# From External to Swap Regret 2.0: An Efficient Reduction for Large Action Spaces

Yuval Dagan* UC Berkeley
Constantinos Daskalakis† MIT CSAIL
Maxwell Fishelson‡ MIT CSAIL
Noah Golowich§ MIT CSAIL

December 6, 2023

**Abstract**

We provide a novel reduction from *swap-regret* minimization to *external-regret* minimization, which improves upon the classical reductions of Blum-Mansour [BM07] and Stoltz-Lugosi [SL05] in that it does not require finiteness of the space of actions. We show that, whenever there exists a no-external-regret algorithm for some hypothesis class, there must also exist a no-swap-regret algorithm for that same class. For the problem of learning with expert advice, our result implies that it is possible to guarantee that the swap regret is bounded by $\epsilon$ after $(\log N)^{\tilde{O}(1/\epsilon)}$ rounds and with $O(N)$ per iteration complexity, where $N$ is the number of experts, while the classical reductions of Blum-Mansour and Stoltz-Lugosi require at least $\Omega(N/\epsilon^2)$ rounds and at least $\Omega(N^3)$ total computational cost. Our result comes with an associated lower bound, which—in contrast to that in [BM07]—holds for *oblivious* and $\ell_1$*-constrained* adversaries and learners that can employ distributions over experts, showing that the number of rounds must be $\tilde{\Omega}(N/\epsilon^2)$ or exponential in $1/\epsilon$.

Our reduction implies that, if no-regret learning is possible in some game, then this game must have approximate *correlated equilibria*, of arbitrarily good approximation. This strengthens the folklore implication of no-regret learning that approximate *coarse* correlated equilibria exist. Importantly, it provides a sufficient condition for the existence of approximate correlated equilibrium which vastly extends the requirement that the action set is finite or the requirement that the action set is compact and the utility functions are continuous, allowing for games with finite Littlestone or finite sequential fat shattering dimension, thus answering a question left open by [DG22; AAD+23]. Moreover, it answers several outstanding questions about equilibrium computation and/or learning in games. In particular, for constant values of $\epsilon$: (a) we show that $\epsilon$-approximate correlated equilibria in *extensive-form games* can be computed efficiently, advancing a long-standing open problem for extensive-form games; see e.g. [VF08; FP23]; (b) we show that the query and communication complexities of computing $\epsilon$-approximate correlated equilibria in $N$-action normal-form games are $N \cdot \operatorname{poly}\log(N)$ and $\operatorname{poly}\log N$ respectively, advancing an open problem of [Bab20]; (c) we show that $\epsilon$-approximate correlated equilibria of sparsity $\operatorname{poly}\log N$ can be computed efficiently, advancing an open problem of [BBP14]; (d) finally, we show that in the adversarial bandit setting, sublinear swap regret can be achieved in only $\tilde{O}(N)$ rounds, advancing an open problem of [BM07; Ito20].

*Email: yuvald@berkeley.edu.
†Email: costis@csail.mit.edu. Supported by NSF Awards CCF-1901292, DMS-2022448, and DMS2134108, a Simons Investigator Award, the Simons Collaboration on the Theory of Algorithmic Fairness, and a DSTA grant.
‡Email: maxfish@mit.edu.
§Email: nzg@mit.edu. Supported by a Fannie & John Hertz Foundation Fellowship and an NSF Graduate Fellowship.

# 1 Introduction

*No-regret learning* has been a central topic of study in game theory and online learning over the last several decades [Han57; FL98; CL06]. In view of the worst-case nature of the associated learning guarantee, no-regret learning has found myriad applications in a variety of settings, with varying degrees of restriction on the adversary's behavior. They are also particularly salient in game theory due to their connection with decentralized equilibrium computation. Indeed, it is well understood that, if players in a normal-form game iteratively update their strategies using a no-regret learning algorithm, then the empirical distribution of their strategies over time converges to a type of correlated equilibrium, depending on the notion of regret used.

The most commonly studied type of regret, called *external regret*, measures the amount of extra utility that the agent could have gained if, instead of her realized sequence of strategies, she had instead played her best fixed action in hindsight. In a multi-agent interaction, if each agent uses a sublinear external regret learning algorithm to iteratively update her strategy, the empirical distribution of the agents' play converges to a *coarse correlated equilibrium* (CCE). A CCE is a correlated distribution over actions under which no player can improve her utility if, instead of playing according to the distribution, she unilaterally switches to playing any single fixed action. CCEs are a convex relaxation of *Nash equilibria*, which are computationally intractable even for normal-form games with a finite number of actions per player [DGP09; CDT09]. While a plethora of efficient algorithms for minimizing external regret are known even when the size of the game is large (see e.g. [FL98; CL06; BC+12]), the twin notions of external regret and coarse correlated equilibrium are too weak for many applications. In particular, the notion of CCE does not capture the fact that the action sampled from the CCE distribution for some player may leak information about what actions were sampled for the other players, which the player could potentially exploit to improve her utility.

Using the perspective of Bayesian rationality, Aumann introduced the concept of *correlated equilibrium* (CE), which corrects for this deficit [Aum74]. A CE is a correlated distribution with the property that the action sampled for each player maximizes her expected utility against the distribution over actions sampled for the other players, *conditioning on the action sampled for this player*. Like CCE, the concept of CE is a convex relaxation of Nash equilibrium, and it can be reached in a decentralized manner by averaging the empirical play of algorithms which have sublinear *swap regret*. This measures the amount of extra utility that the agent could have gained, in hindsight, if she were to go back in time and transform the strategies that she played using the best, fixed *swap function* (see Definition 2.2). The stronger nature of swap regret leads it to have numerous applications, including in calibration and multicalibration [GHK+23; KLST23] and Bayesian games [MMSS22], amongst others.

