Replicable Constrained Bandits

Matteo Bollini[*]
Politecnico di Milano
matteo.bollini@polimi.it

Gianmarco Genalti[*]
Politecnico di Milano
gianmarco.genalti@polimi.it

Francesco Emanuele Stradi[*]
Politecnico di Milano
francescoemanuele.stradi@polimi.it

Matteo Castiglioni
Politecnico di Milano
matteo.castiglioni@polimi.it

Alberto Marchesi
Politecnico di Milano
alberto.marchesi@polimi.it

Abstract

Algorithmic *replicability* has recently been introduced to address the need for reproducible experiments in machine learning. A *replicable online learning* algorithm is one that takes the same sequence of decisions across different executions in the same environment, with high probability. We initiate the study of algorithmic replicability in *constrained* MAB problems, where a learner interacts with an unknown stochastic environment for T rounds, seeking to maximize reward while satisfying multiple constraints. Our main result is that replicability can be achieved in constrained MABs. Specifically, we design replicable algorithms whose regret and constraint violation match those of non-replicable ones in terms of T. As a key step, we develop the first replicable UCB-like algorithm for *unconstrained* MABs, showing that algorithms that employ the optimism in-the-face-of-uncertainty principle can be replicable, a result that we believe is of independent interest.

1 Introduction

To produce scientific findings that are valid and reliable, the reproducibility of the experimental process is vital. However, in a 2016 survey involving 1,500 researchers, it was reported that more than 70% of respondents had attempted but failed to replicate another researcher's findings [Baker, 2016]. A more recent article [Ball, 2023] discussed how recent trends in AI research are creating a replicability crisis, which may have disruptive consequences in several scientific fields that are increasingly relying on AI tools. Indeed, major AI conferences have taken some countermeasures to respond to this crisis (see, *e.g.*, the NeurIPS 2019 Reproducibility Program [Pineau et al., 2021]).

The concept of *algorithmic replicability* in machine learning has recently been introduced to address reproducibility issues within a principled framework (see, *e.g.*, [Impagliazzo et al., 2022, Ghazi et al., 2021, Ahn et al., 2022]). Reproducibility in *online learning*, which is the focus of this paper, has been addressed by only a few works [Esfandiari et al., 2022, Ahmadi et al., 2024, Komiyama et al., 2024]. A *replicable online learning algorithm* is one that takes the *same* sequence of decisions over the rounds across different executions in the same environment, with high probability. Previous works have only considered replicability in the classic *multi-armed bandit* (MAB) problem, where the learner faces a sequence of decisions over T rounds. At each round t, the learner chooses an action a_t from a finite

[*]Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

set of K available actions and then receives a reward $r_t(a_t)$ for the selected action. The learner's goal is to maximize the cumulative reward or, equivalently, to minimize the *regret* with respect to always playing an optimal action in hindsight. Perhaps surprisingly, previous results [Esfandiari et al., 2022, Ahmadi et al., 2024] showed that, in stochastic MABs, not only do replicable algorithms exist, but they also achieve regret that is almost the same as that of non-replicable algorithms in terms of T.

A recent and growing line of research in online learning is concerned with *constraints* (see, *e.g.*, [Mannor et al., 2009, Castiglioni et al., 2022a, Pacchiano et al., 2021, Liu et al., 2021]). In a *constrained* MAB, the learner is subject to m cost constraints and, at each round t, also observes a cost $g_{t,i}(a_t)$ associated with each constraint i. The learner's goal thus becomes twofold: minimizing the regret while simultaneously ensuring that the constraints are satisfied. Constrained MABs find application in several real-world scenarios of interest, where it is often the case that additional requirements need to be satisfied during learning. For instance, this is the case in online auctions [Badanidiyuru et al., 2018]—where a bidder must *not* deplete the budget—wireless communication [Mannor et al., 2009], and safety-critical systems [Amodei et al., 2016]. Despite this, replicability in constrained MAB problems is still unexplored, to the best of our knowledge. In this paper, we initiate the study of *algorithmic replicability in stochastic constrained* MABs addressing the following question:

"Are there replicable algorithms for stochastic constrained MABs that (almost) match the guarantees achieved by non-replicable ones?"

We answer the question affirmatively, as described next.

1.1 Original Contributions

The main challenge in designing replicable algorithms for stochastic constrained MABs is that the state-of-the-art replicable algorithms for unconstrained settings [Esfandiari et al., 2022, Komiyama et al., 2024] cannot be easily adapted to handle constraints. This is because such algorithms implement complex action-elimination procedures. On the one hand, action-elimination approaches do not readily generalize to settings in which the learner must optimize over arbitrary polytopes contained in the simplex, and hence must select randomized strategies (rather than deterministic actions) throughout the learning process. This is the case of constrained MABs, where the safety constraints restrict the decision space to a generic polytope and the optimal solution is generally a non-uniform *distribution* over actions. On the other hand, those procedures generally fail to cope with the tension among multiple objectives that is inherently present in constrained settings. Intuitively, since constraint estimation can potentially classify high-reward action-selection strategies as infeasible, reward-based action elimination could prematurely discard optimal strategies. Even if such strategies were reintroduced later, they would still be underexplored, thus invalidating the standard analysis of arm-elimination algorithms, which relies on the uniform exploration of the active decision space.

As a first step toward answering our research question, we design a novel replicable algorithm for learning in stochastic (unconstrained) MABs. This is the first of its kind to be based on a UCB-like approach, which is crucial for extending the algorithm to the constrained case. The algorithm, called `DEBORA`, is based on a *dynamic epoch-based* approach. At the beginning of each epoch h, the algorithm selects an action using a UCB-style rule and plays it for all the rounds of the epoch. The epoch ends when the selected action has doubled the number of rounds in which it has been chosen. Crucially, the UCB-style action selection is performed using suitable *replicable estimators* and their confidence bounds [Impagliazzo et al., 2022]. The dynamic nature of `DEBORA` allows it to go beyond classic epoch-based approaches in which epochs are *fixed a priori*, thereby avoiding the use of an arm-elimination approach. Remarkably, the regret guarantees of `DEBORA` match, up to constant factors, those of state-of-the-art replicable algorithms [Esfandiari et al., 2022, Komiyama et al., 2024]. This may be of interest not only for constrained problems, but also for settings in which a UCB-like approach is desirable over action elimination (such as, *e.g.*, non-stationary settings).

Our results for constrained MABs are concerned with two classic settings, called the *soft* and *hard* constraints settings (see, *e.g.*, [Gangrade et al., 2024, Pacchiano et al., 2021]).

In the *soft* constraints setting, the goal is to minimize the regret while controlling the *cumulative positive violation* of the constraints. For this setting, we propose an extension of `DEBORA`, called `DEBORA-S`, which works similarly but, at each epoch, employs an action-selection (possibly randomized) strategy chosen via a UCB-style approach from a subset of strategies that *optimistically* satisfy

the constraints. Such a subset is computed using replicable estimators of the constraint costs. We prove that DEBORA-S attains both regret and constraint violation that grow as $\widetilde{\mathcal{O}}(\sqrt{T})$ in T.[2]

In the *hard* constraints setting, the goal is to minimize the regret while guaranteeing that the constraints are satisfied at every round. The algorithm for this setting, called DEBORA-H, builds on top of DEBORA-S by only assuming access to a suitable *strictly* constraint-satisfying strategy $\boldsymbol{x}^\diamond$, as is standard in the literature (see, *e.g.*, [Pacchiano et al., 2021]). The approach of DEBORA-S may fail in the hard constraints setting, since the strategy selected by the algorithm during an epoch may violate some constraints throughout all the rounds of the epoch. DEBORA-H resolves this issue by mixing such a strategy with $\boldsymbol{x}^\diamond$, in order to balance constraint satisfaction and optimism. The regret of DEBORA-H grows as $\widetilde{\mathcal{O}}(\sqrt{T})$ in T, while satisfying the constraints at every round with high probability.

In conclusion, in both the *soft* and *hard* constraints settings, the regret guarantees of our algorithms match, in terms of T, those of state-of-the-art non-replicable algorithms [Gangrade et al., 2024, Pacchiano et al., 2021]. This shows that replicability comes at almost no cost in constrained MABs.

1.2 Related Works

Recent works have addressed the problem of developing replicable regret minimization strategies in *unconstrained* settings, such as, in the standard MAB problems. We summarize the main existing results in the following, deferring a more detailed discussion to Appendix A.

[Esfandiari et al., 2022] is the first work to formally introduce the notion of *replicable bandit algorithm*. For the stochastic MAB problem, their best algorithm is guaranteed to suffer a cumulative regret of at most $\mathcal{O}(\frac{K^2}{\rho^2}\sum_{a\neq a^*}\frac{\ln T}{\Delta(a)})$, where $\Delta(a)$ is the sub-optimality of action a w.r.t. to the optimal one a^*. This bound can be easily converted to a worst-case form of $\widetilde{\mathcal{O}}(\rho^{-1}K^{\frac{3}{2}}\sqrt{T})$. Subsequently, Komiyama et al. [2024] provide an algorithm based on arm elimination called RSE, which is guaranteed to suffer a cumulative regret of at most $\mathcal{O}(\sum_{a\neq a^*}\frac{\ln T}{\Delta(a)}+\frac{K^2\ln\ln T}{\rho^2\Delta(a)})$, improving the previous result from [Esfandiari et al., 2022] by decoupling the standard MAB regret from an additional contribution due to the replicability, which is of lower order in T. [Ahmadi et al., 2024] expand the concept of replicability from stochastic bandits to their adversarial counterparts. They establish a regret lower bound of $\Omega(\rho^{-1}\sqrt{T})$ for both settings and provide algorithmic contributions for the *experts* setting.

Two additional, closely related lines of work are the literature on constrained MABs [Pacchiano et al., 2021, Castiglioni et al., 2022b] and on constrained MDPs [Altman, 1999, Efroni et al., 2020]. In both these frameworks, two kinds of results are generally presented. Some works focus on guaranteeing per-round satisfaction of the constraints, by assuming the knowledge of a strictly safe strategy $\boldsymbol{x}^\diamond$ [Pacchiano et al., 2021, Liu et al., 2021]. These works attain sublinear regret of order $\widetilde{\mathcal{O}}(1/\lambda\sqrt{T})$, where λ measures the margin by which $\boldsymbol{x}^\diamond$ satisfies the constraints, which is tight [Stradi et al., 2025a]. In contrast, many works focus on guaranteeing $\widetilde{\mathcal{O}}(\sqrt{T})$ regret while allowing the algorithm to suffer sublinear constraint violation, that is, attaining $\widetilde{O}(\sqrt{T})$ violation of the constraints [Efroni et al., 2020, Castiglioni et al., 2022b]. Notice that, in this setting, the $1/\lambda$ dependence in the regret bound can be removed and prior knowledge of a strictly safe strategy $\boldsymbol{x}^\diamond$ is not required. To the best of our knowledge, this is the first work focusing on both constraints and replicability.

2 Preliminaries

In this section, we review the key definitions and fundamental technical tools for constrained MAB problems, as well as the notion of replicability in learning.

2.1 Constrained MABs

In a *constrained* MAB, a learner faces a decision problem involving $K \in \mathbb{N}_{>0}$ actions over $T \in \mathbb{N}_{>0}$ rounds. At each round $t \in [T]$,[3] the learner selects an action $a_t \in [K]$ (possibly at random) and observes a reward $r_t(a_t) \in [0,1]$. Furthermore, the learner also observes a cost $g_{t,i}(a_t) \in [0,1]$ for each constraint $i \in [m]$. We consider a stochastic setting in which, for each action $a \in [K]$ and each

[2]The $\widetilde{\mathcal{O}}(\cdot)$ notation hides poly-logarithmic factors.

[3]We let $[n] := \{1, 2, \ldots, n\}$ be the first n natural numbers.

constraint $i \in [m]$, rewards and costs are independently drawn from fixed distributions $\mathcal{R}_a$ and $\mathcal{G}_{a,i}$, respectively. The means of these distributions are given by the unknown vectors $\boldsymbol{r}$ and $\boldsymbol{g}_i$, for all $i \in [m]$. We will refer to their a-th element as $\boldsymbol{r}(a)$ and $\boldsymbol{g}_i(a)$, respectively.

The learner selects an action $a_t \in [K]$ at random according to a probability distribution $\boldsymbol{x}_t \in \Delta_K$,[4] referred to as the learner's *strategy*. The learner chooses $\boldsymbol{x}_t$ by employing a learning algorithm, based on the history of their observations up to round $t-1$. We say that a constraint $i \in [m]$ is *satisfied* by a learner's strategy $\boldsymbol{x} \in \Delta_K$ if its expected cost does *not* exceed a known threshold $\alpha_i \in [0,1]$, *i.e.*, if $\boldsymbol{g}_i^\top \boldsymbol{x} \leq \alpha_i$. When a strategy satisfies all the m constraints, we say that it is *safe* for the learner.

As is customary in the literature (see, *e.g.*, [Castiglioni et al., 2022b]), we measure the performance of a learning algorithm over T rounds in terms of the regret with respect to an optimal *safe* strategy $\boldsymbol{x}^\star \in \Delta_K$, which is formally defined as follows:

$$\boldsymbol{x}^\star \in \operatorname*{argmax}_{\boldsymbol{x} \in \Delta_K} \boldsymbol{r}^\top \boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{g}_i^\top \boldsymbol{x} \leq \alpha_i \quad \forall i \in [m].$$

The *expected cumulative regret* (hereafter simply called *regret* for short) is then defined as follows:

$$R_T \coloneqq \sum_{t \in [T]} \boldsymbol{r}^\top \boldsymbol{x}^\star - \sum_{t \in [T]} \boldsymbol{r}^\top \boldsymbol{x}_t.$$

To evaluate the performance of a learning algorithm, two different settings are typically considered (see, *e.g.*, [Gangrade et al., 2024, Pacchiano et al., 2021]), which we describe below.

Soft Constraints In the *soft constraints* setting, the constraint satisfaction of a learning algorithm is measured by the cumulative *positive* constraint violation over the T rounds (called *violation* for short), which is defined as:

$$V_T \coloneqq \max_{i \in [m]} \sum_{t \in [T]} \max\left\{0, \boldsymbol{g}_i^\top \boldsymbol{x}_t - \alpha_i\right\}.$$

Notice that a negative constraint violation $\boldsymbol{g}_i^\top \boldsymbol{x}_t - \alpha_i < 0$ at round t (*i.e.*, a satisfaction of the constraint) is capped at 0, so that it does *not* cancel out a positive violation incurred in another round. In this setting, the goal is to design learning algorithms that achieve both sublinear regret and sublinear violation, *i.e.*, $R_T = o(T)$ and $V_T = o(T)$.

Hard Constraints In the *hard constraints* setting, the goal is to design learning algorithms that satisfy the constraints at every round $t \in [T]$ with high probability, while simultaneously attaining sublinear regret. Clearly, the initial rounds of learning are the most critical, since no information about the environment is available and, accordingly, it is *not* possible to select a safe strategy. For this reason, hard constraints settings can be addressed only under the following two (necessary) assumptions, which are common in the literature (see, *e.g.*, [Pacchiano et al., 2021, Liu et al., 2021]). These assumptions are needed to guarantee a minimum amount of exploration in the first rounds.

Assumption 2.1 (Slater's condition)**.** *There exists a strictly feasible strategy,* i.e.*, a strategy $\boldsymbol{x}^\diamond \in \Delta_K$ such that $\boldsymbol{g}_i^\top \boldsymbol{x}^\diamond < \alpha_i$ for all $i \in [m]$, where the inequalities are strict.*

Assumption 2.2. *The learner knows the strictly feasible strategy $\boldsymbol{x}^\diamond \coloneqq \operatorname{argmax}_{\boldsymbol{x} \in \Delta_K} \min_{i \in [m]} \{\alpha_i - \boldsymbol{g}_i^\top \boldsymbol{x}\}$ and the margins $\lambda_i \coloneqq \alpha_i - \boldsymbol{g}_i^\top \boldsymbol{x}^\diamond$ by which it satisfies each constraint $i \in [m]$.*[5] *We also let $\lambda_{\min} \coloneqq \min_{i \in [m]} \{\lambda_i\}$ be the margin by which $\boldsymbol{x}^\diamond$ satisfies all the constraints.*

2.2 Replicability in Learning

Our main goal in this paper is to develop *replicable* learning algorithms for constrained MABs. Intuitively, a replicable algorithm should suggest the same sequence of decisions across different realizations of the stochastic events when interacting with the same environment. In particular, a replicable learning algorithm takes as input a source of internal randomness ξ and, at each $t \in [T]$, outputs a strategy $\boldsymbol{x}_t \in \Delta_K$ based on the observations up to round $t-1$—including selected actions and collected rewards and costs. Moreover, the selected action $a_t \sim \boldsymbol{x}_t$ is sampled using the same internal randomness ξ. We adopt the definition of replicability of Esfandiari et al. [2022].

[4] We denote by Δ_n the $(n-1)$-dimensional simplex, namely, we let $\Delta_n \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n_{\geq 0} \mid \mathbf{1}^\top \boldsymbol{x} = 1\}$.

[5] Previous works (see, *e.g.*, [Pacchiano et al., 2021, Liu et al., 2021, Bernasconi et al., 2022]) usually assume knowledge of a generic strictly feasible strategy. For ease of presentation, in this work we assume knowledge of a strategy that satisfies the constraints as much as possible. Nonetheless, our results can be generalized to the case commonly considered in the literature.

Definition 2.3 (ρ-Replicable algorithm)**.** Fix $\rho \in (0,1)$. We say that a learning algorithm is ρ-*replicable* if, for any two runs of the algorithm on the same constrained MAB instance, when given as input the same source of internal randomness ξ, the following condition holds:

$$\mathbb{P}_{\xi}\left((\boldsymbol{x}_1^1, \boldsymbol{x}_2^1, \ldots, \boldsymbol{x}_T^1) = (\boldsymbol{x}_1^2, \boldsymbol{x}_2^2, \ldots, \boldsymbol{x}_T^2)\right) \geq 1-\rho,$$

where $\boldsymbol{x}_t^i \in \Delta_K$ is the strategy selected by the algorithm at round $t \in [T]$ during the i-th execution.

