Yue and Joachims, 2009; Zoghi et al., 2014; Bengs et al., 2021). Moreover, $R_T^{\texttt{SI-DB}}$ also generalizes the notion of Sleeping Dueling Bandit of Saha and Gaillard (2021) 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 Saha and Gaillard (2021)). 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 $A_t = \{(i,j) \in [K]^2, i \neq j\}$. At each round, our algorithm is then required to select two distinct items, which is a new constraint in the context of dueling bandits. This restriction is relevant for various applications, including recommendation systems that recommend pairs of different items. Our algorithm is designed to converge to the best-performing item pair. An intriguing future research direction would be to extend our approach to subsets of unique battling items of any size, which may be larger than two. A related scenario was explored by Saha and Gopalan (2018b), but they allow the selection of duplicate items.

(ii). Preference learning with categories. An application that illustrates this point involves arms that are categorized into different groups. For instance, consider a movie recommendation system that offers a variety of movie types, such as action, documentaries, TV series, and romantic movies. At each round, the system is tasked with suggesting movies from different categories to add diversity to the recommendations. Our algorithm can learn the optimal movies as well as the two best categories simultaneously. Furthermore, the ability to sleep permits the collection of movies to change 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 Saha and Gaillard (2022) (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 Ailon et al. (2014) 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*

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

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 Saha and Gaillard (2021). 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 get $O(\sqrt{T})$.

5 EXPERIMENTS

We provide sleeping multi-armed bandit simulations, averaged across 20 runs[2]. More findings are in App. D. We compare the results of the following algorithms:

- SI-EXP3: Alg. 1;
- S-UCB: A sleeping UCB procedure (Kleinberg et al., 2010) initially designed for ordering regret with stochastic losses;
- S-EXP3: Sleeping-EXP3G (Saha et al., 2020) initially designed for ordering regret with adversarial losses and i.i.d. sleeping.

We compare 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 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 (Policy, External, and Internal) for the algorithms are shown in Fig. 2. It is observed that when there are dependencies between the loss vectors and availabilities, SI-EXP3 outperforms the other two algorithms significantly. This is not surprising since S-EXP3 and S-UCB were designed to perform well under the assumption of a best fixed ordering. Typically, these algorithms first rank the actions based on their average performance and then choose the best action available in S_t. However, when there is a dependency between S_t and ℓ_t, the best ordering may change from round to round, leading to linear Policy regret for S-UCB and S-EXP3. It is important to note that such a situation can occur frequently in real-world scenarios. For example, an e-commerce site that sells products may want to suggest complementary items. If some of the items are out of stock, suggesting certain items may not make sense.