## 1.1 Swap regret: challenges with large action spaces

Despite the more appealing guarantees satisfied by swap regret minimization and its twin notion of CE, no-swap regret learning algorithms have not been as widely adopted as no-external regret ones. This is due in part to the substantially inferior quantitative guarantees offered by the best-known swap-regret-minimizing algorithms in terms of their dependence in the number of actions available to the learner. In particular, existing algorithms are inefficient in many settings of interest where the action space is exponentially large in the game's description complexity, or even infinite. To illustrate, we first consider the case of no-regret learning with a finite set of $N$ actions, which is

known as the "experts setting." Standard external-regret-minimizing algorithms, such as exponential weights [CL06], guarantee that the average external regret over $T$ rounds is bounded by $\epsilon$ as long as $T \gtrsim \frac{\log N}{\epsilon^2}$.[1] In contrast, the best-known swap-regret-minimizing algorithms, which are all based on generic reductions from swap regret minimization to external regret minimization [SL05; BM07], guarantee that the average swap regret over $T$ rounds is $\epsilon$ as long as $T \gtrsim \frac{N \log N}{\epsilon^2}$. Thus, prior work left an *exponential gap* between the best-known algorithms for swap and external regret. It was explicitly asked by Blum and Mansour [BM07] if this gap could be improved. This gap is particularly noteworthy in light of many recent applications of no-regret learning, such as for solving games such as Poker [BS19] and Diplomacy [BBD+22], all of which have the property that $N$ is moderate or large.

Prior work also left a polynomial-sized gap in the *bandit* setting, in which the learner must choose a single action each round and only receives the utility for that action. While it is known that $T \gtrsim \frac{N^2 \log N}{\epsilon^2}$ rounds suffice [JLWY22; Ito20] to ensure that swap regret is bounded by $\epsilon$, the best known lower bound was that $\frac{N \log N}{\epsilon^2}$ rounds are necessary [Ito20; BM07]. The bandit setting is particularly useful due to its applications in reinforcement learning [JLWY22] and related areas.

Prior to the present work, the gap between swap regret and external regret was even larger in settings where the number of actions available to the learner is unbounded or infinite. For instance, suppose that each agent's action space is the set of parameters of a neural network: multi-agent interactions in which each agent chooses a neural network can be used to model tasks such as training generative adversarial networks [GPM+14], autonomous driving [SSS16], or economic decision making [ZTS+20]. In these cases, the number of possible networks is very large. In a more general setting, the action space is typically assumed to be constrained by a combinatorial complexity measure, such as the *Littlestone dimension* or *sequential fat shattering dimension* (see Section B for formal definitions). In particular, if the learner's action space has Littlestone dimension $L$, then it was known [BPS09; ABD+21] that as long as the number $T$ of rounds satisfies $T \geq \frac{L}{\epsilon^2}$, there is an algorithm which achieves at most $\epsilon$ external regret.[2] Since the reductions of [SL05; BM07] for bounding swap regret assume that the number $N$ of actions is bounded, prior to our work it was not known whether any class of finite Littlestone dimension has an algorithm with $o(T)$ swap regret, leaving open the possibility of an *infinite* gap between swap and external regrets for classes of finite Littlestone dimension.

**Gaps in equilibrium computation.** The above gaps between swap and external regret also manifest as gaps between the best known results for computing $\epsilon$-approximate CE and CCE in various models of computation. We improve upon these gaps in the following settings:

- *Normal-form games with $N$ actions.* We consider two computation problems. For simplicity we assume the number of players and $\epsilon$ are constants.
  - In the *communication complexity* model of computation (Definition 2.6), $\epsilon$-CCE may be computed with $O(\log^2 N)$ bits of communication using no-external regret algorithms together with a sampling procedure. In contrast, prior to this work, the best known bound for $\epsilon$-CE was exponentially worse, $O(N \log^2 N)$, using the swap regret algorithm of [BM07].

[1] We consider normalized regret throughout the paper, i.e., we divide the cumulative regret by the number of rounds $T$.

[2] This bound is optimal; see [BPS09].

  - In the *query complexity* model of computation (Definition 2.7), $\epsilon$-CCE may be computed using $O(N \log N)$ queries. Prior to this work, the best known bound of $O(N^2 \log N)$ was quadratically worse for $\epsilon$-CE.
  - Finally, $\epsilon$-CCE which are $\text{poly}\log(N)$-sparse may be computed in polynomial time [BBP14], whereas prior to this work, it was unknown how to efficiently compute $\epsilon$-CE which are $o(N)$-sparse, marking another exponential gap (in the sparsity).

- In *infinite games* of Littlestone dimension $L < \infty$, for constant $\epsilon > 0$, $\epsilon$-CCE may be found in a decentralized manner by running $O(L)$ rounds of no-external regret algorithms [DG22]. In contrast, prior to our work it was not known if $\epsilon$-CE even *exist* in games of finite Littlestone dimension.