Remark 2.4. *By fixing the algorithm's internal randomness, we ensure that also the actions selected across two independent runs coincide with high probability. In particular, $a_t^1 = a_t^2$ when $a_t^1 \sim \boldsymbol{x}_t^1$ and $a_t^2 \sim \boldsymbol{x}_t^2$ and $\boldsymbol{x}_t^1 = \boldsymbol{x}_t^2$. Let us also remark that fixing the internal randomness of the algorithm is standard in the literature on replicable learning [Impagliazzo et al., 2022].*

2.2.1 Replicable Mean Estimation

In Algorithm 1, we provide the pseudocode of *Replicable Mean Estimator* (`RepMean`), namely, the algorithm originally proposed by Impagliazzo et al. [2022] to compute a ρ-replicable mean estimator.

Algorithm 1 `RepMean`

Require: $\mathcal{S} = \{s_i\}_{i\in[n]}, 0 < 2\delta < \rho < 1, \xi$

1: $\alpha \leftarrow \frac{2\delta}{1+\rho-2\delta}$
2: $\beta \sim \mathcal{U}_y\left([0,\alpha]\right)$
3: $R \leftarrow \{[0,\beta), [\beta,\beta+\alpha), \ldots, [\beta+k\alpha, 1)\}$
4: $\bar{r} \leftarrow \frac{1}{n}\sum_{i=1}^{n} s_i$
5: $z \leftarrow \lfloor \max\{\frac{\bar{r}-\beta}{\alpha}, 0\} \rfloor$
6: $\widehat{r} \leftarrow \min\{\beta + \left(z+\frac{1}{2}\right)\alpha, 1\}$
7: **Return** $\widehat{r}$

`RepMean` requires as input a set $\mathcal{S}$ of n samples drawn from a random variable with support in $[0,1]$, a tolerance $\delta \in (0,1)$, a replicability parameter $\rho \in (2\delta, 1)$, and a source of randomness ξ. The algorithm works by discretizing the interval $[0,1]$ into a finite set of possible outputs. To do so, it computes the size α of every interval (Line 1), a randomized offset β (Line 2), and the resulting grid R (Line 3). Then, `RepMean` computes the sample average $\bar{r}$ given $\mathcal{S}$ (Line 4) and identifies the index of the cell of the grid where it belongs (Line 5). Finally, the algorithm returns the middle point of the same cell as output (Line 6). We remark that β is chosen (at random) according to the internal randomness ξ, which ensures replicability. We formalize the theoretical guarantees provided by `RepMean` in the following lemma.

Lemma 2.5. *[Impagliazzo et al. [2022, Theorem 2.3]] Let $\mathcal{D}$ be a distribution with support in $[0,1]$ and mean r, and let $\epsilon, \rho', \delta' \in (0,1)$ with $\rho' > 2\delta'$. Then, there exists a ρ'-replicable algorithm that, given $\frac{2\log(2/\delta')}{\epsilon^2(\rho'-2\delta')^2}$ i.i.d. samples drawn from $\mathcal{D}$, outputs an estimator $\widehat{r}$ such that $\mathbb{P}(|\widehat{r}-r| \leq \epsilon) \geq 1-\delta'$.*

Lemma 2.5 implies that, given n i.i.d. samples from a distribution with mean r, it is possible to design a ρ-replicable estimator $\widehat{r}$ of r with precision $\epsilon = \sqrt{\frac{2\log(2/\delta')}{n(\rho'-2\delta')^2}}$, where δ' is the error probability.

3 Warm-Up: Unconstrained MABs

In this section, we introduce *Dynamic Epoch-Based Optimism with ReplicAbility* (`DEBORA`), a ρ-replicable algorithm for unconstrained MABs. The goal of this section is twofold. First, it serves as an introduction to the general algorithmic template that allows us to deal with constrained MABs while being ρ-replicable. Second, it shows how `DEBORA` itself attains regret of the same order as state-of-the-art approaches for unconstrained MABs, while simultaneously avoiding arm-elimination.

3.1 Beyond Arm Elimination

Our algorithm (`DEBORA`) leverages the well-known *optimism* principle [Auer et al., 2002]. Optimistic algorithms are the gold standard for most stochastic MAB problems. Indeed, as discussed in [Esfandiari et al., 2022], a straightforward approach to obtain a ρ-replicable no-regret algorithm for stochastic MABs is to plug the replicable estimator from [Impagliazzo et al., 2022] into the `UCB1` algorithm of [Auer et al., 2002] and adjust the confidence intervals accordingly. However, requiring all confidence intervals to be valid at every round, together with the estimators being ρ-replicable, leads to a statistical complexity of K^2T^2, which is excessive for any learning algorithm. This issue has been addressed by splitting the learning process into *fixed* epochs of increasing length, computing statistics only between consecutive epochs [Esfandiari et al., 2022, Komiyama et al., 2024]. Crucially, a naive optimistic strategy may fail to guarantee sublinear regret if a suboptimal action is selected at

the start of the last (and longest) epoch. As a consequence, they rely on *eliminating* actions, *i.e.*, they permanently discard some actions from the sampling pool at the end of each epoch.

Unfortunately, even though action elimination algorithms can attain nearly optimal regret in standard MABs, they become nonviable in more general settings, such as constrained MABs and nonstationary MABs, where an action discarded early can later turn out to be optimal. For instance, in the constrained MABs case, action elimination procedures generally fail to cope with the tension among multiple objectives. Indeed, constraints estimation may label high-reward strategies as *non-safe*, whereas reward based elimination may prematurely discard strategies which are truly optimal compared to the other *safe* strategies. Reintroducing such strategies at a later stage does not resolve the issue, since they remain insufficiently explored, thereby breaking the standard action elimination analysis that hinges on (approximately) uniform exploration among active actions. Additionally, action elimination approaches do not easily extend to settings where *strategies* are selected in place of actions and algorithms typically optimize over arbitrary polytopes contained in the simplex.

Crucially, DEBORA is the first replicable algorithm to achieve nearly optimal regret guarantees while avoiding any arm elimination procedure. Indeed, we show how a *dynamic* epoch-based protocol allows one to employ a UCB-style approach while still guaranteeing replicability.

3.2 Algorithm

In Algorithm 2, we report the pseudocode of DEBORA. At a high level, the algorithm performs a UCB-based action selection at the beginning of each epoch and continues to choose the selected action throughout the entire epoch. Unlike existing approaches for replicable MABs, the epoch size is *not* fixed *a priori*. Instead, it is dynamically chosen given the history of both observations and decisions. This approach allows DEBORA to operate without eliminating actions.

Algorithm 2 DEBORA

Require: $T \in \mathbb{N}_{>0}$, $K > 1$, $0 < 2\delta < \rho < 1$, ξ.

1: $\delta' \leftarrow \frac{\delta}{2K\lceil \log(T) \rceil}$, $\rho' \leftarrow \frac{\rho}{2K\lceil \log(T) \rceil}$
2: $h \leftarrow 0$
3: $N_1(a), \widetilde{N}_0(a) \leftarrow 0$, $\widehat{\boldsymbol{r}}_0(a) \leftarrow \frac{1}{2}\ \forall a \in [K]$
4: $a^\circ_0 \leftarrow \operatorname{argmax}_{a\in[K]} \widehat{\boldsymbol{r}}_0(a) + \boldsymbol{\zeta}_0(a)$
5: **for** $t = 1, \dots, T$ **do**
6: **if** $N_t(a^\circ_h) \geq \max\{2\widetilde{N}_h(a^\circ_h), 1\}$ **then**
7: $h \leftarrow h + 1$
8: $\widetilde{N}_h(a) \leftarrow N_t(a) \quad \forall a \in [K]$
9: $\mathcal{S}_{a^\circ_{h-1}}(t) \leftarrow \{r_\tau(a) \mid \tau < t, a_\tau = a\}$
10: $\widehat{\boldsymbol{r}}_h(a^\circ_{h-1}) \leftarrow$ RepMean$(\mathcal{S}_{a^\circ_{h-1}}(t), \delta', \rho', \xi)$
11: $\widehat{\boldsymbol{r}}_h(a) \leftarrow \widehat{\boldsymbol{r}}_{h-1}(a) \quad \forall a \in [K]$
12: $a^\circ_h \leftarrow \operatorname{argmax}_{a\in[K]} \widehat{\boldsymbol{r}}_h(a) + \boldsymbol{\zeta}_h(a)$
13: Select $a_t = a^\circ_h$ and receive $r_t(a_t)$
14: $N_{t+1}(a) \leftarrow N_t(a) + \mathbb{1}\{a = a_t\}\ \forall a \in [K]$

Algorithm 2 starts by initializing the necessary quantities for epoch-based and UCB-style algorithms (Lines 1–4). In particular, at round $t \in [T]$, it keeps track of both $N_t(a) := \sum_{\tau=1}^{t-1} \mathbb{1}\{a_\tau = a\}$, *i.e.*, the number of times action $a \in [K]$ has been selected up to (but excluding) the current round t, and $\widetilde{N}_h(a) := N_{t_h}(a)$, *i.e.*, the number of times action $a \in [K]$ has been chosen up to round t_h, which is the starting round of the current epoch h. At the beginning of each round, these two quantities are compared (Line 6), and if the currently selected action has doubled its counter $N_t(a)$ since the beginning of the current epoch, the epoch terminates (Line 7) and a UCB-based selection of the action for the next epoch is performed (Lines 8–12). Note that DEBORA uses its own internal source of randomness ξ, provided in input, to compute the replicable mean estimator. Specifically, at the beginning of epoch h, DEBORA computes a ρ'-replicable mean estimator $\widehat{\boldsymbol{r}}_h(a^\circ_{h-1})$ for the action a°_{h-1} selected in the previous epoch, together with the corresponding confidence interval width. For every action $a \in [K]$ and epoch h, this radius is defined as:

$$\boldsymbol{\zeta}_h(a) := \sqrt{\frac{2\ln(2/\delta')}{\max\{\widetilde{N}_h(a), 1\}(\rho' - 2\delta')^2}}. \tag{1}$$

For all $a \neq a^\circ_{h-1}$, the algorithm keeps the same estimator and confidence interval width as in the previous epochs, *i.e.*, $\boldsymbol{\zeta}_{h-1}(a) = \boldsymbol{\zeta}_h(a)$, since no additional samples have arrived in the meantime.

Throughout an epoch h, the selected action is continuously chosen until the epoch-termination condition is met (Lines 13–14). Notice that the decision of the algorithm is deterministic when conditioned on the past history, as no randomization occurs; thus, we formally have that $\boldsymbol{x}_t = \boldsymbol{e}_{a^\circ_h}$ during the epoch h, where $\boldsymbol{e}_{a^\circ_h}$ is the a°_h-th element of the canonical basis.

The following lemma formally states the two key properties of DEBORA: an upper bound on the number of epochs, which we denote with H, and its ρ-replicability.

Lemma 3.1. DEBORA *is a ρ-replicable algorithm and executes $H \leq K\lceil \log_2(T) \rceil$ epochs.*

The two properties are linked, as replicability is established by applying a union bound over the number of epochs. The complete proof can be found in Appendix C.

Now, we bound the expected regret of DEBORA.

Theorem 3.2. *Let $\delta \in (0,1)$ and $\rho \in (2\delta, 1)$. Consider an instance of the stochastic K-armed MAB problem defined by the reward gaps $\Delta(a) := \arg\max_{\bar{a}\in[K]} \boldsymbol{r}(\bar{a}) - \boldsymbol{r}(a)$ for all $a \in [K]$. Then, with probability at least $1-\delta$,* DEBORA *suffers an expected cumulative regret bounded as:*

$$R_T \leq \widetilde{\mathcal{O}}\left(\frac{1}{(\rho-\delta)^2} \log\left(\frac{1}{\delta}\right) K^2 \log^2(T) \sum_{a \neq a^\star} \frac{1}{\Delta(a)} \right).$$

Moreover, for every instance, with probability at least $1-\delta$, the regret can be bounded as:

$$R_T \leq \widetilde{\mathcal{O}}\left(\frac{1}{(\rho-\delta)} K \sqrt{KT \log\left(\frac{1}{\delta}\right)} \right).$$

The instance-dependent guarantee of DEBORA matches, up to constant factors, that of Algorithm 1 in [Esfandiari et al., 2022]. To the best of our knowledge, the only work providing worst-case guarantees for replicable MABs is [Komiyama et al., 2024]. DEBORA provides worst-case guarantees matching, up to universal constants, those of the main algorithm presented in their Theorem 11. We remark that, among these results, ours are the only guarantees that hold with high probability, rather than in expectation. Finally, we underline that an existing lower bound for constrained MABs shows that the $1/\rho$ dependence in the regret bound is unavoidable [Ahmadi et al., 2024].

4 Replicability in MABs With Soft Constraints

In this section, we address the problem of designing a ρ-replicable learning algorithm for stochastic MABs with *soft constraints*. Our approach, called DEBORA-S, builds on top of the previously introduced DEBORA, integrating additional tools to control the violation incurred by the algorithm by restricting the decision space to a specific subset of strategies every epoch.

4.1 Algorithm

Algorithm 3 provides the pseudocode of DEBORA-S. This algorithm builds on DEBORA, sharing the epoch-based UCB approach. The presence of constraints precludes the algorithm from choosing the action that maximizes the UCB index, and instead restricts the decision space from the full simplex to a convex polytope. In Algorithm 3, we highlight in light-blue the block containing the set of instructions necessary to control the violation. At the beginning of every epoch h, for every constraint $i \in [m]$ and action $a \in [K]$, DEBORA-S estimates the average cost via a ρ'-replicable estimator $\widehat{\boldsymbol{g}}_{h,i}(a)$ (Line 12). Then, these estimates are used to compute an optimistic *safe* decision space $\underline{\mathcal{X}}_h$, composed of the strategies $\boldsymbol{x} \in \Delta_K$ such that:

$$(\widehat{\boldsymbol{g}}_{h,i} - \boldsymbol{\zeta}_h)^\top \boldsymbol{x} \leq \alpha_i \ \forall i \in [m], \tag{2}$$

where the estimation uncertainty controls the expected violation. The optimistic safe decision space computed by DEBORA-S is guaranteed to contain the true safe decision space with high probability, as stated in the following.

Lemma 4.1. *Let $\mathcal{X}^\star := \{\boldsymbol{x} \in \Delta_K \mid \boldsymbol{g}_i^\top \boldsymbol{x} \leq \alpha_i \ \forall i \in [m]\}$ be the set of safe strategies. With probability at least $1-\delta$, for every epoch h it holds that $\mathcal{X}^\star \subseteq \underline{\mathcal{X}}_h$.*

Notice that, as the number of epochs increases, the set $\underline{\mathcal{X}}_h$ *shrinks* toward the set $\mathcal{X}^\star$ of truly safe strategies. In this way, the incurred per-round violation becomes increasingly close to that of safe strategies. Finally, the proposed strategy $\boldsymbol{x}_h^\circ$ maximizes the optimistic reward within $\underline{\mathcal{X}}_h$ (Line 14).

The aforementioned reasoning is one the main motivation behind the use of UCB-style algorithms for constrained MABs. Unlike unconstrained settings, where the learner deals with a single objective and can discard arms based on a unique reward signal, the constrained setting presents a more complex

trade-off. Indeed, standard arm elimination techniques fail to cope with the tension between multiple objectives. For instance, since constraints estimation can potentially classify high-reward regions of the simplex as infeasible, reward-based elimination could prematurely discard optimal safe solutions that appear suboptimal in terms of rewards. Additionally, even if such solutions were reintroduced at a later stage of the learning process, the corresponding region of the simplex would still not have been explored sufficiently. This invalidates the standard analysis of arm elimination algorithms, which typically assumes that all active actions are sampled roughly uniformly.

It is worth noticing some additional differences from DEBORA, which are highlighted in green in Algorithm 3. Differently from the unconstrained setting, here the algorithm chooses a strategy, and the selected action is sampled accordingly (Line 13). This aspect further motivates the use of UCB-style approaches, which—unlike action elimination methods—can be easily extended to operate over an arbitrary space of strategies. Moreover, the strategy selection implies that, during an epoch, every action with positive probability can be selected. As a consequence, the epoch-termination condition (Line 6) accounts for every action, and the average reward estimator has to be computed for every action in every epoch (Line 10), or at least for every action that has been played at least once.