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*

[2] The code is available at: https://github.com/sdan2/Internal-Regret-Bandits

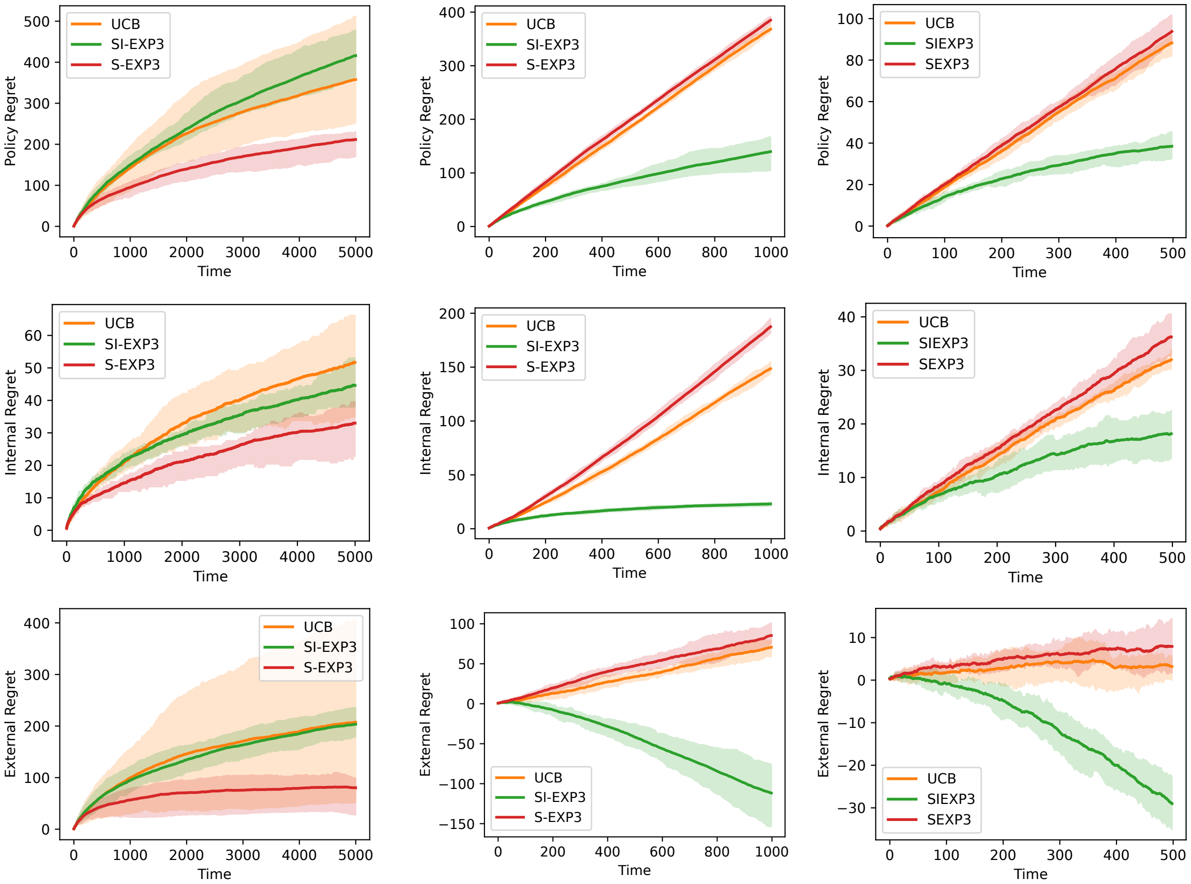

Figure 2: [Left] Stochastic environment [Middle] Dependent environment [Right] Rock Paper Scissors

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 Policy 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 for two-player zero-sum randomized games with a random payoff matrix P (App. D.1) and also Dueling Bandit experiments (App. D.2). 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).

6 CONCLUSION

In this paper, we introduce the notion of *internal sleeping regret* for multi-armed bandits. Under some assumptions this notion implies simultaneously existing notions considered in the literature in adversarial or stochastic settings. We provide an efficient algorithm with $O(\sqrt{T})$ regret and motivate our regret with applications for dueling bandits. 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.

Future Directions. 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 (Kleinberg et al., 2010) or in a recent work (Saha and Gaillard, 2021) 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* (e.g., Saha and Gopalan, 2019a, 2020; Ren et al., 2018).

References

Alekh Agarwal, Haipeng Luo, Behnam Neyshabur, and Robert E Schapire. Corralling a band of bandit algorithms. In *Conference on Learning Theory*, pages 12–38. PMLR, 2017.

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

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

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.

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.

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

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

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.

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

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.

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.

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

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

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.

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.

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

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.

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.

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

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.

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

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

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

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

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.

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

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, 2014a.

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

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

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.

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

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

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

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

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

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

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

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.

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.

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.

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

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

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

Gilles Stoltz and Gábor Lugosi. Internal regret in online portfolio selection. *Machine Learning*, 59(1): 125–159, 2005.

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.

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

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.

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

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.

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.

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.

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.

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.

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: A Unified Framework for Achieving Optimal Regret Guarantees in Sleeping Bandits

A Appendix for Sec. 2

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

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, $\mathbb{E}\big[\sum_{t=1}^T \ell_t(k_t)\big] = T/2$. 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 $R_T^{\textbf{ordering}}$ with Stochastic losses, Adversarial availabilities does imply low internal regret $R_T^{\textbf{int}}$

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*

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

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,*

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

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*

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

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

Proof. Let $\mu_k = \mathbb{E}\big[\ell_t(k)\big]$ 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 $\mathbb{E}\big[R_T^{\text{ordering}}(\sigma)\big] \leq \mathbb{E}\big[R_T^{\text{ordering}}(\sigma^*)\big]$. 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,