- Finally, in *extensive form games* with description length $n$ denoting the size of the tree, for which the number of actions[3] typically scales as $N = \exp(\Theta(n))$, $\epsilon$-CCE may be computed in $\text{poly}(n)$ time (e.g., [FLLK22]). However, prior to this work, the best known algorithms for computing $\epsilon$-CE took time exponential in $n$. Determining the complexity of $\epsilon$-CE was a well-known open question in this field; see e.g. [VF08; FP23].[4]

### 1.2 Main results: near-optimal upper and lower bounds for swap regret

Our main upper bound is a new reduction from swap regret to external regret: any no-external regret learning algorithm can be transformed into a no-distributional swap regret learner. (These regret notions are formally defined in Section 2.1.) We assume that a learner chooses, in each iteration $t \in [T]$, a distribution $\mathbf{x}^{(t)} \in \Delta_{\mathcal{X}}$ over a set of actions $\mathcal{X}$. After observing $\mathbf{x}^{(t)}$, an adversary selects a reward function $\mathbf{f}^{(t)} : \mathcal{X} \to \mathbb{R}$, and the learner receives the reward $\mathbf{f}^{(t)}(\mathbf{x}^{(t)}) = \mathbb{E}_{s^{(t)} \sim \mathbf{x}^{(t)}} \mathbf{f}^{(t)}[s^{(t)}]$. We assume the adversary's choices $\mathbf{f}^{(t)}$ are constrained to lie in some convex function class $\mathcal{F} \subset [0,1]^{\mathcal{X}}$.

**Theorem 1.1** (Informal version of Theorem 3.1)**.** *Let $d, M \in \mathbb{N}$ be given, and suppose that there is a learner for some function class $\mathcal{F}$ which achieves external regret of $\epsilon$ after $M$ iterations. Then there is a learner for $\mathcal{F}$ (*`TreeSwap`*; Algorithm 1) which achieves a swap regret of at most $\epsilon + \frac{1}{d}$ after $T = M^d$ iterations.*

*If the per-iteration runtime complexity of the external-regret learner is $C$, then the swap regret learner* `TreeSwap` *has a per-iteration amortized runtime complexity of $O(C)$.*

Notice that the swap regret of `TreeSwap` depends only on the external regret of the assumed learner, and is *independent of the number of actions* of the learner. In particular, it holds also for exponentially large or even infinite function classes.

**Applications: concrete swap regret bounds.** As applications of Theorem 1.1, in the setting of constant $\epsilon$, we are able to close all of the gaps discussed for the regret minimization and equilibrium computation problems in Section 1.1. We begin with the case that the learner has $N$ actions, also

[3] An action is specified by a contingency plan, mapping each information set to an outgoing edge at that information set.

[4] To be clear, $\epsilon$-CE here refers to the notion of $\epsilon$-approximate *normal-form correlated equilibrium* (sometimes denoted $\epsilon$-NFCE), as opposed to relaxations of this notion, such as extensive-form correlated equilibrium, which have been recently proposed, motivated in part by the apparent intractability of $\epsilon$-NFCE [VF08].

known as *learning with expert advice*. By applying Theorem 1.1 with action set $\mathcal{X} = [N]$, and reward class given by all $[0,1]$-bounded functions, i.e., $\mathcal{F} = [0,1]^{[N]}$, we obtain:

**Corollary 1.2** (Upper bound for finite action swap regret; informal version of Corollary 3.2)**.** *Fix $N \in \mathbb{N}$ and $\epsilon \in (0,1)$, and consider the setting of online learning with $N$ actions. Then for any $T$ satisfying $T \geq (\log(N)/\epsilon^2)^{\Omega(1/\epsilon)}$, there is an algorithm that, when faced with any adaptive adversary, has swap regret bounded above by $\epsilon$. Further, the amortized per-iteration runtime of the algorithm is $O(N)$, its worst-iteration runtime is $O(N/\epsilon)$ and its space-complexity is $O(N/\epsilon)$.*

In the regime of constant $\epsilon$, Corollary 1.2 improves on the previously best-known complexity of $T \geq \tilde{\Omega}(N/\epsilon^2)$, providing an exponential improvement in the dependence on $N$. We note that $N/\epsilon^2$ is tight for all $\epsilon$ in the *non-distributional* setting, where the learner is allowed to randomize over her actions but has to play a concrete action rather than a probability distribution [BM07]. Thus, Theorem 1.2 shows that for a constant $\epsilon$, a distributional swap regret of at most $\epsilon$ can be achieved with exponentially fewer rounds. Another advantage of our result is an improved total runtime of $\tilde{O}(N)$ for constant $\epsilon$, compared to the previous $\Omega(N^3)$ runtime of [BM07], which answers an open question from that paper for constant $\epsilon$.

Next, we apply Theorem 1.1 to an arbitrary function class $\mathcal{F} \subset \{0,1\}^{\mathcal{X}}$ whose dual has finite Littlestone dimension. That is, the class of functions indexed by actions of the learner, which, via slight abuse of notation, we denote by $\mathcal{X} := \{f \mapsto f(s) \ : \ s \in \mathcal{X}\} \subset \{0,1\}^{\mathcal{F}}$, [5]

**Corollary 1.3** (Swap regret for Littlestone classes; informal version of Corollaries 3.4 and 3.6)**.** *If the class $\mathcal{X}$ has Littlestone dimension at most $L$, then for any $T \geq (L/\epsilon^2)^{\Omega(1/\epsilon)}$, there is a learner whose swap regret is at most $\epsilon$. In particular, games with finite Littlestone dimension admit no-swap regret learners and thus have $\epsilon$-approximate CE for all $\epsilon > 0$.*

We remark that even the *existence* of approximate CEs in games of finite Littlestone dimension was previously unknown. We refer the reader to Section B for a definition of Littlestone dimension and its real-valued generalizations.