Algorithm 3 DEBORA-S

Require: $T \in \mathbb{N}_{>0}$, $K > 1$, $0 < 2\delta < \rho < 1$, ξ

1: $\delta' \leftarrow \frac{\delta}{2(m+1)K\lceil \log_2(T)\rceil}$, $\rho' \leftarrow \frac{\rho}{2(m+1)K\lceil \log_2(T)\rceil}$
2: $h \leftarrow 0$
3: $N_1(a) \leftarrow 0,\ \widetilde{N}_0(a) \leftarrow 0,\ \widehat{\boldsymbol{r}}_0(a) \leftarrow \frac{1}{2} \quad \forall a \in [K]$
4: $\boldsymbol{x}_0^\circ \leftarrow \operatorname{argmax}_{\boldsymbol{x}\in\Delta_K} (\widehat{\boldsymbol{r}}_0 + \boldsymbol{\zeta}_0)^\top \boldsymbol{x}$
5: **for** $t = 1, \dots, T$ **do**
6: **if** $\exists a \in [K] : N_t(a) \geq \max\{2\widetilde{N}_h(a), 1\}$ **then**
7: $h \leftarrow h + 1$
8: $\widetilde{N}_h(a) \leftarrow N_t(a) \quad \forall a \in [K]$
9: $\mathcal{S}^{\mathrm{r}}_a(t) \leftarrow \{r_\tau(a) \mid \tau < t, a_\tau = a\}\ \forall a \in [K]$
10: $\widehat{\boldsymbol{r}}_h(a) \leftarrow \texttt{RepMean}(\mathcal{S}^{\mathrm{r}}_a(t), \delta', \rho', \xi)\ \forall a \in [K]$
11: $\mathcal{S}^{\mathrm{g}}_{a,i}(t) \leftarrow \{g_{\tau,i}(a) \mid \tau < t, a_\tau = a\}\ \forall a \in [K]$
12: $\widehat{\boldsymbol{g}}_{h,i}(a) \leftarrow \texttt{RepMean}(\mathcal{S}^{\mathrm{g}}_{a,i}(t), \delta', \rho', \xi)\ \forall a \in [K]$
13: $\underline{\mathcal{X}}_h \leftarrow \{\boldsymbol{x} \in \Delta_K \mid \text{Equation (2) holds}\}$
14: $\boldsymbol{x}_h^\circ \leftarrow \operatorname{argmax}_{\boldsymbol{x}\in\underline{\mathcal{X}}_h} (\widehat{\boldsymbol{r}}_h + \boldsymbol{\zeta}_h)^\top \boldsymbol{x}$
15: Select $a_t \sim \boldsymbol{x}_h^\circ$, receive $r_t(a_t), g_{t,i}(a_t)\ \forall i \in [m]$
16: $N_{t+1}(a) \leftarrow N_t(a) + \mathbb{1}\{a = a_t\} \quad \forall a \in [K]$

As for DEBORA, the number and the duration of the phases are not decided beforehand, but it is possible to show that there are at most $H \leq \mathcal{O}(K \log T)$ phases. In every epoch, DEBORA-S computes $(m+1)K$ estimators, and the algorithm is ρ-replicable as long as each estimator is ρ'-replicable, with $\rho' \leq \rho/\big((m+1)K \log T\big)$. Analogously to DEBORA, we summarize the bound on the number of epochs and the ρ-replicability of DEBORA-S in a single lemma.

Lemma 4.2. DEBORA-S *is a ρ-replicable algorithm and executes $H \leq K\lceil \log_2(T)\rceil$ epochs.*

We are now ready to state the main results of this section, which are an upper bound on the regret and an upper bound on the cumulative violation of DEBORA-S. Both results hold with high probability.

Theorem 4.3. *Let $\delta \in (0,1)$ and $\rho \in (2\delta, 1)$. Then, with probability at least $1-\delta$,* DEBORA-S *suffers an expected cumulative regret bounded as:*

$$R_T \leq \widetilde{\mathcal{O}}\left(\frac{1}{\rho - 2\delta} \log\left(\frac{1}{\delta}\right) mK^2\sqrt{KT}\right),$$

and a cumulative positive constraint violation bounded as:

$$V_T \leq \widetilde{\mathcal{O}}\left(\frac{1}{\rho - 2\delta} \log\left(\frac{1}{\delta}\right) mK^2\sqrt{KT}\right).$$

Comparing the regret bound of DEBORA with that of DEBORA-S, we obtain the same dependence on T and ρ, while we pay an additional factor m due to the constraints and a factor K due to the randomized strategies. The upper bound on the cumulative violation is of the same order as that on the regret, and is thus sublinear.

Compared to the state-of-the-art results for constrained MABs (and constrained MDPs) with *soft* constraints provided in [Efroni et al., 2020], our algorithm attains the optimal rate in T for both regret and violation. Finally, we remark that, since constrained MABs are a generalization of unconstrained ones, the dependence in $1/\rho$ is unavoidable for constrained settings too.

5 Replicability in MABs with Hard Constraints

In this section, we design a ρ-replicable learning algorithm for stochastic MABs with *hard constraints*. We propose DEBORA-H, an algorithm that puts an additional layer over DEBORA-S, by interpolating the proposed strategy with a provided strictly safe strategy in order to play a safe strategy every round.

5.1 Algorithm

Algorithm 4 reports the pseudocode of DEBORA-H, a ρ-replicable algorithm for MABs with hard constraints. DEBORA-H builds directly on top of DEBORA-S, by adding a block of instructions (highlighted in pink) that uses Slater's condition and the strictly safe strategy $\boldsymbol{x}^\diamond$, which is required as an input, in order to enforce safety over the proposed strategy $\boldsymbol{x}_h^\circ$ for every epoch h.

Let us remark that optimizing inside the safe set employed by DEBORA-S would result in a strategy $\widetilde{\boldsymbol{x}}_h$ (Line 14) that may be unsafe during each of the rounds of epoch h. For instance, in the first rounds, the optimistic safe set employed in DEBORA-S is equivalent to the entire simplex. In such a case, the first strategy played by DEBORA-S can be trivially unsafe. Furthermore, even in later rounds, DEBORA-S tends to play at the frontier of the decision space, due to the linearity of the objective function, which is included thank to the optimism. Accordingly, since the frontier is the unsafest part of the estimated decision space (this is particularly true when the optimal solution is mixed), DEBORA-S can play unsafe strategies even in the last rounds of the learning dynamics.

Algorithm 4 DEBORA-H

Require: $T \in \mathbb{N}_{>0}$, $K > 1$, $\delta > 0$, $\rho > 2\delta$, ξ, strictly safe strategy $\boldsymbol{x}^\diamond$

1: $\delta' \leftarrow \frac{\delta}{2(m+1)K\lceil\log_2(T)\rceil}$, $\rho' \leftarrow \frac{\rho}{2(m+1)K\lceil\log_2(T)\rceil}$
2: $h \leftarrow 0$
3: $N_1(a) \leftarrow 0$, $\widetilde{N}_0(a) \leftarrow 0$, $\widehat{\boldsymbol{r}}_0(a) \leftarrow \frac{1}{2}$ $\quad \forall a \in [K]$
4: $\sigma_0 \leftarrow \max_{i\in[m]} \frac{1-\alpha_i}{1-\alpha_i+\lambda_i}$, $\boldsymbol{x}_0^\circ \leftarrow \sigma_h \boldsymbol{x}^\diamond + (1-\sigma_h)\widetilde{\boldsymbol{x}}_h$
5: **for** $t = 1, \dots, T$ **do**
6: **if** $\exists a \in [K] : N_t(a) \geq \max\{2\widetilde{N}_h(a), 1\}$ **then**
7: $h \leftarrow h + 1$
8: $\widetilde{N}_h(a) \leftarrow N_t(a)$ $\quad \forall a \in [K]$
9: $\mathcal{S}_a^{\mathrm{r}}(t) \leftarrow \{r_\tau(a) \mid \tau < t, a_\tau = a\}$ $\forall a \in [K]$
10: $\widehat{\boldsymbol{r}}_h(a) \leftarrow \texttt{RepMean}(\mathcal{S}_a^{\mathrm{r}}(t), \delta', \rho', \xi)$ $\forall a \in [K]$
11: $\mathcal{S}_{a,i}^{\mathrm{g}}(t) \leftarrow \{g_{\tau,i}(a) \mid \tau < t, a_\tau = a\}$ $\forall a \in [K]$
12: $\widehat{\boldsymbol{g}}_{h,i}(a) \leftarrow \texttt{RepMean}(\mathcal{S}_{a,i}^{\mathrm{g}}(t), \delta', \rho', \xi)$ $\forall a \in [K]$
13: $\underline{\mathcal{X}}_h \leftarrow \{\boldsymbol{x} \in \Delta_K \mid \text{Equation (2) holds}\}$
14: $\widetilde{\boldsymbol{x}}_h \leftarrow \operatorname{argmax}_{\boldsymbol{x}\in\underline{\mathcal{X}}_h} (\widehat{\boldsymbol{r}}_h + \boldsymbol{\zeta}_h)^\top \boldsymbol{x}$
15: $\mathcal{V}_h \leftarrow \{i \in [m] \mid (\widehat{\boldsymbol{g}}_{h,i} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h > \alpha_i\}$
16: Compute σ_h according to Equation (4)
17: $\boldsymbol{x}_h^\circ \leftarrow \sigma_h \boldsymbol{x}^\diamond + (1-\sigma_h)\widetilde{\boldsymbol{x}}_h$
18: Select $a_t \sim \boldsymbol{x}_h^\circ$, receive $r_t(a_t), g_{t,i}(a_t)\ \forall i \in [m]$
19: $N_{t+1}(a) \leftarrow N_t(a) + \mathbb{1}\{a = a_t\}$ $\quad \forall a \in [K]$

To tackle this problem, we define the set of constraints:

$$\mathcal{V}_h := \{i \in [m] \,|\, (\widehat{\boldsymbol{g}}_{h,i} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h > \alpha_i\} \quad (3)$$

which the strategy $\widetilde{\boldsymbol{x}}_h$ *may* violate during epoch h (Line 15). This is done to adopt a *pessimistic* perspective by checking which constraints might be violated with probability at least $1-\delta$ during epoch h, and by *how much* they are violated in the worst-case. If there exists one or more constraints that are at risk of violation, DEBORA-H computes a combination factor:

$$\sigma_h = \max_{i\in\mathcal{V}_h} \frac{\min\{(\widehat{\boldsymbol{g}}_{h,i} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h, 1\} - \alpha_i}{\min\{(\widehat{\boldsymbol{g}}_{h,i} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h, 1\} - \alpha_i + \lambda_i}, \quad (4)$$

by weighting the worst-case violation of a constraint over itself plus the margin of the strictly safe strategy $\boldsymbol{x}^\diamond$ for that specific constraint, and taking the maximum of this weight over the constraints in $\mathcal{V}_h$ (Line 16). If no constraint is considered at risk of violation (*i.e.*, $\mathcal{V}_h = \varnothing$), then DEBORA-H sets $\sigma_h = 0$.[6] The actual strategy $\boldsymbol{x}_h^\star$ employed by DEBORA-H during epoch h is computed as a linear combination of $\boldsymbol{x}^\diamond$ and $\widetilde{\boldsymbol{x}}_h$, weighted by σ_h, formally, $\boldsymbol{x}_h^\star = \sigma_h \boldsymbol{x}^\diamond + (1-\sigma_h)\widetilde{\boldsymbol{x}}_h$ is selected. Notice that this procedure is also applied to the very first strategy played by DEBORA-H (Line 4), ensuring safety since the beginning of the learning process. Thanks to this procedure, we can prove that DEBORA-H is a safe algorithm for MABs with hard constraints.

Lemma 5.1. *With a probability of at least $1-\delta$,* DEBORA-H *plays a safe strategy $\boldsymbol{x}_t$ for every $t \in [T]$.*

As for DEBORA and DEBORA-S, DEBORA-H executed at most $\mathcal{O}(KT\log(T))$ epochs and ρ' is chosen small enough so as to guarantee ρ-replicability. We state both results in a single lemma.

[6] We use the convention that the maximum over an empty set is zero, so that Equation (4) is well-defined when $\mathcal{V}_h = \emptyset$.

Lemma 5.2. DEBORA-H *is a ρ-replicable algorithm and executes $H \leq K\lceil \log_2(T) \rceil$ epochs.*

Finally, we provide a high probability bound on the regret of DEBORA-H.

Theorem 5.3. *Let $\delta \in (0,1)$ and $\rho \in (2\delta, 1)$. Then, with probability at least $1-\delta$,* DEBORA-H *suffers an expected cumulative regret bounded as:*

$$R_T \leq \widetilde{\mathcal{O}}\left(\frac{1}{\lambda_{\min}(\rho - 2\delta)} \log\left(\frac{1}{\delta}\right) mK^2\sqrt{KT} + \gamma\right)$$

where γ does not depend on T.

Comparing the performance of Algorithm 4 with the state-of-the-art, our algorithm matches the regret bound of [Pacchiano et al., 2021] for the MAB case, in terms of T. To conclude, we underline that the $1/\lambda_{\min}$ dependence in the regret bound is provably necessary [Stradi et al., 2025a].

Acknowledgements

This work was supported by the EU Horizon project ELIAS (European Lighthouse of AI for Sustainability, No. 101120237).

References

Saba Ahmadi, Siddharth Bhandari, and Avrim Blum. Replicable online learning. *arXiv preprint arXiv:2411.13730*, 2024.

Kwangjun Ahn, Prateek Jain, Ziwei Ji, Satyen Kale, Praneeth Netrapalli, and Gil I Shamir. Reproducibility in optimization: Theoretical framework and limits. *Advances in Neural Information Processing Systems*, 35:18022–18033, 2022.

E. Altman. *Constrained Markov Decision Processes*. Chapman and Hall, 1999.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

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

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *Journal of the ACM (JACM)*, 65(3):1–55, 2018.

Qinbo Bai, Vaneet Aggarwal, and Ather Gattami. Provably efficient model-free algorithm for mdps with peak constraints. *arXiv preprint arXiv:2003.05555*, 2020.

Monya Baker. 1,500 scientists lift the lid on reproducibility, 2016.

Philip Ball. Is ai leading to a reproducibility crisis in science? *Nature*, 624(7990):22–25, 2023.

Martino Bernasconi, Federico Cacciamani, Matteo Castiglioni, Alberto Marchesi, Nicola Gatti, and Francesco Trovò. Safe learning in tree-form sequential decision making: Handling hard and soft constraints. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 1854–1873. PMLR, 2022.

Matteo Castiglioni, Andrea Celli, and Christian Kroer. Online learning with knapsacks: the best of both worlds. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2767–2783. PMLR, 17–23 Jul 2022a. URL https://proceedings.mlr.press/v162/castiglioni22a.html.

Matteo Castiglioni, Andrea Celli, Alberto Marchesi, Giulia Romano, and Nicola Gatti. A unifying framework for online optimization with long-term constraints. *Advances in Neural Information Processing Systems*, 35:33589–33602, 2022b.

Tianrui Chen, Aditya Gangrade, and Venkatesh Saligrama. Strategies for safe multi-armed bandits with logarithmic regret and risk. In *International Conference on Machine Learning*, pages 3123–3148. PMLR, 2022.

Yonathan Efroni, Shie Mannor, and Matteo Pirotta. Exploration-exploitation in constrained mdps, 2020. URL https://arxiv.org/abs/2003.02189.

Hossein Esfandiari, Alkis Kalavasis, Amin Karbasi, Andreas Krause, Vahab Mirrokni, and Grigoris Velegkas. Replicable bandits. *arXiv preprint arXiv:2210.01898*, 2022.

Aditya Gangrade, Tianrui Chen, and Venkatesh Saligrama. Safe linear bandits over unknown polytopes. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1755–1795. PMLR, 2024.

Gianmarco Genalti, Francesco Emanuele Stradi, Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Data-dependent regret bounds for constrained MABs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=vWeyFCtYxx.

Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. User-level differentially private learning via correlated sampling. *Advances in Neural Information Processing Systems*, 34:20172–20184, 2021.

Russell Impagliazzo, Rex Lei, Toniann Pitassi, and Jessica Sorrell. Reproducibility in learning. In *Proceedings of the 54th annual ACM SIGACT symposium on theory of computing*, pages 818–831, 2022.

Junpei Komiyama, Shinji Ito, Yuichi Yoshida, and Souta Koshino. Replicability is asymptotically free in multi-armed bandits. *arXiv preprint arXiv:2402.07391*, 2024.

Nikolaos Liakopoulos, Apostolos Destounis, Georgios Paschos, Thrasyvoulos Spyropoulos, and Panayotis Mertikopoulos. Cautious regret minimization: Online optimization with long-term budget constraints. In *International Conference on Machine Learning*, pages 3944–3952. PMLR, 2019.

Tao Liu, Ruida Zhou, Dileep Kalathil, Panganamala Kumar, and Chao Tian. Learning policies with zero or bounded constraint violation for constrained mdps. *Advances in Neural Information Processing Systems*, 34:17183–17193, 2021.

Shie Mannor, John N. Tsitsiklis, and Jia Yuan Yu. Online learning with sample path constraints. *Journal of Machine Learning Research*, 10(20):569–590, 2009. URL http://jmlr.org/papers/v10/mannor09a.html.

Adrian Müller, Pragnya Alatur, Volkan Cevher, Giorgia Ramponi, and Niao He. Truly no-regret learning in constrained mdps. In *Proceedings of the 41st International Conference on Machine Learning*, pages 36605–36653, 2024.

Aldo Pacchiano, Mohammad Ghavamzadeh, Peter Bartlett, and Heinrich Jiang. Stochastic bandits with linear constraints. In *International conference on artificial intelligence and statistics*, pages 2827–2835. PMLR, 2021.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program). *Journal of machine learning research*, 22(164):1–20, 2021.

Shuang Qiu, Xiaohan Wei, Zhuoran Yang, Jieping Ye, and Zhaoran Wang. Upper confidence primal-dual reinforcement learning for cmdp with adversarial loss. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15277–15287. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/ae95296e27d7f695f891cd26b4f37078-Paper.pdf.

Francesco Emanuele Stradi, Jacopo Germano, Gianmarco Genalti, Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Online learning in CMDPs: Handling stochastic and adversarial constraints. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 46692–46721. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/stradi24a.html.

Francesco Emanuele Stradi, Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Learning adversarial mdps with stochastic hard constraints. In *Forty-second International Conference on Machine Learning*, 2025a.

Francesco Emanuele Stradi, Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Optimal strong regret and violation in constrained mdps via policy optimization. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Francesco Emanuele Stradi, Anna Lunghi, Matteo Castiglioni, Alberto Marchesi, and Nicola Gatti. Policy optimization for cmdps with bandit feedback: Learning stochastic and adversarial constraints. In *Forty-second International Conference on Machine Learning*, 2025c.

Xiaohan Wei, Hao Yu, and Michael J. Neely. Online learning in weakly coupled markov decision processes: A convergence time study. *Proc. ACM Meas. Anal. Comput. Syst.*, 2(1), apr 2018. doi: 10.1145/3179415. URL https://doi.org/10.1145/3179415.