$$\begin{aligned}
\mathbb{E}\big[R_T^{\text{ordering}}(\sigma)\big] &:= \mathbb{E}\Big[\sum_{t=1}^{T}\ell_t(k_t) - \ell_t\big(\sigma(S_t)\big)\Big] = \mathbb{E}\Big[\sum_{t=1}^{T}\mu_{k_t} - \mu_{k_t^*}\Big] \\
&= \mathbb{E}\Big[\sum_{t=1}^{T}\sum_{i=1}^{K}\sum_{j\in D_i}(\mu_i-\mu_j)\mathbb{1}\{i=k_t, j=k_t^*\}\Big] \qquad \leftarrow k_t^* \in D_i \text{ because it is the best item in } S_t \\
&\leq \mathbb{E}\Big[\sum_{t=1}^{T}\sum_{i=1}^{K}\sum_{j\in D_i}(\mu_i-\mu_j)\mathbb{1}\{i=k_t, j\in S_t\}\Big] \qquad \leftarrow \text{because } k_t^* \in S_t \text{ and } \mu_i-\mu_j \geq 0 \text{ for any } j \in D_i \\
&\leq \sum_{i=1}^{K}\sum_{j\in D_i}\mathbb{E}\big[R_T^{\text{int}}(i\to j)\big]\,.
\end{aligned}$$

□

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 $\eta^2 = (\log K)/\big(2\sum_{t=1}^{T}|S_t|\big)$. 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 $\mathcal{F}_t := \sigma(S_1, \ell_1, k_1, \ell_1, \dots, k_t, S_{t+1}, \ell_{t+1})$ 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 Hazan et al. (2021))

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

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 $j \in [K]$

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

$$
\begin{aligned}
&= \mathbb{E}_{t-1}\Big[\ell_t(k_t)\mathbb{1}\{j \in S_t\} - \ell_t(i)\mathbb{1}\{i = k_t, j \in S_t\} + \frac{p_t(i)\ell_t(j)}{p_t(j)}\mathbb{1}\{j = k_t\} + \ell_t(k_t)\mathbb{1}\{j \notin S_t\}\Big] \\
&= \mathbb{E}_{t-1}\Big[\ell_t(k_t) - \ell_t(i)\mathbb{1}\{i = k_t, j \in S_t\} + \frac{p_t(i)\ell_t(j)}{p_t(j)}\mathbb{1}\{j = k_t\}\Big] \\
&= \mathbb{E}_{t-1}\Big[\ell_t(k_t) + p_t(i)(\ell_t(j) - \ell_t(i))\mathbb{1}\{j \in S_t\}\Big] \\
&= \mathbb{E}_{t-1}\Big[\ell_t(k_t) + (\ell_t(j) - \ell_t(i))\mathbb{1}\{i = k_t, j \in S_t\}\Big]\,.
\end{aligned}
$$

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

$$
\begin{aligned}
\mathbb{E}_{t-1}\Big[\sum_{i \neq j} \tilde{q}_t(i \to j)\widehat{\ell}_t(i \to j)\Big] &\overset{(6)}{=} \mathbb{E}_{t-1}\Big[\sum_{i \neq j} \tilde{q}_t(i \to j) \sum_{k=1}^{K} p_t^{i \to j}(k)\widehat{\ell}_t(k)\mathbb{1}\{j \in S_t\} + \sum_{i \neq j} \tilde{q}_t(i \to j)\ell_t(k_t)\mathbb{1}\{j \notin S_t\}\Big] \\
&\overset{(8)}{=} \mathbb{E}_{t-1}\Big[\tilde{Q}_t \sum_{i \neq j} q_t(i \to j) \sum_{k=1}^{K} p_t^{i \to j}(k)\widehat{\ell}_t(k) + (1 - \tilde{Q}_t)\ell_t(k_t)\Big] \\
&\overset{(9)}{=} \mathbb{E}_{t-1}\Big[\tilde{Q}_t \sum_{k=1}^{K} p_t(k)\widehat{\ell}_t(k) + (1 - \tilde{Q}_t)\ell_t(k_t)\Big] \\
&= \mathbb{E}_{t-1}\big[\tilde{Q}_t\ell_t(k_t) + (1 - \tilde{Q}_t)\ell_t(k_t)\big] = \mathbb{E}_{t-1}\big[\ell_t(k_t)\big]\,.
\end{aligned}
$$

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

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

On the other hand,

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

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

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

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.