Finally, we prove an upper bound on the swap regret in the bandit setting that is tight up to $\operatorname{poly}\log N$ factors when $\epsilon = O(1)$. While the result is not a direct consequence of Theorem 1.1, the overall structure of the algorithm and analysis are similar:

**Theorem 1.4** (Bandit swap regret; Informal version of Theorem 3.12)**.** *Let $N \in \mathbb{N}, \epsilon \in (0,1)$ be given, and consider any $T \geq N \cdot (\log(N)/\epsilon)^{O(1/\epsilon)}$. Then there is an algorithm in the* adversarial bandit *setting with $N$ actions (*`BanditTreeSwap`*; Algorithm 4) which achieves swap regret bounded above by $\epsilon$ after $T$ iterations.*

Concretely, for $\epsilon = O(1)$, Theorem 1.4 guarantees that $\tilde{O}(N)$ rounds suffice to achieve swap regret of at most $\epsilon$. Interestingly, this implies that, for obtaining swap regret bounded by $\epsilon = O(1)$, there is only a polylogarithmic gap between the number of rounds needed in the adversarial bandit setting and the full-information non-distributional setting [BM07]. This is in stark contrast to the situation for *external regret*, for which there is an *exponential* gap between the full-information non-distributional setting (where $O(\log N)$ rounds suffice) and the adversarial bandit setting (where $\Omega(N)$ rounds are needed) [LS20]. Finally, we remark that our algorithm for the bandit setting is readily seen to be computationally efficient.

[5] Technically, in order to ensure convexity of $\mathcal{F}$, we need to apply Theorem 1.1 to the *convex hull* of $\mathcal{F}$. Doing so does not materially affect the guarantees; see Section 3.2 for a more detailed discussion.

**Applications: equilibrium computation.** Next, we discuss implications of Corollary 1.2 for equilibrium computation. By considering the setting where players in a normal-form game run (a slight variant of) the algorithm of Corollary 1.2, we may obtain low query and communication protocols for learning in normal-form games.

**Corollary 1.5** (Query and communication complexity upper bound; informal version of Corollaries 3.7 and 3.9)**.** *In normal-form games with a constant number of players and $N$ actions per player, the communication complexity of computing an $\epsilon$-approximate CE is $\log(N)^{\tilde{O}(1/\epsilon)}$ and the query complexity of computing an $\epsilon$-approximate CE is $N \cdot \log(N)^{\tilde{O}(1/\epsilon)}$.*

Finally, we remark that our main reduction can be used to obtain efficient algorithms for computing $\epsilon$-CE when $N$ is exponentially large if there are nevertheless efficient external regret algorithms. This is the case in particular for the setting of extensive form games [FLLK22; KWKS20; FKS21]:

**Corollary 1.6** (Extensive form games; informal version of Theorem 3.11)**.** *For any constant $\epsilon$, there is an algorithm which computes an $\epsilon$-approximate CE of any given extensive form game, with runtime polynomial in the representation of the game (i.e., polynomial in the number of nodes in the game tree and in the number of outgoing edges per node).*

Corollary 1.6 is an immediate consequence of Theorem 1.1 (i.e., Theorem 3.1) and the fact that there are efficient external regret minimization algorithms in extensive-form games. This is classically known as a consequence of the *counterfactual regret minimization* algorithm, i.e., Theorem 4 of [ZJBP07], or improved recent results, such as Theorem 5.5 of [FLLK22], as well as [CMBG19; FKS21; KWKS20].

**Near-matching lower bounds.** Theorem 1.1 and Corollary 1.2 require the number of rounds $T$ to be exponential in $1/\epsilon$, where $\epsilon$ denotes the desired swap regret. The following lower bound shows this dependence is necessary, even facing an *oblivious* adversary that is constrained to choose reward vectors with *constant $\ell_1$ norm*:

**Theorem 1.7** (Lower bound for swap regret with oblivious adversary; restatement of Corollary 4.2)**.** *Fix $N \in \mathbb{N}$, $\epsilon \in (0,1)$, and let $T$ be any number of rounds satisfying*

$$T \leq O(1) \cdot \min\left\{\exp(O(\epsilon^{-1/6})), \frac{N}{\log^{12}(N) \cdot \epsilon^2}\right\}. \tag{1}$$

*Then, there exists an* oblivious *adversary on the function class $\mathcal{F} = \left\{\mathbf{f} \in [0,1]^N \middle| \|\mathbf{f}\|_1 \leq 1\right\}$ such that any learning algorithm run over $T$ steps will incur swap regret at least $\epsilon$.*[6]

Theorem 1.7 establishes:

- The first $\tilde{\Omega}\left(\min(1, \sqrt{N/T})\right)$ swap regret lower bound for *distributional swap regret.*
- The first $\tilde{\Omega}\left(\min(1, \sqrt{N/T})\right)$ swap regret lower bound achieved by an oblivious adversary. In particular, the adversary samples reward functions $\mathbf{f}^{(1:T)}$ from some fixed distribution before the first round of learning, independently of the actions of the learner. Moreover, this distribution is independent of the description of the learning algorithm.