Liyuan Zheng and Lillian Ratliff. Constrained upper confidence reinforcement learning. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger, editors, *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 620–629. PMLR, 10–11 Jun 2020. URL https://proceedings.mlr.press/v120/zheng20a.html.

A Additional Related Works

In this section, we provide additional discussion on the constrained online learning literature.

Online learning with *unknown* constraints has been extensively studied (see, *e.g.*, [Mannor et al., 2009, Liakopoulos et al., 2019, Castiglioni et al., 2022b]). Two main settings are typically considered. In *soft-constraint* regimes (see, *e.g.*, [Chen et al., 2022]), the goal is to ensure that the algorithm's cumulative constraint violations grow sub-linearly. In *hard-constraint* regimes, instead, algorithms are required to satisfy the constraints at every round, usually under the assumption that a strictly feasible decision is known in advance (see, *e.g.*, [Pacchiano et al., 2021]). Both soft and hard constraints have also been extended to settings that are more challenging than MABs, such as linear bandits (see, *e.g.*, [Gangrade et al., 2024]) and games (see, *e.g.*, [Bernasconi et al., 2022]). Finally, stronger notions of regret bound (such as data-dependent ones) have been provided for both *soft* and *hard* constraints in [Genalti et al., 2025]. To the best of our knowledge, none of these contributions in the constrained online learning literature focus on replicability.

A strongly related body of work is the one on online learning in constrained Markov decision processes (see, *e.g.*, [Wei et al., 2018, Zheng and Ratliff, 2020, Bai et al., 2020, Efroni et al., 2020, Qiu et al., 2020, Stradi et al., 2024, 2025c, Müller et al., 2024, Stradi et al., 2025b]). The most related work is [Stradi et al., 2025a], where both *soft* and *hard* constraints are studied in CMDPs with adversarial losses and stochastic constraints. Nonetheless, none of these works focus on developing replicable algorithms.

B Additional Notation

We introduce some additional notation that we will use throughout the following sections. We let $h(t)$ be the epoch associated with round $t \in [T]$. Specifically, when we analyze Algorithm 2 we let $h(t) \in \mathbb{N}$ be the value such that the algorithm plays action $a_t = a_h^\circ$ at round t. Similarly, analyzing Algorithm 3 and Algorithm 4 we let $h(t)$ be such that the algorithm plays according to strategy $\boldsymbol{x}_t = \boldsymbol{x}_h^\circ$ at round t. With this notation, we can formally define the first round of epoch

h as $t_h \coloneqq \min\{t \in [T] \mid h(t) = h\}$ for every epoch $h \in \{0, \dots, H\}$. Let us also recall that for every action $a \in [K]$, round $t \in [T]$ and epoch $h \in \{0, \dots, H\}$ the quantities $N_t(a)$ and $\widetilde{N}_t(a)$ are defined as:

$$N_t(a) \coloneqq \sum_{j=1}^{t-1} \mathbb{I}\{a_j = a\},$$

$$\widetilde{N}_t(a) \coloneqq N_{t_h}(a) = \sum_{j=1}^{t_h-1} \mathbb{I}\{a_j = a\},$$

where $N_t(a)$ is the number of times action a has been selected up to round t *excluded*, Notice that all the algorithms presented in the paper correctly keep track of the quantities defined above.

For the ease of exposition, we also define the event under which the estimators computed by Algorithm 3 and Algorithm 4 belong to the appropriate confidence interval. Formally, we define the *clean event* $\mathcal{E}$ as follows:

Definition B.1 (Clean Event)**.** We let $\mathcal{E}$ be the event under which, for every epoch $h \geq 1$ and arm $a \in [K]$, it holds that:

$$\widehat{\boldsymbol{r}}_h(a) - \boldsymbol{\zeta}_h(a) \leq \boldsymbol{r}(a) \leq \widehat{\boldsymbol{r}}_h(a) + \boldsymbol{\zeta}_h(a) \tag{5}$$

and

$$\widehat{\boldsymbol{g}}_{h,i}(a) - \boldsymbol{\zeta}_h(a) \leq \boldsymbol{g}(a) \leq \widehat{\boldsymbol{g}}_{h,i}(a) + \boldsymbol{\zeta}_h(a) \tag{6}$$

for every constraint $i \in [m]$.

For the analysis of Algorithm 4, we also let $\overline{\mathcal{X}}_h$ be the pessimistic set of safe strategies defined as:

$$\overline{\mathcal{X}}_h \coloneqq \{\boldsymbol{x} \in \Delta_K \mid (\widehat{\boldsymbol{g}}_{h,i} + \boldsymbol{\zeta}_h)^\top \boldsymbol{x} \leq \alpha_i \quad \forall i \in [m]\}, \tag{7}$$

which contains only safe strategies under the event $\mathcal{E}$.

C Proofs Omitted From Section 3

Lemma 3.1. `DEBORA` *is a ρ-replicable algorithm and executes $H \leq K\lceil \log_2(T) \rceil$ epochs.*

Proof. When $T = 1$, there is just a single epoch $h = 0 \leq K\lceil \log_2(T) \rceil$ and Algorithm 2 deterministically chooses the strategy to play during the single round. We will therefore consider the non-trivial case where $T \geq 2$. Furthermore, we will denote with H be the last epoch.

For every $a \in [K]$, let $\mathcal{H}_a$ be the set of epochs $h \in [2, H]$ such that $a^\circ_{h-1} = a$. Denoting with $h_a(i)$ the i-th element of $\mathcal{H}_a$ in ascending order, we show that $\widetilde{N}_{h_a(i)}(a) = 2^{i-1}$. When $i = 1$, we have $\widetilde{N}_{h_a(1)}(a) = 1$ as an epoch terminates when an action is played for the first time. Instead, when $i > 1$, we observe that:

$$\widetilde{N}_{h_a(i)}(a) = 2\widetilde{N}_{h_a(i)-1}(a) = \widetilde{N}_{h_a(i-1)}(a).$$

As a result, it holds that $\widetilde{N}_{h_a(i)}(a) = 2^{i-1}$. Since $T \geq \widetilde{N}_{h_a(i)}(a)$, we also get $i \leq \log_2(T) + 1$, which in turn implies that $\mathcal{H}_a \leq \lfloor \log_2(T) + 1 \rfloor \leq \lceil \log_2(T) \rceil$. Finally, we can upper bound the number of epochs of Algorithm 2 as:

$$H \leq \sum_{a \in [K]} |\mathcal{H}_a| \leq \sum_{a \in [K]} \lceil \log_2(T) \rceil \leq K \lceil \log_2(T) \rceil.$$

Now we show that Algorithm 2 is ρ-replicable. Consider two different trials of Algorithm 2. Let H_i be the last epoch of trial i, and $a^i_0, a^i_1, \dots a^i_{H_i}$ and $\widehat{\boldsymbol{r}}^i_0, \widehat{\boldsymbol{r}}^i_1, \dots, \widehat{\boldsymbol{r}}^i_{H_i}$ be the sequence of actions and estimators computed at each epoch of trial $i \in \{1, 2\}$, respectively. By construction, Algorithm 2 plays the same sequence of actions in both trials if $a^1_h = a^2_h$ for every $h \in [H_1]$ – notice that in this case $H_1 = H_2$.

Initially (epoch $h = 0$), Algorithm 2 plays always the same action $a^\circ_0 = a^1_0 = a^2_0$ computed without any sample from the environment. Hence the algorithm plays the same actions during the first epoch

in both trials, which also always lasts for a single round only. Furthermore, the estimators $\widehat{\boldsymbol{r}}_0^1$ and $\widehat{\boldsymbol{r}}_0^2$ coincide.

Now suppose, by induction, that with probability at least $1-(h-1)\rho'$ a phase $h \geq 1$ begins at rounds $t \in [T]$ in both trials, and that $a_{h'}^1 = a_{h'}^2 =: a_{h'}$ and $\widehat{\boldsymbol{r}}_{h'}^i = \widehat{\boldsymbol{r}}_h^i =: \boldsymbol{r}_{h'}$ for every previous epoch $0 \leq h' < h$. Observe that this inductive hypothesis holds for epoch $h = 1$. For every action $a \neq a_{h-1}$ Algorithm 2 computes the same estimator $\widehat{\boldsymbol{r}}_h^1(a) = \widehat{\boldsymbol{r}}_h^2(a) = \widehat{\boldsymbol{r}}_{h-1}(a)$. For action $a = a_{h-1}$, it employs a ρ'-replicable estimator (Lemma 2.5). Therefore, the estimators $\widehat{\boldsymbol{r}}^1$ and $\widehat{\boldsymbol{r}}^2$ coincide with probability at least $1 - h\rho'$. When this happens, the Algorithm selects the same action $a_h^1 = a_h^2$ to be played at epoch h. As a result, the epoch lasts the same number of rounds in both trials with probability at least $1 - h\rho'$ and the inductive hypothesis holds at epoch $h+1$.

Since there are $K\lceil \log_2(T) \rceil$ epochs, the algorithm plays the same sequence of actions in both trials with probability at least $1 - K\lceil \log_2(T) \rceil \rho'$. Taking $\rho' \leq \frac{\rho}{K\lceil \log_2(T) \rceil}$, the algorithm is ρ-replicable. □

Lemma C.1. *During the execution of Algorithm 2, each suboptimal action $a \in [K]$ such that $\Delta(a) > 0$ is played at most:*

$$N_{T+1}(a) \leq \frac{15 \ln\left(\frac{1}{\delta'}\right)}{\Delta(a)^2(\rho' - 2\delta')^2}$$

times with probability at least $1 - \delta$.

Proof. We first prove the upper bound on the number of pulls under the event that $|\boldsymbol{r}(a) - \widehat{\boldsymbol{r}}_h(a)| \leq \boldsymbol{\zeta}_h(a)$ for every $a \in [K]$. We will later show that this event has probability at least $1 - \delta$.

Consider a suboptimal action $a \in [K] : \Delta(a) > 0$ pulled at least once during the execution of Algorithm 2 (the upper bound trivially holds otherwise). Let h be the last epoch where action a is selected, *i.e.* $a_h^\circ = a$ and $a_{h'}^\circ \neq a$ for every subsequent epoch $h' > h$. For such an action to be selected at an epoch h, it must be the case that:

$$\widehat{\boldsymbol{r}}_h(a) + \boldsymbol{\zeta}_h(a) \geq \widehat{\boldsymbol{r}}_h(a^\star) + \boldsymbol{\zeta}_h(a^\star).$$

We further observe that, under the event that each estimator lies within the appropriate confidence interval, we have:

$$\boldsymbol{r}(a) + 2\boldsymbol{\zeta}_h(a) \geq \widehat{\boldsymbol{r}}_h(a) + \boldsymbol{\zeta}_h(a) \geq \widehat{\boldsymbol{r}}_h(a^\star) + \boldsymbol{\zeta}_h(a^\star) \geq \boldsymbol{r}(a^\star).$$

Therefore:

$$\boldsymbol{\zeta}_h(a) \geq \frac{\Delta(a)}{2}.$$

By the definition of $\boldsymbol{\zeta}_h(a)$ (Equation (1)) and simple computations we get:

$$\widetilde{N}_h(a) \leq \max\{\widetilde{N}_h(a), 1\} \leq \frac{8 \ln\left(\frac{2}{\delta'}\right)}{\Delta(a)^2(\rho' - 2\delta')^2}.$$

We now observe that since h is the last epoch where action a is selected, we have $N_{T+1}(a) \leq \max\{2\widetilde{N}_h(a), 1\}$. Indeed, the epoch terminates when action a has been played $\max\{2\widetilde{N}_h(a), 1\}$ times, and such an action is never played afterward. As a result:

$$N_{T+1}(a) \leq \max\{2\widetilde{N}_h(a), 1\} \leq \frac{16 \ln\left(\frac{2}{\delta'}\right)}{\Delta(a)^2(\rho' - 2\delta')^2},$$

where we observe that:

$$\frac{16 \ln\left(\frac{2}{\delta'}\right)}{\Delta(a)^2(\rho' - 2\delta')^2} \geq \frac{16 \ln\left(\frac{2}{\delta'}\right)}{(1 - 2\delta')^2} \geq 1$$

for every $\delta' \in (0, 1)$.

In order to conclude the proof, we show that that $|\boldsymbol{r}(a) - \widehat{\boldsymbol{r}}_h(a)| \leq \boldsymbol{\zeta}_h(a)$ for every $a \in [K]$ with probability at least $1 - \delta$. Suppose, by induction, that at some epoch $1 \leq h < H$ it holds $|\boldsymbol{r}(a) - \widehat{\boldsymbol{r}}_h(a)| \leq \boldsymbol{\zeta}_h(a)$ for every $a \in [K]$ with probability at least $1 - (h-1)\delta'$ — H denotes the last epoch. Algorithm 2 plays always the same action a_h° for the entire epoch h. Subsequently, it computes a new estimator $\widehat{\boldsymbol{r}}_{h+1}(a_h^\circ)$ which according to Lemma 2.5 lies at distance at most $\boldsymbol{\zeta}_{h+1}(a_h^\circ)$ from $\boldsymbol{r}(a)$ with probability at least $1 - \delta'$. For every other action $a \neq a_h^\circ$, it computes the estimator

$\widehat{\boldsymbol{r}}_{h+1}(a) = \widehat{\boldsymbol{r}}_h(a)$, which lies at distance $\boldsymbol{\zeta}_h(a)$ from $\boldsymbol{r}(a)$ with probability at least $1 - (h-1)\delta'$ by the inductive hypothesis. We observe that $\widetilde{N}_{h+1}(a) = \widetilde{N}_h(a)$, as the action has not been pulled during the epoch h. Therefore, $\boldsymbol{\zeta}_h(a) = \boldsymbol{\zeta}_{h+1}(a)$ by Equation (1), and the estimator lies within the correct interval with probability at least $1 - (h-1)\delta'$. By an union bound, every estimator lies within the correct interval with probability at least $1 - h\delta'$, and the inductive hypothesis holds at epoch $h+1$. Furthermore, the inductive hypothesis holds at the epoch $h = 1$, as the algorithm plays deterministically at epoch $h = 0$ and $\boldsymbol{\zeta}_1(a) \geq 1$ for every action $a \in [K]$. As a result, since there are at most $K\lceil \log_2(T) \rceil$ epochs (Lemma 3.1), with probability at least $1 - K\lceil \log_2(T) \rceil \delta'$ every estimator computed by Algorithm 2 lies within the appropriate confidence interval. The proof is concluded by observing that $\delta' \leq \delta / (K \lceil \log_2(T) \rceil)$. □

Theorem 3.2. *Let $\delta \in (0,1)$ and $\rho \in (2\delta, 1)$. Consider an instance of the stochastic K-armed MAB problem defined by the reward gaps $\Delta(a) \coloneqq \arg\max_{\bar{a} \in [K]} \boldsymbol{r}(\bar{a}) - \boldsymbol{r}(a)$ for all $a \in [K]$. Then, with probability at least $1 - \delta$,* DEBORA *suffers an expected cumulative regret bounded as:*

$$R_T \leq \widetilde{\mathcal{O}} \left(\frac{1}{(\rho - \delta)^2} \log\left(\frac{1}{\delta}\right) K^2 \log^2(T) \sum_{a \neq a^\star} \frac{1}{\Delta(a)} \right).$$

Moreover, for every instance, with probability at least $1 - \delta$, the regret can be bounded as:

$$R_T \leq \widetilde{\mathcal{O}} \left(\frac{1}{(\rho - \delta)} K \sqrt{KT \log\left(\frac{1}{\delta}\right)} \right).$$

Proof. Thanks to Lemma C.1, the number of times suboptimal action $a \in [K]$ is played is at most:

$$N_{T+1}(a) \leq \frac{15 \ln\left(\frac{1}{\delta'}\right)}{\Delta(a)^2 (\rho' - 2\delta')^2}$$

with probability at least $1 - \delta$. Let us continue the analysis conditioned on such an event.

Let $\mathcal{S} \subseteq [K]$ be a set of actions such that $\Delta(a) > 0$ for every $a \in \mathcal{S}$. We have that:

$$\begin{aligned} \sum_{a \in \mathcal{S}} \Delta(a) N_{T+1}(a) &\leq \mathcal{O}\left(\frac{1}{(\rho' - 2\delta')^2} \log\left(\frac{1}{\delta'}\right) \sum_{a \in \mathcal{S}} \frac{1}{\Delta(a)} \right) \\ &= \mathcal{O}\left(\frac{K^2 \log^2(T)}{(\rho - \delta)^2} \log\left(\frac{K \log(T)}{\delta}\right) \sum_{a \in \mathcal{S}} \frac{1}{\Delta(a)} \right). \end{aligned} \tag{8}$$

In particular, by taking $\mathcal{S} = \{a \in [K] \mid \Delta(a) > 0\}$, we have:

$$R_T = \sum_{a : \Delta(a) > 0} \Delta(a) N_{T+1}(a) \leq \mathcal{O}\left(\frac{K^2 \log^2(T)}{(\rho - \delta)^2} \log\left(\frac{K \log(T)}{\delta}\right) \sum_{a : \Delta(a) > 0} \frac{1}{\Delta(a)} \right),$$

proving the instance-dependent regret bound. Furthermore, for every $\epsilon \in (0,1)$ we have:

$$\begin{aligned} R_T &= \sum_{a : \Delta(a) \leq \epsilon} \Delta(a) N_{T+1}(a) + \sum_{a : \Delta(a) > \epsilon} \Delta(a) N_{T+1}(a) \\ &\leq \mathcal{O}\left(\epsilon T + \frac{K^2 \log^2(T)}{(\rho - \delta)^2} \log\left(\frac{K \log(T)}{\delta}\right) \sum_{a : \Delta(a) > \epsilon} \frac{1}{\Delta(a)} \right) \\ &\leq \mathcal{O}\left(\epsilon T + \frac{K^3 \log^2(T)}{(\rho - \delta)^2} \log\left(\frac{K \log(T)}{\delta}\right) \frac{1}{\epsilon} \right), \end{aligned}$$

where we employed Equation (8) with $\mathcal{S} = \{a \in [K] \mid \Delta(a) > \epsilon\}$. Taking:

$$\epsilon \coloneqq \frac{K^{3/2} \log_2(T)}{(\rho - \delta)\sqrt{T}} \sqrt{\ln\left(\frac{K \log(T)}{\delta}\right)},$$

we obtain the instance-independent bound:

$$R_T \leq \mathcal{O}\left(\frac{1}{(\rho-\delta)} K \sqrt{KT \log\left(\frac{K\log(T)}{\delta}\right)}\right),$$

concluding the proof. □

D Proofs Omitted From Section 4

In this Section we provide the proofs of the propositions and theorems of Section 4. Specifically, we first prove Lemma 4.2 and Lemma 4.1. Then, for the ease of exposition, we provide two additional lemmas, showing that conditioned on the clean event (Definition B.1) the regret and the cumulative violation of Algorithm 3 are upper bounded by the same quantity. After another technical lemma (Lemma D.3, which will be employs also in Appendix E), we prove Theorem 4.3.