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

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

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

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;
- S-UCB: The sleeping UCB procedure proposed by Kleinberg et al. (2010) and designed for ordering regret with stochastic losses;

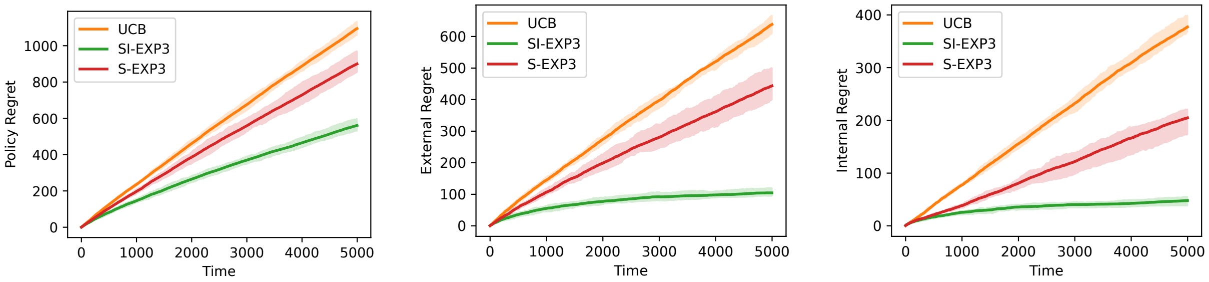

Figure 3: Random environment with dependence

- S-EXP3: The algorithm Sleeping-EXP3G designed by Saha et al. (2020) and designed for ordering regret with adversarial losses and stochastic sleeping.

Again, each experiment is run 20 times and the Policy, External, and Internal regrets are plotted in Fig. 3 and Fig. 4.

Random environment with dependence (Fig. 3) 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 (Fig. 4) This setup is similar to the Rock-Paper-Scissors environment of Sec. 5 but with $K = 5$ 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 $1 \le i < j \le K$, $G_{ij} \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}\big((-1, 1)\big)$, $G_{ii} = 1/2$ 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 $(\mathcal{A}_m)_{1 \le m \le 4}$ 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.

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 all the arms are available for the first bandit. All the arms except the one chosen by the first bandit are 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

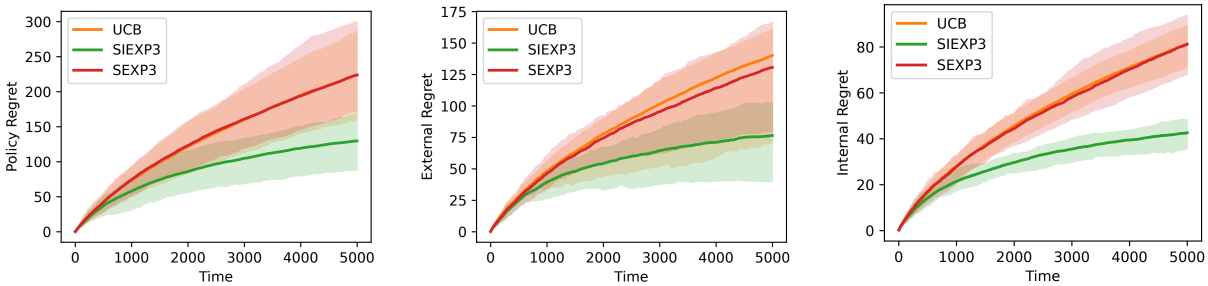

Figure 4: Random games

according to P and the loss is 1 for the bandit that chose this arm and 0 for the other bandit. In Fig. 5, we plot 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. 5, we plot the Policy regret of Sp-SIEXP3 and Sp-UCB and observe that in this relatively simple setting Sp-UCB outperforms Sp-SIEXP3.

Note despite its surprisingly good performance in Fig. 5, 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 algorithms as they are not suited for this setting.

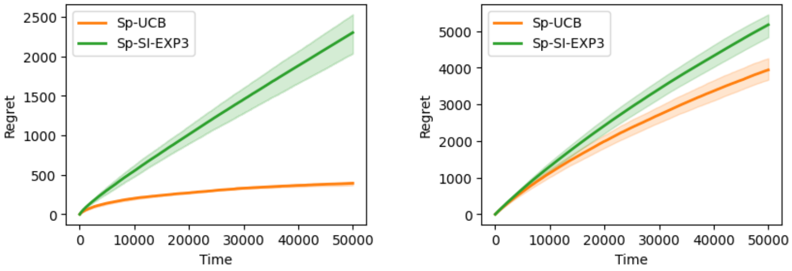

Figure 5: Dueling Bandits with non-repeating arms for $K = 4$ [Left] and $K = 30$ [Right] respectively. ($M = 5$).

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. 6, 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.

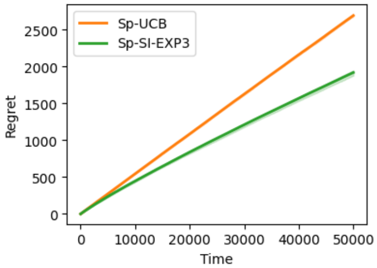

Figure 6: Preference Learning with Categories where Utilities depend on Availability.