[6] If we replace the requirement that $\|\mathbf{f}\|_1 \leq 1$ with $\|\mathbf{f}\|_\infty \leq 1$, then the bounds are slightly improved, with 1/6 replaced with 1/5 and 12 with 10.

- The first $\tilde{\Omega}\left(\min(1, \sqrt{N/T})\right)$ swap regret lower bound from an adversary that plays distributions over a function class of *constant* Littlestone dimension (namely, the class of point functions on $[N]$, which has Littlestone dimension 1).

Finally, while the lower bound of $\exp(\epsilon^{-1/6})$ rounds necessary (to ensure swap regret is bounded by $\epsilon$) from Theorem 1.7 does not quite match the upper bound of $\exp(\epsilon^{-1})$ (from Corollary 1.2; ignoring $\log N$ factors), we can improve the lower bound somewhat if we allow the adversary to be adaptive. In particular, in Theorem C.1, we give an *entirely different* (and somewhat simpler) construction which shows that $T \geq \exp(\Omega(\epsilon^{-3}))$ rounds are necessary to ensure that swap regret is bounded above by $\epsilon$.

**Concurrent work.** We have been recently made aware of concurrent work by Peng and Rubinstein [PR23], which proves similar upper and lower bounds to Theorems 1.1 and 1.7. Moreover, they derive a similar set of applications for equilibrium computation problems.

### 1.3 Proof sketch of the upper bound (Theorem 1.1)

We overview the proof of Theorem 1.1. Recall that we are given $M, d \in \mathbb{N}$, and will construct a swap regret learner for $T = M^d$ rounds. We assume access to a no-external regret learner ($\texttt{Alg}_{\mathsf{Ext}}$) that, over $M$ rounds, produces a sequence of distributions which has an external regret of at most $\epsilon \in (0, 1)$. We will show that there is an algorithm `TreeSwap` with swap regret at most $O(\epsilon + \frac{1}{d})$.

`TreeSwap` is defined formally in Algorithm 1. The algorithm simulates multiple instances of $\texttt{Alg}_{\mathsf{Ext}}$ at *levels* $i = 0, 1, \ldots, d-1$, which are arranged as the nodes a depth-$d$ $M$-ary tree. We traverse the $T = M^d$ leaves of the tree in order, one per round. At each round $t$, the `TreeSwap` algorithm outputs the uniform mixture over the $d$ distributions produced by the $\texttt{Alg}_{\mathsf{Ext}}$ instances on the root-to-leaf path for the current leaf.

**Updating $\texttt{Alg}_{\mathsf{Ext}}$ instances.** Next we describe how the $\texttt{Alg}_{\mathsf{Ext}}$ instances at each node of the tree are updated over the course of the $T$ rounds. Notice that the $M^i$ instances of $\texttt{Alg}_{\mathsf{Ext}}$ at each level $i$ are used during a disjoint set of $M^{d-i}$ consecutive rounds: the first algorithm in level $i$ is used during rounds $1, \ldots, M^{d-i}$, the second during rounds $M^{d-i} + 1, \ldots, 2M^{d-i}$, and so on. Each of these $\texttt{Alg}_{\mathsf{Ext}}$ instances will be run in a *lazy* fashion, only producing $M$ different distributions over the corresponding $M^{d-i}$ rounds. The first algorithm in level $i$ will be called to produce a distribution at round 1, and then play that distribution repeatedly for rounds $1, \ldots, M^{d-i-1}$. At round $M^{d-i-1}$, we finally update the state of the algorithm based on the average reward over the previous $M^{d-i-1}$ rounds. The algorithm then produces a new distribution, which it plays for rounds $M^{d-i-1} + 1, \ldots, 2M^{d-i-1}$, and so on. All algorithms in level $i$ will be run in this way: updating every $M^{d-i-1}$ rounds on an *average reward function* from the previous $M^{d-i-1}$ rounds. According to the guarantee of our external regret algorithm, each of these instances will have external regret bounded above by $\epsilon$ relative to the $M$ distributions it produces and the $M$ average reward functions on which it updates.

**Swap regret bound.** To bound the swap regret of our algorithm, let us first denote by $R_i$ the average reward of all the algorithms $\texttt{Alg}_{\mathsf{Ext}}$ in level $i$ over all $T$ rounds. Further, for each $i = 0, \ldots, d$, we define $S_i$ in the following manner. For each block of rounds of size $M^{d-i}$, consider the average

reward of the best fixed action in hindsight; then we define $S_i$ to be the average of these best-in-hindsight rewards over all blocks at level $i$. By the external regret guarantee of $\texttt{Alg}_{\texttt{Ext}}$, we know that $S_i - R_i \leq \epsilon$. This is due to the fact that each level-$i$ algorithm is run during a block of $M^{d-i}$ rounds, and therefore competes with the best fixed action over that block of rounds. Moreover, the contribution to the swap regret of `TreeSwap` from all algorithms at level $i$ is at most $S_{i+1} - R_i$. This is due to the fact that these level-$i$ algorithms repeatedly play actions for blocks of $M^{d-i-1}$ rounds, and so the best swaps of these actions correspond to the best fixed actions over the blocks of that length. The total swap regret is then bounded by

$$\frac{1}{d}\sum_{i=0}^{d-1}(S_{i+1} - R_i) = \frac{1}{d}\sum_{i=0}^{d-1}(S_i - R_i) + \frac{S_d - S_0}{d} \leq \epsilon + \frac{1}{d},$$

where we used that $S_i - R_i \leq \epsilon$ and that $S_d - S_0 \leq 1$ since the utilities are bounded between 0 and 1. This concludes the proof.