Lemma 4.2. `DEBORA-S` *is a ρ-replicable algorithm and executes $H \leq K\lceil \log_2(T) \rceil$ epochs.*

Proof. Let H be the number of epochs during the execution of Algorithm 3. First, we show that $H \leq K\lceil \log(T) \rceil$. Let us define the set of rounds $\mathcal{T}_a$ as:

$$\mathcal{T}_a \coloneqq \{t \in [T] \mid N_t(a) = \widetilde{N}_{h(t)}(a) = \max\{2\widetilde{N}_{h(t)-1}(a), 1\}\}.$$

By construction, $h(t) = h(t-1)+1$ if and only if $t \in \mathcal{T}_a$ for some $a \in [K]$ (see Line 6). Let $t_a(i)$ denote the i-th element of $\mathcal{T}_a$ in ascending order. We prove by induction that $\widetilde{N}_{h(t_a(i))}(a) \geq 2^{i-1}$. First, we observe that $\widetilde{N}_{h(t_a(1))}(a) = 1$, as an epoch terminates after an action is played for the first time. Reasoning by induction, for $i > 1$ we assume that $\widetilde{N}_{h(t_a(i-1))}(a) \geq 2^{i-2}$. Therefore:

$$\widetilde{N}_{t_a(i)}(a) = 2\widetilde{N}_{h(t_a(i))-1}(a) = 2\widetilde{N}_{h(t_a(i)-1)}(a) \geq 2^{i-1},$$

where the first equality is due to the definition of $\mathcal{T}_a$, the second as $\widetilde{N}_h(a)$ is monotone non-decreasing in h, and the inequality holds by induction. As a result, $\widetilde{N}_{h(t_a(i))}(a) \geq 2^{i-1}$. Since $\widetilde{N}_{h(t_a(i))}(a) \leq T$, we get $i \leq \log_2(T)+1$. Consequently, action a produces at most $|\mathcal{T}_a| \leq \lfloor \log_2(T)+1 \rfloor \leq \lceil \log_2(T) \rceil$ changes of epoch. Finally, we observe that:

$$H = \sum_{a \in [K]} |\mathcal{T}_a| \leq K\lceil \log_2(T) \rceil,$$

proving the upper bound on the number fo epochs.

Now we show that Algorithm 3 is ρ-replicable. Consider two different trials of Algorithm 3 and and let $\boldsymbol{x}_1^1, \ldots, \boldsymbol{x}_{H^1}^1$ and $\boldsymbol{x}_1^2, \ldots, \boldsymbol{x}_T^2 H^2$ be the strategies x_h° selected by the algorithm in the first and second trial, respectively. Observe that if $\boldsymbol{x}_h^i = \boldsymbol{x}_h^1$ for every epoch up until h, then the Algorithm plays the same sequence of actions at epoch h in both trials.

The algorithm plays always the same strategy during the first epoch, which lasts a single round. Now we reason by induction over the epochs $1 \leq h \leq H^1$. Suppose that $\boldsymbol{x}_{h'}^1 = \boldsymbol{x}_{h'}^2$ for every $0 \leq h < h'$. Then $\boldsymbol{x}_h^1 = \boldsymbol{x}_h^2$ if the estimators computed by Algorithm 3 at the beginning of epoch h are the same in the two trials. According to Lemma 2.5, each single estimator is ρ' replicable. Since there are $(m+1)K$ estimators, with probability at least $1-(m+1)K\rho'$ we have $\boldsymbol{x}_h^1 = \boldsymbol{x}_h^2$. By a union bound over the $H \leq K\lceil \log_2(T) \rceil$ epochs, we have that the algorithm selects the same sequence of strategies in both trial – which therefore last for the same number of rounds – with probability at least $1-(m+1)K^2\lceil \log_2(T) \rceil \rho'$. Hence Algorithm 3 is $(m+1)K^2\lceil \log_2(T) \rceil \rho'$-replicable. Taking $\rho' \leq \frac{\rho}{(m+1)K^2\lceil \log_2(T) \rceil}$, the algorithm is ρ-replicable. □

Lemma 4.1. *Let $\mathcal{X}^\star \coloneqq \{\boldsymbol{x} \in \Delta_K \mid \boldsymbol{g}_i^\top \boldsymbol{x} \leq \alpha_i \ \forall i \in [m]\}$ be the set of safe strategies. With probability at least $1-\delta$, for every epoch h it holds that $\mathcal{X}^\star \subseteq \underline{\mathcal{X}}_h$.*

Proof. By Lemma 4.2, the number of epochs of Algorithm 3 is at most $K\log_2(T)$. Employing Lemma 2.5 and an union bound over the epochs and the m estimators, we have that $\widehat{\boldsymbol{g}}_{h,i}(a) - \boldsymbol{\zeta}_h(a) \leq$

$\boldsymbol{g}_i(a) \leq \widehat{\boldsymbol{g}}_{h,i}(a) + \boldsymbol{\zeta}_h(a)$ with probability at least $1 - \rho' mK \log_2(T) \geq 1 - \delta$. Conditioned on such an event, consider any safe strategy $\boldsymbol{x} \in \mathcal{X}^\star$. We observe that for every constraint $i \in [m]$:

$$(\widehat{\boldsymbol{g}}_{h,i} - \boldsymbol{\zeta}_h)^\top \boldsymbol{x} \leq \boldsymbol{g}_i^\top \boldsymbol{x} \leq \alpha_i,$$

where the first inequality holds under the event defined above, and the second due to the definition of $\mathcal{X}^\star$. As a result, we can conclude that $\boldsymbol{x} \in \underline{\mathcal{X}}_h$. □

Lemma D.1. *Conditioned on the event $\mathcal{E}$ (Definition B.1), the regret of Algorithm 3 satisfies:*

$$R_T \leq 2\sqrt{2} \sum_{t\in[K]} \sum_{a\in[K]} \sqrt{\frac{2\log(2/\delta')}{\max\{1, N_t(a)\}(\rho' - 2\delta')^2}} \boldsymbol{x}_t(a).$$

Proof. As a first step, we prove that, under the event $\mathcal{E}$, it holds that: $(\widehat{\boldsymbol{r}}_h + \boldsymbol{\zeta}_h)^\top \boldsymbol{x}_h^\circ \geq \boldsymbol{r}^\top \boldsymbol{x}^\star$ for every epoch h, where $\boldsymbol{x}^\star$ is any optimal safe strategy. We observe that $\boldsymbol{x}^\star \in \underline{\mathcal{X}}_h$, as:

$$(\widehat{\boldsymbol{g}}_{h,i} - \boldsymbol{\zeta}_h)^\top \boldsymbol{x}^\star \leq \boldsymbol{g}_i^\top \boldsymbol{x}^\star \leq \alpha_i$$

due to Equation 6 and the fact that $\boldsymbol{x}^\star$ is safe. By construction (see Algorithm 3), we have:

$$\boldsymbol{x}_h^\circ \in \operatorname*{argmax}_{\boldsymbol{x}\in\underline{\mathcal{X}}_h} (\widehat{\boldsymbol{r}}_h + \boldsymbol{\zeta}_h)^\top \boldsymbol{x}.$$

Consequently:

$$(\widehat{\boldsymbol{r}}_h + \boldsymbol{\zeta}_h)^\top \boldsymbol{x}_h^\circ \geq (\widehat{\boldsymbol{r}}_h + \boldsymbol{\zeta}_h)^\top \boldsymbol{x}^\star \geq \boldsymbol{r}^\top \boldsymbol{x}^\star, \tag{9}$$

where the first inequality holds due to the definition of $\boldsymbol{x}_h^\circ$ and the fact that $\boldsymbol{x}^\star \in \underline{\mathcal{X}}_h$, while the second one is true under the event $\mathcal{E}$ (Definition B.1).

We now derive an upper bound of the regret as follows:

$$\begin{aligned}
R_T &= \sum_{t\in[T]} (\boldsymbol{r}^\top \boldsymbol{x}^\star - \boldsymbol{r}^\top \boldsymbol{x}_t) \\
&= \sum_{t\in[T]} (\boldsymbol{r}^\top \boldsymbol{x}^\star - \boldsymbol{r}^\top \boldsymbol{x}_{h(t)}) && \text{(10a)} \\
&\leq \sum_{t\in[T]} ((\widehat{\boldsymbol{r}}_{h(t)} + \boldsymbol{\zeta}_{h(t)})^\top \boldsymbol{x}_{h(t)} - \boldsymbol{r}^\top \boldsymbol{x}_{h(t)}) && \text{(10b)} \\
&\leq \sum_{t\in[T]} ((\widehat{\boldsymbol{r}}_{h(t)} + \boldsymbol{\zeta}_{h(t)})^\top \boldsymbol{x}_{h(t)} - (\widehat{\boldsymbol{r}}_{h(t)} - \boldsymbol{\zeta}_{h(t)})^\top \boldsymbol{x}_{h(t)}) && \text{(10c)} \\
&= 2 \sum_{t\in[T]} \boldsymbol{\zeta}_{h(t)}^\top \boldsymbol{x}_{h(t)} \\
&= 2 \sum_{t\in[T]} \sum_{a\in[K]} \sqrt{\frac{2\log(2/\delta')}{\max\{\widetilde{N}_{h(t)}(a), 1\}(\rho' - 2\delta')^2}} \boldsymbol{x}_t(a) && \text{(10d)} \\
&\leq 2\sqrt{2} \sum_{t\in[T]} \sum_{a\in[K]} \sqrt{\frac{2\log(2/\delta')}{\max\{N_t(a), 1\}(\rho' - 2\delta')^2}} \boldsymbol{x}_t(a), && \text{(10e)}
\end{aligned}$$

where Equation (10a) holds by construction ($\boldsymbol{x}_t = \boldsymbol{x}_{h(t)}$), Equation (10b) by Equation (9), Equation (10c) is satisfied under $\mathcal{E}$, Equation (10d) by the definition of $\boldsymbol{\zeta}_{h(t)}$ and the fact that $\boldsymbol{x}_{h(t)} = \boldsymbol{x}_t$, and Equation (10e) as $N_t(a) \leq 2\widetilde{N}_{h(t)}(a)$ for every round $t \in [T]$ and action $a \in [K]$. □

Lemma D.2. *Under the event $\mathcal{E}$ (Definition B.1), the violation of Algorithm 3 is at most:*

$$V_T \leq 2\sqrt{2} \sum_{t\in[T]} \sum_{a\in[K]} \sqrt{\frac{2\log(2/\delta')}{\max\{1, N_t(a)\}(\rho' - 2\delta')^2}} \boldsymbol{x}_t(a).$$

Proof. Consider a single constraint $i \in [m]$ and let us define its positive violation:

$$V_{T,i} \coloneqq \sum_{t\in[T]} \max\{0, \boldsymbol{g}_i^\top \boldsymbol{x}_t - \alpha_i\},$$

so that:

$$V_T = \max_{i\in[m]} V_{T,i}$$

We proceed to bound $V_{T,i}$ for any given $i \in [m]$.

Under the event $\mathcal{E}$, the estimators $\widehat{\boldsymbol{g}}_{h,i}$ computed by Algorithm 3 satisfy:

$$\widehat{\boldsymbol{g}}_{h,i}(a) - \boldsymbol{\zeta}_h(a) \leq \boldsymbol{g}_i(a) \leq \widehat{\boldsymbol{g}}_{h,i}(a) + \boldsymbol{\zeta}_h(a),$$

for any $a \in [K]$ and epoch $h \geq 0$, where $\boldsymbol{\zeta}_h(a)$ is defined in Equation (1). Therefore, for any strategy $\boldsymbol{x} \in \Delta(K)$ the following holds:

$$(\widehat{\boldsymbol{g}}_{h,i} - \boldsymbol{\zeta}_h)^\top \boldsymbol{x} \leq \boldsymbol{g}_i^\top \boldsymbol{x} \leq (\widehat{\boldsymbol{g}}_{h,i} + \boldsymbol{\zeta}_h)^\top \boldsymbol{x}.$$

This allows us to bound the cumulative violation as follows:

$$\begin{aligned}
V_{T,i} &= \sum_{t\in[T]} \max\{0, \boldsymbol{g}_i^\top \boldsymbol{x}_{h(t)} - \alpha_i\} \\
&\leq \sum_{t\in[T]} \max\{0, (\widehat{\boldsymbol{g}}_{h(t),i} + \boldsymbol{\zeta}_{h(t)})^\top \boldsymbol{x}_{h(t)} - \alpha_i\} \\
&= \sum_{t\in[T]} \max\{0, (\widehat{\boldsymbol{g}}_{h(t),i} + 2\boldsymbol{\zeta}_{h(t)} - \boldsymbol{\zeta}_{h(t)})^\top \boldsymbol{x}_{h(t)} - \alpha_i\} \\
&= \sum_{t\in[T]} \max\{0, (\widehat{\boldsymbol{g}}_{h(t),i} - \boldsymbol{\zeta}_{h(t)})^\top \boldsymbol{x}_{h(t)} - \alpha_i + 2\boldsymbol{\zeta}_{h(t)}^\top \boldsymbol{x}_{h(t)}\} \\
&\leq \sum_{t\in[T]} \max\{0, 2\boldsymbol{\zeta}_{h(t)}^\top \boldsymbol{x}_{h(t)}\} \\
&\leq 2 \sum_{t\in[T]} \boldsymbol{\zeta}_{h(t)}^\top \boldsymbol{x}_{h(t)}, \qquad (11)
\end{aligned}$$

where the second inequality holds since $\boldsymbol{x}_{h(t)} \in \underline{\mathcal{X}}_{h(t)}$ (see the proof of Lemma D.1), and therefore $\widehat{\boldsymbol{g}}_{h(t),i} - \boldsymbol{\zeta}_{h(t)})^\top \boldsymbol{x}_{h(t)} \leq \alpha_i$, while the last one holds as every component of $\boldsymbol{\zeta}_{h(t)}$ is non-negative (see Equation (1)). We can now employ the same argument similar to the one provided in the proof of Lemma D.1. In particular, by employing Equation (1), Equation (11) and the fact that $N_t(a) \leq 2\widetilde{N}_{h(t)}(a)$ for every round $t \in [T]$ and action $a \in [K]$, we get:

$$\begin{aligned}
V_T &= \max_{i\in[m]} V_{T,i} \\
&\leq 2 \sum_{t\in[T]} \boldsymbol{\zeta}_{h(t)}^\top \boldsymbol{x}_{h(t)} \\
&= 2 \sum_{t\in[T]} \sum_{a\in[K]} \sqrt{\frac{2\log(2/\delta')}{\max\{\widetilde{N}_{h(t)}(a), 1\}(\rho' - 2\delta')^2}} \boldsymbol{x}_t(a) \\
&\leq 2\sqrt{2} \sum_{t\in[T]} \sum_{a\in[K]} \sqrt{\frac{2\log(2/\delta')}{\max\{N_t(a), 1\}(\rho' - 2\delta')^2}} \boldsymbol{x}_t(a),
\end{aligned}$$

concluding the proof. □

Lemma D.3. *Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ be the strategies played by any learning algorithm. For every $\delta_{\mathrm{A}} \in (0, 1)$ it holds that:*

$$\sum_{t\in[T]} \sum_{a\in[K]} \frac{1}{\sqrt{\max\{1, N_t(a)\}}} \boldsymbol{x}_t(a) \leq 2\sqrt{KT} + \sqrt{2T \log\left(\frac{2}{\delta_{\mathrm{A}}}\right)}$$

with probability at least δ_{A}.

Proof. Let us consider the martingale difference sequence $\{Z_t\}_{t=1}^T$, where:

$$Z_t \coloneqq \sum_{a\in[K]} \frac{\boldsymbol{x}_t(a) - \mathbb{I}\{a_t = a\}}{\sqrt{\max\{1, N_t(a)\}}} \quad \forall t \in [T].$$

We observe that for every $t \in [T]$:

$$\begin{aligned} Z_t &= \sum_{a\in[K]} \frac{\boldsymbol{x}_t(a)}{\sqrt{\max\{1, N_t(a)\}}} - \frac{1}{\sqrt{\max\{1, N_t(a_t)\}}} \\ &\leq \sum_{a\in[K]} \frac{\boldsymbol{x}_t(a)}{\min_{a'\in[K]}\sqrt{\max\{1, N_t(a')\}}} \\ &= \frac{1}{\min_{a'\in[K]}\sqrt{\max\{1, N_t(a')\}}} \leq 1, \end{aligned}$$

and $Z_t \geq 1$. Therefore $|Z_t| \leq 1$, and by applying the Azuma-Hoeffding inequality we have:

$$\sum_{t\in[T]} Z_t \leq \sqrt{2T\log\left(\frac{2}{\delta_{\mathrm{A}}}\right)} \tag{12}$$

with probability at least δ_{A}.