### 1.4 Proof sketch for the lower bound (Theorem 1.7)

To prove Theorem 1.7, we consider two cases depending on the values of $N, T$ (which correspond to which of the terms on the right-hand side of (1) is larger):

**Case 1: $N \geq 4T$.** As a warm-up, we present a strategy for the adversary that does not quite work. Then, we show how to fix it, describing a true strategy that achieves the desired lower bound. In both the warm-up and true strategies, we will consider an adversary that selects "point function" rewards at each time step $t$: one action $u^{(t)} \in [N]$ will receive a reward of 1, and all other actions 0. To describe these strategies, we will relate the actions to vertices in a full binary tree. Assume that $T = 2^D$ for some $D \in \mathbb{N}$. Consider a full binary tree of depth $D$, containing $2^{D+1} - 1$ vertices, and denote its vertex set by $V$. In our warm-up construction, each vertex will correspond to a single-action.[7] That is, in each round $t$, the learner plays a vertex $v^{(t)} \in V$ and the adversary plays a vertex $u^{(t)} \in V$. The reward of the learner is $\mathbb{1}[v^{(t)} = u^{(t)}]$. While our lower bound is valid also for the distributional setting, we analyze for simplicity the case where the learner has to play a concrete action in each round. However, the same proof goes through if they are allowed to output a distribution over vertices. Here is the strategy of the adversary: let us order the children of each internal node by 'left' and 'right'. This will create an ordering over the root-to-leaf paths in the tree: the first path goes left until reaching the leaf, the second path goes left except for the last step that is taken right, etc. Enumerate the paths by indices in $[T]$ according to this ordering, where path $t$ is called $P_t$. For each time step $t$, the adversary will select at random a vertex, out of the $D + 1$ vertices in path $P_t$, according to the following distribution: the probability of the vertex at depth $i$ is $(i + 1)/(1 + 2 + \cdots + (D + 1))$. The important property here, is that vertices get higher weight as we go down the tree.

Let us analyze the swap regret of any learner facing this adversary. Recall that this approach does not quite work for the adversary, but we will show how to fix it. At a high level, the goal of the adversary strategy is to increase swap regret every round $t$ as follows.

- If the learner plays an internal node on $P_t$, they will incur swap regret to the node at depth one greater on $P_t$, which gets slightly more expected reward.

[7] Eventually, we will consider a construction wherein each vertex corresponds to two actions.

- If the learner plays the leaf node of $P_t$, there is a constant probability that the adversary will not play this leaf. In this case, the learner will incur a swap regret from that leaf.
- If the learner plays a node not on $P_t$, they will receive no reward and incur swap regret.

However, the problem is that the learner will later have a chance to undo this incurred swap regret. For an internal node $v$, let $[\underline{t}_v, \bar{t}_v]$ be the interval of time steps for which $v$ is on $P_t$. Let's say the learner plays $v$ heavily during the first half of these time steps $[\underline{t}_v, (\underline{t}_v + \bar{t}_v)/2]$: the times in which the left child of $v$ is present on $P_t$. The learner will incur swap regret from $v$ to its left child. However, let's say the learner continues to play $v$ for much of the second half of the interval $[(\underline{t}_v + \bar{t}_v)/2, \bar{t}_v]$. During these times, the adversary never plays the left child of $v$, while continuing to play $v$ with some probability, undoing the swap regret of $v$.

To account for this, the adversary actually plays the following "true" strategy instead. In this strategy, each node is associated with two actions: $v, \dot{v}$. During the first half $[\underline{t}_v, (\underline{t}_v + \bar{t}_v)/2]$, as before, the adversary will choose $v$ with probability $(\text{depth}(v) + 1)/(1 + \cdots + (D + 1))$. However, with probability $1/2$, the adversary replaces $v$ with $\dot{v}$ at the halfway point $t = (\underline{t}_v + \bar{t}_v)/2$. That means, with probability $1/2$, for the second half $[(\underline{t}_v + \bar{t}_v)/2, \bar{t}_v]$, the adversary will choose $\dot{v}$ with probability $(\text{depth}(v) + 1)/(1 + \cdots + (D + 1))$ and never choose $v$. On the other hand, with probability $1/2$ there is no replacement, and the adversary continues to select $v$ with probability $(\text{depth}(v) + 1)/(1 + \cdots + (D + 1))$, not $\dot{v}$. This accomplishes the following. With probability $1/4$, $v$ will get replaced at the halfway point $t = (\underline{t}_v + \bar{t}_v)/2$ but the left child of $v$ will *not* get replaced at its halfway point $t = (3/4)\underline{t}_v + (1/4)\bar{t}_v$. In this event, which happens with constant probability, both $v$ and its left child will be played with non-zero probability over the interval $[\underline{t}_v, (\underline{t}_v + \bar{t}_v)/2]$ and at no other time steps. Thus, the learner will not have a chance to undo the swap regret.