As a further step, we bound the quantity $\sum_{t\in[T]}\sum_{a\in[K]} \frac{\mathbb{I}\{a_t=a\}}{\sqrt{\max\{1,N_t(a)\}}}$ as follows:

$$\begin{aligned} \sum_{t\in[T]}\sum_{a\in[K]} \frac{\mathbb{I}\{a_t = a\}}{\sqrt{\max\{1, N_t(a)\}}} &= \sum_{t\in[T]} \frac{1}{\sqrt{\max\{1, N_t(a_t)\}}} \\ &= \sum_{a\in[K]}\sum_{t:a_t=a} \frac{1}{\sqrt{\max\{1, N_t(a)\}}} \\ &= \sum_{a\in[K]}\sum_{s=0}^{N_T(a)} \frac{1}{\sqrt{\max\{1, s\}}} \\ &= \sum_{a\in[K]}\left(1 + \sum_{s=1}^{N_T(a)} \frac{1}{\sqrt{s}}\right) \\ &\leq \sum_{a\in[K]} (1 + 2\sqrt{N_T(a)} - 1) \\ &\leq 2\sqrt{KT}, \end{aligned}$$

where the first inequality holds applying the bound $\sum_{s=1}^{N} 1/\sqrt{s} \leq 2\sqrt{N} - 1$, and the second by Cauchy-Schwartz and the fact that $\sum_{a\in[K]} N_T(a) = T - 1$. We conclude the proof by combining this result with Equation (12), showing that:

$$\begin{aligned} \sum_{t\in[T]}\sum_{a\in[K]} \frac{1}{\sqrt{\max\{1, N_t(a)\}}}\boldsymbol{x}_t(a) &\leq \sum_{a\in[K]} \frac{\mathbb{I}\{a_t = a\}}{\sqrt{\max\{1, N_t(a)\}}} + \sqrt{2T\log\left(\frac{2}{\delta_{\mathrm{A}}}\right)} \\ &\leq 2\sqrt{KT} + \sqrt{2T\log\left(\frac{2}{\delta_{\mathrm{A}}}\right)}, \end{aligned}$$

with probability at least $1 - \delta_{\mathrm{A}}$. □

Theorem 4.3. *Let $\delta \in (0, 1)$ and $\rho \in (2\delta, 1)$. Then, with probability at least $1 - \delta$,* `DEBORA-S` *suffers an expected cumulative regret bounded as:*

$$R_T \leq \widetilde{\mathcal{O}}\left(\frac{1}{\rho - 2\delta}\log\left(\frac{1}{\delta}\right) mK^2\sqrt{KT}\right),$$

and a cumulative positive constraint violation bounded as:

$$V_T \leq \widetilde{\mathcal{O}}\left(\frac{1}{\rho - 2\delta}\log\left(\frac{1}{\delta}\right) mK^2\sqrt{KT}\right).$$

Proof. According to Lemma D.1 and Lemma D.2, with probability $1 - \mathbb{P}(\bar{\mathcal{E}})$ we have:

$$R_T \leq 2\sqrt{2}\sum_{t\in[K]}\sum_{a\in[K]}\sqrt{\frac{2\log(2/\delta')}{\max\{1, N_t(a)\}(\rho' - 2\delta')^2}}\boldsymbol{x}_t(a).$$

and:

$$V_T \leq 2\sqrt{2}\sum_{t\in[T]}\sum_{a\in[K]}\sqrt{\frac{2\log(2/\delta')}{\max\{1, N_t(a)\}(\rho' - 2\delta')^2}}\boldsymbol{x}_t(a).$$

Thanks to Lemma D.3, we can bound the double summation as:

$$\sum_{t\in[T]}\sum_{a\in[K]}\sqrt{\frac{2\log(2/\delta')}{\max\{1, N_t(a)\}(\rho' - 2\delta')^2}}\boldsymbol{x}_t(a) \leq \sqrt{\frac{2\log\left(\frac{2}{\delta'}\right)}{(\rho' - 2\delta')^2}}\left(2\sqrt{KT} + \sqrt{2T\log\left(\frac{4}{\delta}\right)}\right)$$

with probability at least $1 - \delta/2$.

As a result, with probability at least $1 - \mathbb{P}(\bar{\mathcal{E}}) - \delta/2$:

$$\begin{aligned}
R_T &\leq 2\sqrt{2}\sqrt{\frac{2\log\left(\frac{2}{\delta'}\right)}{(\rho' - 2\delta')^2}}\left(2\sqrt{KT} + \sqrt{2T\log\left(\frac{4}{\delta}\right)}\right) \\
&\leq \mathcal{O}\left(\sqrt{\frac{\log\left(\frac{1}{\delta'}\right)}{(\rho' - 2\delta')^2}KT\log\left(\frac{1}{\delta}\right)}\right) \\
&= \mathcal{O}\left((m+1)K^2\log(T)\sqrt{\frac{\log\left(\frac{mK^2\log(T)}{\delta}\right)}{(\rho - 2\delta)^2}KT\log\left(\frac{1}{\delta}\right)}\right) \\
&= \widetilde{\mathcal{O}}\left(\frac{1}{\rho - 2\delta}\log\left(\frac{1}{\delta}\right) mK^2\sqrt{KT}\right),
\end{aligned}$$

where the first equality holds as $\delta' = \Theta\left(\frac{\delta}{mK^2\log(T)}\right)$ and $\rho' = \Theta\left(\frac{\rho}{mK^2\log(T)}\right)$. A similar argument shows that, conditioned on the same event, the cumulative positive violation is at most:

$$V_T \leq \widetilde{\mathcal{O}}\left(\frac{1}{\rho - 2\delta}\log\left(\frac{1}{\delta}\right) mK^2\sqrt{KT}\right).$$

Therefor,e the inequalities in the statement hold with probability at least $1 - \mathbb{P}(\bar{\mathcal{E}}) - \delta/2$.

In order to conclude the proof, we show that $\mathbb{P}(\bar{\mathcal{E}}) \leq \delta/2$. By Lemma 2.5, there are at most $K\lceil\log(T)\rceil$ epochs (excluding epoch $h = 0$ where no estimator is computed). Furthermore, by construction, Algorithm 3 computes $K(m+1)$ estimators for epoch, each one being within the appropriate interval with probability at least $1 - \delta'$. (Lemma 2.5). By a union bound, we have that the clean event $\mathcal{E}$ holds with probability at least $1 - K^2(m+1)\lceil\log_2(T)\rceil\delta'$. Since $\delta' \leq \frac{\delta}{2K^2(m+1)\lceil\log_2(T)\rceil}$, we have that $\mathbb{P}(\bar{\mathcal{E}}) \leq \delta/2$, concluding the proof. □

E Proofs Omitted From Section 5

Lemma 5.2. `DEBORA-H` *is a ρ-replicable algorithm and executes $H \leq K\lceil\log_2(T)\rceil$ epochs.*

Proof. The statement can be proven by the same argument employed for Lemma 3.1. □

Lemma E.1. *During the execution of Algorithm 4, it holds that* $0 \leq \sigma_h \leq \frac{1}{1+\lambda_{\min}} \leq 1$ *for every epoch* $h \in [0, H]$.

Proof. Fix an epoch h. If $\mathcal{V}_h = \emptyset$, then $\sigma_h = 0$ and the statement holds. Suppose instead that $\mathcal{V}_h \neq \emptyset$. There exists a constraint $j \in \mathcal{V}_h$ (*i.e.*, $\min\{(\widehat{\boldsymbol{g}}_{h,j} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h, 1\} \geq \alpha_j$) such that:

$$\sigma_h = \frac{y}{y + \lambda_j} \leq \frac{1}{1 + \lambda_j} \leq \frac{1}{1 + \lambda_{\min}} \leq 1,$$

where $y \coloneqq \min\{(\widehat{\boldsymbol{g}}_{h,j} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h, 1\} - \alpha_j \in [0, 1]$. The equality holds by definition (Equation (4)), while the chain of inequalities is verified as $y > 0$ and $\lambda_j \geq \lambda_{\min} > 0$. This concludes the proof. □

Lemma 5.1. *With a probability of at least* $1 - \delta$, DEBORA-H *plays a safe strategy* $\boldsymbol{x}_t$ *for every* $t \in [T]$.

Proof. We first show that the statement holds with probability one conditioned on the event $\mathcal{E}$. We will then show that this event has probability at least $1 - \delta$.

Fix a round $t \in [T]$ and let $h = h(t)$ be its epoch. Algorithm 4 plays $\boldsymbol{x}_t = \boldsymbol{x}_h^\circ$, where $\boldsymbol{x}_h^\circ$ has been computed at the beginning of epoch h. Therefore, we have to show that $\boldsymbol{x}_h^\circ$ is a safe strategy. We first observe that every strategy $\boldsymbol{x} \in \overline{\mathcal{X}}_h$ is safe, conditioned on the event $\mathcal{E}$. Indeed, we have that for every constraint $i \in [m]$:

$$\boldsymbol{g}_i^\top \boldsymbol{x} \leq (\widehat{\boldsymbol{g}}_{h,i} + \boldsymbol{\zeta}_h)^\top \boldsymbol{x} \leq \alpha_i,$$

where the first inequality true under $\mathcal{E}$, and the second holds because $\boldsymbol{x} \in \overline{\mathcal{X}}_h$.

Notably, when $\mathcal{V}_h = \emptyset$, by definition $\widetilde{\boldsymbol{x}}_h \in \overline{\mathcal{X}}_h$ (see Equation (7)) and $\sigma_h = 0$ (Equation 4, where the maximum over an empty set is to be interpreted as zero). Therefore, in this case Algorithm 4 computes the safe strategy $\boldsymbol{x}_h^\circ = \widetilde{\boldsymbol{x}}_h \in \overline{\mathcal{X}}_h$, which contains only safe strategies under $\mathcal{E}$.

Consider now the case $\mathcal{V}_h \neq \emptyset$. There exists a constraint $j \in \mathcal{V}_h$ ($\min\{(\widehat{\boldsymbol{g}}_{h,j} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h, 1\} \geq \alpha_j$ by Equation 3) such that:

$$\sigma_h = \frac{\min\{(\widehat{\boldsymbol{g}}_{h,j} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h, 1\} - \alpha_j}{\min\{(\widehat{\boldsymbol{g}}_{h,j} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h, 1\} - \alpha_j + \lambda_j}.$$

Consider any constraint $i \in [m] \setminus \mathcal{V}_h$. Notice that $\widetilde{\boldsymbol{x}}_h \in \underline{\mathcal{X}}_h$ satisfies the constraint i, as:

$$\boldsymbol{g}_j^\top \widetilde{\boldsymbol{x}}_h \leq (\widehat{\boldsymbol{g}}_{h,j} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h \leq \alpha_i,$$

where the first inequality holds under $\mathcal{E}$ and the second by Equation (2). Since $\boldsymbol{x}^\diamond$ also satisfies the constraint, by linearity the linear combination $\boldsymbol{x}_h$ of $\boldsymbol{x}^\diamond$ and $\widetilde{\boldsymbol{x}}_h$ with coefficient $\sigma_h \in [0, 1]$ (Lemma E.1) satisfies it too. For any other constraint $i \in \mathcal{V}_h$, let us define:

$$\omega_{h,i} \coloneqq \min\{(\boldsymbol{g}_{h,i} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h, 1\}$$

fro the sake of notation. We observe that:

$$\begin{aligned}
\boldsymbol{g}_i^\top \boldsymbol{x}_h &= \sigma_h \boldsymbol{g}_i^\top \boldsymbol{x}^\diamond + (1 - \sigma_h) \boldsymbol{g}_i^\top \widetilde{\boldsymbol{x}}_h \\
&\leq \sigma_h (\alpha_i - \lambda_i) + (1 - \sigma_h) \boldsymbol{g}_i^\top \widetilde{\boldsymbol{x}}_h \\
&\leq \sigma_h (\alpha_i - \lambda_i) + (1 - \sigma_h) \omega_{h,i} \\
&= \omega_{h,i} - \sigma_h (\omega_{h,i} - \alpha_i + \lambda_i) \\
&\leq \omega_{h,i} - \frac{\omega_{h,i} - \alpha_i}{\omega_{h,i} - \alpha_i + \lambda_i} (\omega_{h,i} - \alpha_i + \lambda_i) \\
&= \alpha_i,
\end{aligned}$$

where the first inequality holds due to the Slater's condition, and the second holds under the event $\mathcal{E}$. For the last one, we use Equation (4) to upper bound σ_h after observing that $\omega_{h,i} - \alpha_i + \lambda_i \geq 0$ as $i \in \mathcal{V}_h$. As a result, $\boldsymbol{x}_h^\circ$ is safe under event $\mathcal{E}$.

Finally, we recall that there are at most $K \log_2(T)$ epochs (Lemma 5.2). By Lemma 2.5 and an union bound over the epochs and the $(m + 1)K$ estimators computed every epoch, we have that $\mathcal{P}(\mathcal{E}) \geq 1 - (m + 1)K^2 \log_2(T)\delta'$. Since Algorithm 4 initializes $\delta' \leq \frac{\delta}{(m+1)K^2 \log_2(T)}$, the probability of event $\mathcal{E}$ is at least $1 - \delta$. □

Lemma E.2. *Algorithm 4 computes a sequence of strategies such that:*

$$\sum_{t\in[T]}(1-\sigma_{h(t)})\boldsymbol{\zeta}_{h(t)}^\top\widetilde{\boldsymbol{x}}_{h(t)} \le 2\sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{(\rho'-2\delta')^2}}\left(2\sqrt{KT}+\sqrt{2T\log\left(\frac{2}{\delta'}\right)}\right)$$

with probability at least $1-\delta'$.

Proof. Let us observe that:

$$\begin{aligned}\sum_{t=1}^{T}(1-\sigma_{h(t)})\boldsymbol{\zeta}_{h(t)}^\top\widetilde{\boldsymbol{x}}_{h(t)} &= \sum_{t=1}^{T}(1-\sigma_{h(t)})\sum_{a\in[K]}\sqrt{\frac{2\log\left(\frac{2}{\delta'}\right)}{\min\{\widetilde{N}_{h(t)}(a),1\}(\rho'-2\delta')^2}}\widetilde{\boldsymbol{x}}_{h(t)}(a)\\ &\le \sqrt{2}\sum_{t=1}^{T}(1-\sigma_{h(t)})\sum_{a\in[K]}\sqrt{\frac{2\log\left(\frac{2}{\delta'}\right)}{\min\{N_t(a),1\}(\rho'-2\delta')^2}}\widetilde{\boldsymbol{x}}_{h(t)}(a),\end{aligned}$$

where we applied Equation (1) and the fact that $N_t(a)\le 2\widetilde{N}_{h(t)}(a)$ by construction.

We now observe that $\widetilde{\boldsymbol{x}}_h$ can be rewritten as:

$$\widetilde{\boldsymbol{x}}_h = \frac{1}{1-\sigma_h}\boldsymbol{x}_h - \frac{\sigma_h}{1-\sigma_h}\boldsymbol{x}^\diamond.$$

By employing the identity above and discarding the negative terms, we get:

$$\begin{aligned}\sum_{t=1}^{T}(1-\sigma_{h(t)})\boldsymbol{\zeta}_{h(t)}^\top\widetilde{\boldsymbol{x}}_{h(t)} &\le \sqrt{2}\sum_{t=1}^{T}\sum_{a\in[K]}\frac{(1-\sigma_{h(t)})}{(1-\sigma_{h(t)})}\sqrt{\frac{2\log\left(\frac{2}{\delta'}\right)}{\min\{N_t(a),1\}(\rho'-2\delta')^2}}\boldsymbol{x}_{h(t)}(a)\\ &= \sqrt{2}\sum_{t=1}^{T}\sum_{a\in[K]}\sqrt{\frac{2\log\left(\frac{2}{\delta'}\right)}{\min\{N_t(a),1\}(\rho'-2\delta')^2}}\boldsymbol{x}_{h(t)}(a)\\ &= \sqrt{2}\sum_{t=1}^{T}\sum_{a\in[K]}\sqrt{\frac{2\log\left(\frac{2}{\delta'}\right)}{\min\{N_t(a),1\}(\rho'-2\delta')^2}}\boldsymbol{x}_{t}(a)\\ &= 2\sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{(\rho'-2\delta')^2}}\sum_{t=1}^{T}\sum_{a\in[K]}\sqrt{\frac{1}{\min\{N_t(a),1\}}}\boldsymbol{x}_{t}(a)\end{aligned}$$

The statement follows by employing Lemma D.3. □

Lemma E.3. *Conditioned on the event* $\mathcal{E}$*, Algorithm 4 computes a sequence of strategies and coefficients such that;*

$$\sum_{t=1}^{T}(1-\sigma_{h(t)})\boldsymbol{r}^\top(\boldsymbol{x}^\star-\widetilde{\boldsymbol{x}}_{h(t)}) \le 4\sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{(\rho'-2\delta')^2}}\left(2\sqrt{KT}+\sqrt{2T\log\left(\frac{2}{\delta'}\right)}\right),$$

with probability at least $1-\delta'$.

Proof. The proof is similar to the one provided for Lemma D.1.

As a first step, we prove that, under the event $\mathcal{E}$, we have $(\widehat{\boldsymbol{r}}_h+\boldsymbol{\zeta}_h)^\top\boldsymbol{x}_t \ge \boldsymbol{r}^\top\widetilde{\boldsymbol{x}}^\star$ for every epoch $h\ge 0$. By construction, we have:

$$\widetilde{\boldsymbol{x}}_h \in \max_{\boldsymbol{x}\in\mathcal{X}_h}(\widehat{\boldsymbol{r}}_h+\boldsymbol{\zeta}_h)^\top\boldsymbol{x},$$

where $\boldsymbol{x}^\star\in\mathcal{X}_{h(t)}$ according to Lemma 4.1. Consequently:

$$(\widehat{\boldsymbol{r}}_h+\boldsymbol{\zeta}_h)^\top\widetilde{\boldsymbol{x}}_h \ge (\widehat{\boldsymbol{r}}_h+\boldsymbol{\zeta}_h)^\top\boldsymbol{x}^\star \ge \boldsymbol{r}^\top\boldsymbol{x}^\star, \tag{13}$$

where the first inequality holds due to the optimality of $\widetilde{\boldsymbol{x}}_h$ and the second is verified under event $\mathcal{E}$.