The formal version of this argument, presented in Section 4, breaks into case work. We lower bound the swap regret of action $v$ by considering the reward of swapping $v$ with the best of the 4 actions associated with its 2 children. In addition, we consider a swap from $v$ to the root of the tree, in the event that $v$ is played on many rounds outside of the interval $[\underline{t}_v, \bar{t}_v]$. Bounds for each case culminate in the following. Letting $X_v$ be the total number of rounds the learner plays action $v$, we show that the best swap of action $v$ increases expected total reward by $\tilde{\Omega}(X_v)$. Thus, the total swap regret of the learner would be $\tilde{\Omega}\left(\sum_v X_v\right) = \tilde{\Omega}\left(T\right)$, and her average swap regret would be $\tilde{\Omega}\left(1\right) = \Omega\left(\frac{1}{\text{polylog}(T)}\right)$, which is at least $\epsilon$ for $T \leq \exp(\text{poly } 1/\epsilon)$.

**Case 2: $N < 4T$.** This case is very similar to the first. In fact, in Section 4, we define the adversary strategy in a general way that avoids breaking into cases manually. The key difference in this case is that we don't have enough actions to associate 2 actions with each node of a full binary tree with $T$ leaves. In this case, we consider a full binary tree with $N/4$ leaves. We again have an adversary that iterates through the root-to-leaf paths in DFS order. In this case though, each iteration corresponds to a batch in which the adversary plays a distribution over that root-to-leaf path repeatedly for $4T/N$ time steps.

The other key difference here is that we need to associate each leaf with two actions $\ell, \dot{\ell}$. As discussed before, there is a single coin flip for each of the internal nodes $v$ that determines if it gets replaced at time $t = (\underline{t}_v + \bar{t}_v)/2$. On the other hand, for leaf nodes $\ell$, we have a coin flip at every single time step in its batch, determining which of $\ell, \dot{\ell}$ will be played with non-zero probability. Letting $X_\ell$ be the total number of rounds the learner plays $\ell$, due to the random deviation in the selection of the adversary, the expected swap regret to $\dot{\ell}$ in $\tilde{\Omega}\left(\sqrt{X_\ell}\right)$. Thus, the total swap regret

of a learner that plays only leaf actions over all $N/4$ batches will be $\tilde{\Omega}\left(\sum_\ell \sqrt{T/N}\right) = \tilde{\Omega}\left(\sqrt{NT}\right)$ and her average swap regret will be $\tilde{\Omega}\left(\sqrt{N/T}\right)$, as desired.

### 1.5 Discussion

We next compare the guarantees of Corollary 1.2, Theorem 1.7, and Theorem C.1 (recall that Theorem C.1 yields a quantitatively stronger lower bound than Theorem 1.7 with the stronger notion of *adaptive* adversary). Let $\mathscr{M}(N, \epsilon)$ denote the smallest $T_0 \in \mathbb{N}$ so that, for all $T \geq T_0$, there is a learning algorithm whose action set is $[N]$ and for which the swap regret over $T$ rounds is bounded above by $\epsilon$. Then by Corollary 1.2 and Theorems 1.7 and C.1,[8]

$$\begin{aligned} &\Omega(1) \cdot \min\left\{\frac{\log N}{\epsilon^2} + 2^{\Omega(\epsilon^{-1/3})}, \frac{N}{\log^{10}(N) \cdot \epsilon^2}\right\} \\ &\leq \mathscr{M}(N, \epsilon) \leq O(1) \cdot \min\left\{(\log(N)/\epsilon^2)^{O(1/\epsilon)}, \frac{N \log N}{\epsilon^2}\right\}. \end{aligned} \tag{2}$$

The second terms in the upper and lower bounds in (2) differ by a $\operatorname{poly}\log(N)$ factor, which is insignificant compared to $N$. The first terms differ in that (a) the $\log(N)$ is in the base of the exponent in the upper bound but not the lower bound, and (b) the exponent in the lower bound is $\epsilon^{-1/3}$ but is $\epsilon^{-1}$ in the upper bound. We remark that the term $2^{\Omega(\epsilon^{-1/3})}$ in the lower bound comes from Theorem C.1, which is stronger than the bound of $2^{\Omega(\epsilon^{-1/6})}$ from Theorem 1.7.

The swap regret bound of Corollary 1.2 improves upon those of of Stoltz-Lugosi and of Blum-Mansour [SL05; BM07] when the accuracy parameter $\epsilon$ and number of actions $N$ satisfy $\epsilon \gg \frac{\log\log N}{\log N}$. In particular, for $\epsilon = O(1)$, our reduction bounds swap regret above by $\epsilon$ via an efficient algorithm with $T \leq \operatorname{poly}\log N$ rounds, whereas [SL05; BM07] require $T \geq \tilde{\Omega}(N)$.

**Outline of the paper.** After stating preliminaries in Section 2, we prove our main reduction from swap to external regret (Theorem 1.1) in Section 3. Then, in Sections 3.1 to 3.3 and 3.5, we provide applications of this reduction. In Section 4 we prove our main lower bound (Theorem 1.7); part of the proof is deferred to Appendix A. Finally, in Appendix C, we prove our alternative adaptive lower bound with superior rates.

## 2 Preliminaries

### 2.1 Online Learning

The setting of online learning entails a repeated interaction between a learner and adversary over $T$ rounds. For each time step, $t \in [T]$, the learner, whose action space is denoted by $\mathcal{X}$, selects a distribution $\mathbf{x}^{(t)} \in \Delta_{\mathcal{X}}$, and the adversary responds with a reward function $\mathbf{f}^{(t)} \in \mathcal{F}$ where $\mathcal{F} \subseteq [0,1]^{\mathcal{X}}$. The learner receives reward $\mathbf{f}^{(t)}(\mathbf{x}^{(t)}) := \mathbb{E}_{s\sim\mathbf{x}^{(t)}}[\mathbf{f}^{(t)}[s]]$. (For $f \in \mathcal{F}$, $s \in \mathcal{S}$, and $\mathbf{x} \in \Delta_{\mathcal{X}}$, we use square brackets to denote $f[s]$ and parentheses to denote $f(\mathbf{x})$.) To avoid measurability issues, we assume that $\mathcal{X}$ is countable and equipped with the discrete sigma algebra.