We can now prove the statement as follows:

$$\begin{aligned}\sum_{t=1}^{T}(1-\sigma_{h(t)})\left(\boldsymbol{r}^\top\boldsymbol{x}^\star-\boldsymbol{r}^\top\widetilde{\boldsymbol{x}}_{h(t)}\right) &\leq \sum_{t=1}^{T}(1-\sigma_{h(t)})\left((\widehat{\boldsymbol{r}}_{h(t)}+\boldsymbol{\zeta}_{h(t)})^\top\widetilde{\boldsymbol{x}}_{h(t)}-\widetilde{\boldsymbol{x}}_{h(t)}\right)\\ &= \sum_{t=1}^{T}(1-\sigma_{h(t)})\left((\widehat{\boldsymbol{r}}_{h(t)}+\boldsymbol{\zeta}_{h(t)}-\boldsymbol{r})^\top\widetilde{\boldsymbol{x}}_{h(t)}\right)\\ &\leq \sum_{t=1}^{T}(1-\sigma_{h(t)})\left((\widehat{\boldsymbol{r}}_{h(t)}+\boldsymbol{\zeta}_{h(t)}-\widehat{\boldsymbol{r}}_{h(t)}+\boldsymbol{\zeta}_{h(t)})^\top\widetilde{\boldsymbol{x}}_{h(t)}\right)\\ &= 2\sum_{t=1}^{T}(1-\sigma_{h(t)})\boldsymbol{\zeta}_{h(t)}^\top\widetilde{\boldsymbol{x}}_{h(t)},\end{aligned}$$

where the inequalities hold due to Equation (13) and the the event $\mathcal{E}$. Employing Lemma E.2, the following result holds with probability at least $1-\delta'$:

$$\sum_{t=1}^{T}(1-\sigma_{h(t)}) \leq 4\sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{(\rho'-2\delta')^2}}\left(2\sqrt{KT}+\sqrt{2T\log\left(\frac{2}{\delta'}\right)}\right),$$

concluding the proof. □

Lemma E.4. *Let:*

$$T_1 \coloneqq \frac{8K\ln\left(\frac{1}{\delta'}\right)}{\lambda_{\min}^2(\rho'-2\delta')^2}+\frac{8}{\lambda_{\min}^2}\ln\left(\frac{1}{\delta'}\right) \leq \frac{9K\ln\left(\frac{1}{\delta'}\right)}{\lambda_{\min}^2(\rho'-2\delta')^2}.$$

With probability at least $1-\delta'$, Algorithm 4 computes $\boldsymbol{\zeta}_{h(t)}(a) \leq 1$ for every $T_1 \leq t \leq T$ and $a \in [k]$.

Proof. According to Equation (1), we have $\boldsymbol{\zeta}_{h(t)}(a) \leq 1$ when:

$$\widetilde{N}_{h(t)}(a) \geq \frac{2\ln\left(\frac{1}{\delta'}\right)}{(\rho'-2\delta')^2},$$

which by construction ($\widetilde{N}_{h(t)}(a) \leq N_t(a) \leq 2\widetilde{N}_{h(t)}(a)$) is verified when $N_t(a) \geq B$, where B is a quantity defined as:

$$B \coloneqq \frac{4\ln\left(\frac{1}{\delta'}\right)}{(\rho'-2\delta')^2}.$$

Let us define T_1 as:

$$T_1 \coloneqq \frac{2KB}{\lambda_{\min}^2}+\frac{8}{\lambda_{\min}^2}\ln\left(\frac{1}{\delta'}\right) = \frac{8K\ln\left(\frac{1}{\delta'}\right)}{\lambda_{\min}^2(\rho'-2\delta')^2}+\frac{8}{\lambda_{\min}^2}\ln\left(\frac{1}{\delta'}\right),$$

which we assume to be strictly less than T (otherwise the statement is trivially satisfied). In the following, we show that for every $t \in [T_1, T]$ every action has been played at least B times with probability at least $1-\delta'$, therefore proving the statement.

Let $Y_t \coloneqq \mathbb{I}\{\boldsymbol{\zeta}_{h(t)}(a_t) > 1 \ \vee \ \boldsymbol{\zeta}_{h(t)}(a) \leq 1 \ \forall a \in [K]\}$ and $S_{T_1} \coloneqq \sum_{t\in[T_1]} Y_t$. S_{T_1} counts the number of rounds an action with confidence interval larger than one has been played, plus some extra rounds where every confidence interval was at most one. Since every action can be played at most B times before its confidence interval goes below one, the statement is verified if:

$$S_{T_1} \geq KB$$

with probability at least $1-\delta$. In order to do so, we decompose S_{T_1} as:

$$S_{T_1} = \sum_{t\in[T_1]} Y_t = \sum_{t\in[T_1]} \mathbb{E}[Y_t|\mathcal{F}_t] + \sum_{t\in[T_1]} Z_t,$$

where $\mathcal{F}_t$ represents the filtration up to the decision point of round t and $Z_t \coloneqq Y_t - \mathbb{E}[Y_t|\mathcal{F}_t]$ for every $t \in [T_1]$. We will lower bound the last two summations separately.

To lower bound $\sum_{t\in[T_1]} \mathbb{E}[Y_t|\mathcal{F}_t]$, we observe that whenever at least an action a has confidence interval $\boldsymbol{\zeta}_{h(t)}(a)$ larger than one, $\widetilde{\boldsymbol{x}}_{h(t)}$ is a pure strategy, *i.e.*, there exists an action $\widetilde{a}_{h(t)}$ such that $\widetilde{\boldsymbol{x}}_{h(t)}(\widetilde{a}_{h(t)}) = 1$ and $\boldsymbol{\zeta}_{h(t)}(\widetilde{a}_{h(t)}) > 1$. In this case we have:

$$\mathbb{E}[Y_t|\mathcal{F}_t] \geq \mathbb{P}(a_t = \widetilde{a}_{h(t)}) = \boldsymbol{x}_{h(t)}(\widetilde{a}_{h(t)}) \geq (1-\sigma_{h(t)})\widetilde{\boldsymbol{x}}_{h(t)}(\widetilde{a}_{h(t)}) \geq \lambda_{\min},$$

where the first inequality holds because $\boldsymbol{\zeta}_{h(t)}(\widetilde{a}_{h(t)}) > 1$, the second by the definition of the per-round convex combination employed by Algorithm 4, and the last one by Lemma E.1. Instead, when every action has confidence interval at most one, we have $\mathbb{E}[Y_t|\mathcal{F}_t] = 1$ by definition of Y_t. Therefore:

$$\mathbb{E}[Y_t|\mathcal{F}_t] \geq \lambda_{\min} \tag{14}$$

for every $t \in [T_1]$.

We further observe that Z_t is a martingale difference over the filtration $\mathcal{F}_t$. Hence, we have that:

$$\begin{aligned}\mathbb{P}\left(\sum_{t\in[T_1]} Z_t \geq KB - \lambda_{\min}T_1\right) &= \mathbb{P}\left(\sum_{t\in[T_1]} Z_t \geq -(\lambda_{\min}T_1 - KB)\right) \\ &\geq 1 - \exp\left(-\frac{(\lambda_{\min}T_1 - KB)^2}{2T_1}\right) \geq 1 - \delta',\end{aligned}$$

where the first inequality is due to the Azuma–Hoeffding inequality (observe that $\lambda_{\min}T_1 - KB > 0$), and the last one due to the definitions of T_1 and B. Combining this lower bound with Equation (14), we have that:

$$\sum_{t\in[T_1]} Y_t = \sum_{t\in[T_1]} \mathbb{E}[Y_t|\mathcal{F}_t] + \sum_{t\in[T_1]} Z_t \geq KB$$

with probability at least $1-\delta'$. This implies that every confidence interval is at most one after T_1 rounds with probability at least $1-\delta'$. finally, we observe that

$$T_1 = \frac{8}{\lambda_{\min}^2}\ln\left(\frac{1}{\delta'}\right)\left(\frac{K}{(\rho'-2\delta')^2}+1\right) \leq \frac{9K\ln\left(\frac{1}{\delta'}\right)}{\lambda_{\min}^2(\rho'-2\delta')^2},$$

concluding the proof. □

Lemma E.5. *For every epoch h such that Algorithm 4 computes $\sigma_h \geq 1/2$, it holds that:*

$$\boldsymbol{\zeta}_h^\top \boldsymbol{x}_h^\circ \geq \frac{1}{4}\lambda_{\min}^2.$$

Proof. Since $\sigma_h \neq 0$, the set $\mathcal{V}_h$ is not empty. Let $j \in \mathcal{V}_h$ be the constraint such that:

$$\sigma_h = \frac{\min\{(\widehat{\boldsymbol{g}}_{h,j}+\boldsymbol{\zeta}_h)^\top\widetilde{\boldsymbol{x}}_h, 1\} - \alpha_j}{\min\{(\widehat{\boldsymbol{g}}_{h,j}+\boldsymbol{\zeta}_h)^\top\widetilde{\boldsymbol{x}}_h, 1\} - \alpha_j + \lambda_j} \geq \frac{1}{2}. \tag{15}$$

We show that $\boldsymbol{\zeta}_h^\top\widetilde{\boldsymbol{x}}_h \geq \frac{\lambda_j}{2}$. We consider two cases. First, when $(\widehat{\boldsymbol{g}}_{h,j}+\boldsymbol{\zeta}_h)^\top\widetilde{\boldsymbol{x}}_h \leq 1$, we get:

$$\begin{aligned}\alpha_j + \lambda_j &\leq (\widehat{\boldsymbol{g}}_{h,j}+\boldsymbol{\zeta}_h)^\top\widetilde{\boldsymbol{x}}_h \\ &= (\widehat{\boldsymbol{g}}_{h,j}-\boldsymbol{\zeta}_h)^\top\widetilde{\boldsymbol{x}}_h + 2\boldsymbol{\zeta}_h^\top\widetilde{\boldsymbol{x}}_h \\ &\leq \alpha_j + 2\boldsymbol{\zeta}_h^\top\widetilde{\boldsymbol{x}}_h,\end{aligned}$$

where the first inequality is obtained rearranging Equation (15), and the second inequality holds as, by construction, $\widetilde{\boldsymbol{x}}_h \in \underline{\mathcal{X}}_h$ and it satisfies Equation 2. As a result, $\boldsymbol{\zeta}_h^\top\widetilde{\boldsymbol{x}}_h \geq \frac{\lambda_j}{2}$ in this case. The other case is that $(\widehat{\boldsymbol{g}}_{h,j}+\boldsymbol{\zeta}_h)^\top\widetilde{\boldsymbol{x}}_h \geq 1$. This implies that:

$$\begin{aligned}1 &\leq (\widehat{\boldsymbol{g}}_{h,j}+\boldsymbol{\zeta}_h)^\top\widetilde{\boldsymbol{x}}_h \\ &= (\widehat{\boldsymbol{g}}_{h,j}-\boldsymbol{\zeta}_h)^\top\widetilde{\boldsymbol{x}}_h + 2\boldsymbol{\zeta}^\top\widetilde{\boldsymbol{x}}_h \\ &\leq \alpha_j + 2\boldsymbol{\zeta}^\top\widetilde{\boldsymbol{x}}_h\end{aligned}$$

$$\leq 1 - \lambda_j + 2\boldsymbol{\zeta}^\top \widetilde{\boldsymbol{x}}_h,$$

where the second inequality holds holds because $\widetilde{\boldsymbol{x}}_h \in \underline{\mathcal{X}}_h$, and the second as rearranging Equation (15) we get $\alpha_j \leq 1 - \lambda_j$. As a result, $\boldsymbol{\zeta}_h^\top \widetilde{\boldsymbol{x}}_h \geq \frac{\lambda_j}{2}$ in both cases.

To conclude the proof, we lower bound $\boldsymbol{\zeta}_h^\top \boldsymbol{x}_h^\circ$ as follows:

$$\begin{aligned}
\boldsymbol{\zeta}_h^\top \boldsymbol{x}_h^\circ &= \sigma_h \boldsymbol{\zeta}^\top \boldsymbol{x}^\diamond + (1 - \sigma_h)\boldsymbol{\zeta}^\top \widetilde{\boldsymbol{x}}_h \\
&\geq (1 - \sigma_h)\boldsymbol{\zeta}^\top \widetilde{\boldsymbol{x}}_h \\
&\geq \frac{\lambda_{\min}}{1 + \lambda_{\min}} \boldsymbol{\zeta}^\top \widetilde{\boldsymbol{x}}_h \\
&\geq \frac{\lambda_{\min}^2}{2(1 + \lambda_{\min})} \\
&\geq \frac{1}{4}\lambda_{\min}^2,
\end{aligned}$$

where the second inequality holds as $\sigma_h \geq 1/(1 + \lambda_{\min})$ by Lemma E.1, the third as $\boldsymbol{\zeta}_h^\top \widetilde{\boldsymbol{x}}_h \geq \frac{\lambda_j}{2}$ and $\lambda_j \geq \lambda_{\min}$, and the last one as $\lambda_{\min} \leq 1$. □

Lemma E.6. *Consider any execution of of Algorithm 4 and let:*

$$\mathcal{T}_C \coloneqq \left\{ t \in [T] \mid \frac{\lambda_{\min}^2}{C} \leq \boldsymbol{\zeta}_{h(t)}(a_t) \leq 1 \right\}$$

for some $C > 1$. The cardinality of $\mathcal{T}_C$ is bounded by:

$$|\mathcal{T}_C| \leq \frac{16C^2}{\lambda_{\min}^4} \frac{\log\left(\frac{2}{\delta'}\right)}{(\rho' - 2\delta')^2}.$$

Proof. We first observe that due to the definition of $\mathcal{T}_C$:

$$\sum_{t \in \mathcal{T}_C} \boldsymbol{\zeta}_{h(t)}(a_t) \geq |\mathcal{T}_C| \frac{\lambda_{\min}^2}{C}.$$

Therefore:

$$\begin{aligned}
|\mathcal{T}_C| &\leq \frac{C}{\lambda_{\min}^2} \sum_{t \in \mathcal{T}_C} \boldsymbol{\zeta}_{h(t)}(a_t) \\
&= \frac{C}{\lambda_{\min}^2} \sum_{t \in \mathcal{T}_C} \sqrt{\frac{2\log\left(\frac{2}{\delta'}\right)}{\min\{\widetilde{N}_{h(t)}(a_t), 1\}(\rho' - 2\delta')^2}} \\
&\leq \frac{C}{\lambda_{\min}^2} \sqrt{2} \sum_{t \in \mathcal{T}_C} \sqrt{\frac{2\log\left(\frac{2}{\delta'}\right)}{\min\{N_t(a_t), 1\}(\rho' - 2\delta')^2}} \\
&= \frac{C}{\lambda_{\min}^2} 2 \sum_{t \in \mathcal{T}_C} \sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{\min\{N_t(a_t), 1\}(\rho' - 2\delta')^2}} \\
&\leq 4\frac{C}{\lambda_{\min}^2} \sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{(\rho' - 2\delta')^2} K|\mathcal{T}_C|},
\end{aligned}$$

where the second inequality holds since $\widetilde{N}_t(a_t) \leq 2N_t(a)$, and the last one due to the fact that $\sum_{t \in [\mathcal{T}_C]} \frac{1}{\min\{N_t(a_t), 1\}} \leq 2\sqrt{K|\mathcal{T}_C|}$ (the argument is analogous to the one provided in the proof of Lemma D.3). Solving the inequality with respect to $|\mathcal{T}_C|$ yields:

$$|\mathcal{T}_C| \leq \frac{16C^2}{\lambda_{\min}^4} \frac{\log\left(\frac{2}{\delta'}\right)}{(\rho' - 2\delta')^2},$$

concluding the proof. □

Lemma E.7. *The sequence of coefficients* $\{\sigma_{h(0)}, \dots, \sigma_{h(T)}\}$ *computed by Algorithm 4 satisfies:*

$$\sum_{t\in[T]} \sigma_{h(t)} \le \frac{8}{\lambda_{\min}} \sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{(\rho' - 2\delta')^2}} \left(2\sqrt{KT} + \sqrt{2T \log\left(\frac{2}{\delta'}\right)}\right) + \frac{1610K \log\left(\frac{2}{\delta'}\right)}{\lambda_{\min}^6 (\rho' - 2\delta')^2}$$

with probability at least $1 - 2\delta'$.

Proof. First, we argue that σ_h satisfies:

$$\sigma_h \le \frac{2}{\lambda_{\min}} \boldsymbol{\zeta}_h^\top \widetilde{\boldsymbol{x}}_h \tag{16}$$

for every epoch h. Equation (16) is trivially satisfied when $\mathcal{V}_h = \emptyset$. Instead, when $\mathcal{V}_h \neq \emptyset$, there exists some $j \in \mathcal{V}_h$ such that:

$$\begin{aligned}
\sigma_h &= \frac{\min\{(\widehat{\boldsymbol{g}}_{h,j} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h, 1\} - \alpha_j}{\min\{(\widehat{\boldsymbol{g}}_{h,j} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h, 1\} - \alpha_j + \lambda_j} \\
&\le \frac{(\widehat{\boldsymbol{g}}_{h,j} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h - \alpha_j}{(\widehat{\boldsymbol{g}}_{h,j} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h - \alpha_j + \lambda_j} \\
&\le \frac{(\widehat{\boldsymbol{g}}_{h,j} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h - \alpha_j}{\lambda_j} && \text{(17a)} \\
&= \frac{(\widehat{\boldsymbol{g}}_{h,j} - \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h + 2\boldsymbol{\zeta}_h^\top \widetilde{\boldsymbol{x}}_h - \alpha_j}{\lambda_j} \\
&\le \frac{\alpha_j + 2\boldsymbol{\zeta}_h^\top \widetilde{\boldsymbol{x}}_h - \alpha_j}{\lambda_j} && \text{(17b)} \\
&\le \frac{2}{\lambda_{\min}} \boldsymbol{\zeta}_h^\top \widetilde{\boldsymbol{x}}_h
\end{aligned}$$

where Equation (17a) holds since $(\widehat{\boldsymbol{g}}_{h,j} + \boldsymbol{\zeta}_h)^\top \widetilde{\boldsymbol{x}}_h - \alpha_j > 0$ for $j \in \mathcal{V}_h$, and Equation (17b) as $\widetilde{\boldsymbol{x}}_h \in \underline{\mathcal{X}}_h$ defined by Equation (2) by construction. Consequently, Equation (16) is verified.

We now partition the time horizon into three disjoint sets $\mathcal{T}_\mathrm{I}$, $\mathcal{T}_\mathrm{S}$ and $\mathcal{T}_\mathrm{L}$. Specifically, we let $\mathcal{T}_I \coloneqq \{t \in [T] \mid \exists a \in [K] : \boldsymbol{\zeta}_{h(t)}(a) > 1\}$ be the set of initial rounds where the confidence intervals are large. Then, we let $\mathcal{T}_\mathrm{S} \coloneqq \{t \in [T] \setminus \mathcal{T}_\mathrm{I} \mid \sigma_{h(t)} \le 1/2\}$ and $\mathcal{T}_\mathrm{L} \coloneqq [T] \setminus (\mathcal{T}_\mathrm{I} \cup \mathcal{T}_\mathrm{S})$. We will bound the sum of $\sigma_{h(t)}$ over these three set separately.

For the first set $\mathcal{T}_\mathrm{I}$, the following holds with probability at least $1 - \delta'$:

$$\sum_{t\in\mathcal{T}_\mathrm{I}} \sigma_{h(t)} \le |\mathcal{T}_\mathrm{I}| \le \frac{9K \ln\left(\frac{1}{\delta'}\right)}{\lambda_{\min}^2 (\rho' - 2\delta')^2}, \tag{18}$$

where we exploited the fact that $\sigma_{h(t)} \le 1$ by Lemma E.1 and the upper bound on $|\mathcal{T}_\mathrm{I}|$ (Lemma E.4).

For the second set $\mathcal{T}_\mathrm{S}$, we observe the following:

$$\begin{aligned}
\sum_{t\in\mathcal{T}_\mathrm{S}} \sigma_{h(t)} &\le \frac{2}{\lambda_{\min}} \sum_{t\in\mathcal{T}_\mathrm{S}} \boldsymbol{\zeta}_{h(t)}^\top \widetilde{\boldsymbol{x}}_{h(t)} \\
&\le \frac{4}{\lambda_{\min}} \sum_{t\in\mathcal{T}_\mathrm{S}} (1 - \sigma_{h(t)}) \boldsymbol{\zeta}_{h(t)}^\top \widetilde{\boldsymbol{x}}_{h(t)},
\end{aligned}$$

where we applied Equation (16) and the fact that $1 - \sigma_{h(t)} \ge \frac{1}{2}$ for $t \in \mathcal{T}_\mathrm{S}$. As a result, by Lemma E.2 we can prove that:

$$\sum_{t\in\mathcal{T}_\mathrm{S}} \sigma_{h(t)} \le \frac{8}{\lambda_{\min}} \sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{(\rho' - 2\delta')^2}} \left(2\sqrt{KT} + \sqrt{2T \log\left(\frac{2}{\delta'}\right)}\right) \tag{19}$$

with probability at least $1-\delta'$.

Finally, we consider the set of rounds $\mathcal{T}_{\mathrm{L}} = [T] \setminus (\mathcal{T}_{\mathrm{I}} \cup \mathcal{T}_{\mathrm{S}})$. As $\sigma_{h(t)} \in [0,1]$ by Lemma E.1, we can upper bound $\sum_{t\in\mathcal{T}_{\mathrm{L}}} \sigma_h(t)$ with $|\mathcal{T}_{\mathrm{L}}|$. We thus provide an upper bound on $|\mathcal{T}_{\mathrm{L}}|$. In order to do so, consider the auxiliary set:

$$T_C \coloneqq \left\{ t \in [T] \mid \frac{\lambda_{\min}^2}{C} \le \boldsymbol{\zeta}_{h(t)}(a_t) \le 1 \right\}$$

for some constant $C \ge 5$ to be defined later. Let $\mathcal{F}_t$ be the filtration generated by the information up to the decision of $\boldsymbol{x}_t$. For any $t \in \mathcal{T}_{\mathrm{L}}$, by the inverse Markov inequality we have:

$$\begin{aligned}
\mathbb{P}\left(t \in \mathcal{T}_C \middle|\, t \in \mathcal{T}_{\mathrm{L}}, \mathcal{F}_t\right) &= \mathbb{P}\left(\boldsymbol{\zeta}_{h(t)}(a_t) \ge \frac{\lambda_{\min}^2}{C} \middle|\, t \in \mathcal{T}_{\mathrm{L}}, \mathcal{F}_t\right) \\
&\ge \frac{\mathbb{E}[\boldsymbol{\zeta}_{h(t)}(a_t)|\mathcal{F}_t] - \frac{\lambda_{\min}^2}{C}}{1 - \frac{\lambda_{\min}^2}{C}} \\
&\ge \frac{\frac{\lambda_{\min}^2}{4} - \frac{\lambda_{\min}^2}{C}}{1 - \frac{\lambda_{\min}^2}{C}} \\
&= \frac{(C-4)\lambda_{\min}^2}{4(C-\lambda_{\min}^2)} \ge \frac{\lambda_{\min}^2}{4}
\end{aligned}$$

where the first inequality holds by the inverse Markov inequality, the second by Lemma E.5, and the third as $C \ge 5$ and $\lambda_{\min} \le 1$. Therefore, we have:

$$\begin{aligned}
\mathbb{E}\left[\sum_{t\in[T]} \mathbb{I}\{t \in \mathcal{T}_C\} \mid \mathcal{T}_{\mathrm{L}}\right] &\ge \mathbb{E}\left[\sum_{t\in\mathcal{T}_{\mathrm{L}}} \mathbb{I}\{t \in \mathcal{T}_C\} \mid \mathcal{T}_{\mathrm{L}}\right] \\
&= \sum_{t\in\mathcal{T}_{\mathrm{L}}} \mathbb{E}\left[\mathbb{I}\{t \in \mathcal{T}_C\} \mid \mathcal{T}_{\mathrm{L}}\right] \\
&= \sum_{t\in\mathcal{T}_{\mathrm{L}}} \mathbb{E}\left[\mathbb{E}\left[\mathbb{I}\{t \in \mathcal{T}_C\} \mid \mathcal{F}_t\right] \mid \mathcal{T}_{\mathrm{L}}\right] \\
&= \sum_{t\in\mathcal{T}_{\mathrm{L}}} \mathbb{E}\left[\mathbb{P}(t \in \mathcal{T}_C|\mathcal{F}_t) \mid \mathcal{T}_{\mathrm{L}}\right] \\
&= \sum_{t\in\mathcal{T}_{\mathrm{L}}} \mathbb{P}(t \in \mathcal{T}_C|\mathcal{F}_t, t \in \mathcal{T}_{\mathrm{L}}) \\
&\ge \sum_{t\in\mathcal{T}_{\mathrm{L}}} \frac{\lambda_{\min}^2}{4} \\
&= |\mathcal{T}_{\mathrm{L}}| \frac{\lambda_{\min}^2}{4}.
\end{aligned}$$

At the same time, thanks to Lemma E.6, we have:

$$\mathbb{E}\left[\sum_{t\in[T]} \mathbb{I}\{t \in \mathcal{T}_C\} \mid \mathcal{T}_{\mathrm{L}}\right] \le \frac{16C^2}{\lambda_{\min}^4} \frac{\log\left(\frac{2}{\delta'}\right)}{(\rho' - 2\delta')^2}.$$

As a result:

$$\frac{16C^2}{\lambda_{\min}^4} \frac{\log\left(\frac{2}{\delta'}\right)}{(\rho' - 2\delta')^2} \ge |\mathcal{T}_{\mathrm{L}}| \frac{\lambda_{\min}^2}{4}$$

Taking $C = 5$ we get:

$$|\mathcal{T}_{\mathrm{L}}| \le \frac{1600}{\lambda_{\min}^6} \frac{\log\left(\frac{2}{\delta'}\right)}{(\rho' - 2\delta')^2}.$$

Consequently:

$$\sum_{t\in\mathcal{T}_{\mathrm{L}}} \sigma_{h(t)} \le |\mathcal{T}_{\mathrm{L}}| \le \frac{1600}{\lambda_{\min}^6} \frac{\log\left(\frac{2}{\delta'}\right)}{(\rho' - 2\delta')^2}. \tag{20}$$

Finally, if we put together Equation 18 and Equation (19), which independently hold with probability at least $1-\delta'$, and Equation (20), we have:

$$\begin{aligned}\sum_{t\in[T]}\sigma_{h(t)} &= \sum_{t\in\mathcal{T}_{\mathrm{I}}}\sigma_{h(t)} + \sum_{t\in\mathcal{T}_{\mathrm{S}}}\sigma_{h(t)} + \sum_{t\in\mathcal{T}_{\mathrm{L}}}\sigma_{h(t)} \\ &\leq \frac{9K\ln\left(\frac{1}{\delta'}\right)}{\lambda_{\min}^2(\rho'-2\delta')^2} + \frac{8}{\lambda_{\min}}\sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{(\rho'-2\delta')^2}}\left(2\sqrt{KT}+\sqrt{2T\log\left(\frac{2}{\delta'}\right)}\right)+\frac{1600\log\left(\frac{2}{\delta'}\right)}{\lambda_{\min}^6(\rho'-2\delta')^2} \\ &\leq \frac{8}{\lambda_{\min}}\sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{(\rho'-2\delta')^2}}\left(2\sqrt{KT}+\sqrt{2T\log\left(\frac{2}{\delta'}\right)}\right)+\frac{1610K\log\left(\frac{2}{\delta'}\right)}{\lambda_{\min}^6(\rho'-2\delta')^2}\end{aligned}$$

with probability at least $1-2\delta'$. □

Lemma E.8. *Conditioned on the event $\mathcal{E}$, the regret of Algorithm 4 satisfies:*

$$R_T \leq \frac{12}{\lambda_{\min}}\sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{(\rho'-\delta')^2}}\left(2\sqrt{KT}+\sqrt{2T\log\left(\frac{2}{\delta_{\mathrm{A}}}\right)}\right)+\frac{1610K\log\left(\frac{2}{\delta'}\right)}{\lambda_{\min}^6(\rho'-2\delta')^2}$$

with probability at least $1-3\delta'$.

Proof. We decompose the regret as follows:

$$\begin{aligned}R_T = \sum_{t=1}^{T}\left(\boldsymbol{r}^\top(\boldsymbol{x}^\star-\boldsymbol{x}_t)\right) &= \sum_{t=1}^{T}\left(\boldsymbol{r}^\top(\boldsymbol{x}^\star-\boldsymbol{x}^\circ_{h(t)}\right) \\ &= \sum_{t=1}^{T}\left(\boldsymbol{r}^\top(\boldsymbol{x}^\star-\sigma_{h(t)}\boldsymbol{x}^\diamond+(1-\sigma_{h(t)})\widetilde{\boldsymbol{x}}_{h(t)}\right) \\ &= \sum_{t=1}^{T}\left(\boldsymbol{r}^\top(\sigma_{h(t)}\boldsymbol{x}^\star+(1-\sigma_{h(t)})\boldsymbol{x}^\star-\sigma_{h(t)}\boldsymbol{x}^\diamond+(1-\sigma_{h(t)})\widetilde{\boldsymbol{x}}_{h(t)}\right) \\ &= \underbrace{\sum_{t=1}^{T}\sigma_{h(t)}\boldsymbol{r}^\top(\boldsymbol{x}^\star-\boldsymbol{x}^\diamond)}_{①}+\underbrace{\sum_{t=1}^{T}(1-\sigma_{h(t)})\boldsymbol{r}^\top(\boldsymbol{x}^\star-\widetilde{\boldsymbol{x}}_{h(t)})}_{②}.\end{aligned}$$

To bound ①, we employ Lemma E.7 as follows:

$$\begin{aligned}\sum_{t=1}^{T}\sigma_{h(t)}\boldsymbol{r}^\top(\boldsymbol{x}^\star-\boldsymbol{x}^\diamond) &= \sum_{t=1}^{T}\sigma_{h(t)}(\boldsymbol{r}^\top\boldsymbol{x}^\star-\boldsymbol{r}^\top\boldsymbol{x}^\diamond) \\ &\leq \sum_{t=1}^{T}\sigma_{h(t)} \\ &\leq \frac{8}{\lambda_{\min}}\sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{(\rho'-2\delta')^2}}\left(2\sqrt{KT}+\sqrt{2T\log\left(\frac{2}{\delta'}\right)}\right)+\frac{1610K\log\left(\frac{2}{\delta'}\right)}{\lambda_{\min}^6(\rho'-2\delta')^2},\end{aligned}$$

where the first inequality follows from the fact that $0\leq\boldsymbol{r}^\top\boldsymbol{x}^\diamond\leq\boldsymbol{r}^\top\boldsymbol{x}^\star\leq 1$, and the second one holds with probability $1-2\delta'$.

To bound ②, we employ Lemma E.3, which states that:

$$\sum_{t=1}^{T}(1-\sigma_{h(t)})\boldsymbol{r}^\top(\boldsymbol{x}^\star-\widetilde{\boldsymbol{x}}_{h(t)}) \leq 4\sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{(\rho'-\delta')^2}}\left(2\sqrt{KT}+\sqrt{2T\log\left(\frac{2}{\delta'}\right)}\right)$$

with probability at least $1-\delta'$.

Putting all together with an union bound:

$$R_T \leq \frac{12}{\lambda_{\min}} \sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{(\rho'-\delta')^2}} \left(2\sqrt{KT} + \sqrt{2T\log\left(\frac{2}{\delta'}\right)}\right) + \frac{1610K\log\left(\frac{2}{\delta'}\right)}{\lambda_{\min}^6(\rho'-2\delta')^2},$$

with probability at least $1-3\delta'$. □

Theorem 5.3. *Let $\delta \in (0,1)$ and $\rho \in (2\delta, 1)$. Then, with probability at least $1-\delta$,* `DEBORA-H` *suffers an expected cumulative regret bounded as:*

$$R_T \leq \widetilde{\mathcal{O}}\left(\frac{1}{\lambda_{\min}(\rho-2\delta)}\log\left(\frac{1}{\delta}\right) mK^2\sqrt{KT} + \gamma\right)$$

where γ does not depend on T.

Proof. By Lemma 5.2, Algorithm 4 has at most $H \leq K\lceil\log_2(T)\rceil$ epochs (excluding epoch $h=0$, where no estimator is computed from stochastic samples). At every epoch $h = 1, 2, \ldots, H$, the Algorithm employs `RepMean` to compute $(m+1)$ estimators for each action $a \in [K]$. According to Lemma 2.5, each estimator are within distance $\boldsymbol{\zeta}_h(a)$ from the estimated quantity with probability at least $1-\delta'$. By a union bound over the at most $K\lceil\log_2(T)\rceil$ epochs and $K(m+1)$ estimators for epoch, the probability of event $\mathcal{E}$ is at least $1-(m+1)K^2\lceil\log_2(T)\rceil\delta'$. As a result, by Lemma E.8, the regret of Algorithm 4 is:

$$R_T \leq \frac{12}{\lambda_{\min}} \sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{(\rho'-\delta')^2}} \left(2\sqrt{KT} + \sqrt{2T\log\left(\frac{2}{\delta'}\right)}\right) + \frac{1610K\log\left(\frac{2}{\delta'}\right)}{\lambda_{\min}^6(\rho'-2\delta')^2}$$

with probability at least $1-3\delta'-(m+1)K^2\lceil\log_2(T)\rceil\delta'$. We notice that:

$$3\delta' + (m+1)K^2\lceil\log_2(T)\rceil\delta' \leq 2(m+1)K^2\lceil\log_2(T)\rceil\delta',$$

as $K^2 \geq 4$ and $\log_2(T) \geq 1$. Since Algorithm 4 sets $\delta' := \frac{\delta'}{2(m+1)K^2\lceil\log_2(T)\rceil}$ and $\rho' := \frac{\rho'}{2(m+1)K^2\lceil\log_2(T)\rceil}$, simple computations show that its regret can be upper bounded as:

$$\begin{aligned} R_T &\leq \widetilde{\mathcal{O}}\left(\frac{mK^2}{\lambda_{\min}} \sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{(\rho-2\delta)^2}} \left(\sqrt{KT} + \sqrt{T\log\left(\frac{1}{\delta}\right)}\right) + \frac{m^2K^5\log\left(\frac{1}{\delta}\right)\log(T)}{\lambda_{\min}^6(\rho-2\delta)^2}\right) \\ &= \widetilde{\mathcal{O}}\left(\frac{1}{\lambda_{\min}(\rho-2\delta)}\log\left(\frac{1}{\delta}\right) mK^2\sqrt{KT} + \frac{m^2K^5}{\lambda_{\min}^6(\rho-2\delta)^2}\log\left(\frac{1}{\delta}\right)\log(T)\right) \end{aligned}$$

with probability at least $1-\delta$. □

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state all the main contributions made by the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: All the assumptions are clearly stated in Section 2.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All the theoretical results clearly state their assumptions, while all their proofs are provided in the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [N/A]

   Justification: The paper does *not* include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [N/A]

Justification: The paper does *not* include experiments.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [N/A]

   Justification: The paper does *not* include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: The paper does *not* include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: The paper does *not* include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed, since the work is mainly theoretical.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The paper does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.