[8] The $\frac{\log N}{\epsilon^2}$ term in the lower bound of (2) comes the classic external regret lower bound. The second term in the minimum of the upper bound of (2) comes from the Blum-Mansour algorithm [BM07].

As an example, consider the classical "experts" setting, in which the learner selects a distribution $\mathbf{x}^{(t)} \in \Delta_N$ over $N$ actions and the adversary selects an arbitrary reward vector $\mathbf{f}^{(t)} \in [0,1]^N$. Here we have $\mathcal{X} = [N]$ and $\mathcal{F} = [0,1]^N$. The learner's reward at each round $t$ is given by $\langle \mathbf{f}^{(t)}, \mathbf{x}^{(t)} \rangle$.

The learner's performance over the $T$ rounds of learning is typically evaluated via "regret": a comparison between the total reward obtained by the learner and some benchmark. There are several notions of regret, depending on the particular benchmark. The *external regret* of the learner is the difference between her total reward and the best reward obtained by a fixed action $s^\star \in \mathcal{X}$ in hindsight:

**Definition 2.1.** Given sequences $\mathbf{x}^{(1:T)} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(T)})$ and $\mathbf{f}^{(1:T)} = (\mathbf{f}^{(1)}, \dots, \mathbf{f}^{(T)})$ of play by the learner and adversary, the *external regret* corresponding to these sequences is

$$\textbf{ExtRegret}\left(\mathbf{x}^{(1:T)}, \mathbf{f}^{(1:T)}\right) := \sup_{s^\star \in \mathcal{X}} \frac{1}{T} \sum_{t=1}^{T} \left(\mathbf{f}^{(t)}[s^\star] - \mathbf{f}^{(t)}(\mathbf{x}^{(t)})\right). \tag{3}$$

We say that a learner has regret $\epsilon$ if her regret against any adversary is bounded above by $\epsilon$ after $T$ time steps of learning. The *distributional swap regret* of a learner is the difference between her total reward and the reward obtained by swapping each of the learner's played actions with the best action that could have been played in its place.

**Definition 2.2.** Given sequences $\mathbf{x}^{(1:T)} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(T)})$ and $\mathbf{f}^{(1:T)} = (\mathbf{f}^{(1)}, \dots, \mathbf{f}^{(T)})$ of play by the learner and adversary, the *swap regret* corresponding to these sequences is

$$\begin{aligned}\textbf{SwapRegret}\left(\mathbf{x}^{(1:T)}, \mathbf{f}^{(1:T)}\right) &= \sup_{\pi:\mathcal{X}\to\mathcal{X}} \frac{1}{T} \sum_{i \in \mathcal{X}} \sum_{t=1}^{T} \mathbf{x}^{(t)}[i] \cdot \left(\mathbf{f}^{(t)}[\pi(i)] - \mathbf{f}^{(t)}[i]\right) \\ &= \frac{1}{T} \sum_{i \in \mathcal{X}} \sup_{j \in \mathcal{X}} \left(\sum_{t=1}^{T} \mathbf{x}^{(t)}[i] \left(\mathbf{f}^{(t)}[j] - \mathbf{f}^{(t)}[i]\right)\right) \end{aligned} \tag{4}$$

Notice that in the case that $\mathcal{X}$ is finite, we may also write the swap regret as

$$\textbf{SwapRegret}\left(\mathbf{x}^{(1:T)}, \mathbf{f}^{(1:T)}\right) = \max_{\substack{\phi:\Delta_\mathcal{X}\to\Delta_\mathcal{X} \\ \phi \text{ linear}}} \frac{1}{T} \sum_{t=1}^{T} \mathbf{f}^{(t)}\left(\phi(\mathbf{x}^{(t)})\right) - \mathbf{f}^{(t)}\left(\mathbf{x}^{(t)}\right).$$

We note that *distributional* swap-regret relates to a setting in which the learner is allowed to play a distribution over actions. We remark that, for learning with $N$ experts, [BM07] established a lower bound of $\Omega(\sqrt{NT})$ in the setting where the learner must play a concrete action at each time step (possibly in a probabilistic way). Not constraining the learner in this way, we are able to break this lower bound.

When the sequences $\mathbf{x}^{(1:T)}, \mathbf{f}^{(1:T)}$ are understood from context, we will at times abbreviate $\textbf{ExtRegret}(\mathbf{x}^{(1:T)}, \mathbf{f}^{(1:T)})$ by $\textbf{ExtRegret}(T)$ and $\textbf{SwapRegret}(\mathbf{x}^{(1:T)}, \mathbf{f}^{(1:T)})$ by $\textbf{SwapRegret}(T)$.

## 2.2 Games

An $m$-player *normal-form game* is a pair $(S, A)$ where $S = S_1 \times \cdots \times S_m$ and $A = (A_1, \cdots, A_m)$ with each $A_j : S \to \mathbb{R}$. Each $S_j$ is the set of actions (i.e., pure strategies) available to player $j$ and

each $A_j$ is the reward function, or payoff matrix, of player $j$, which maps the set of action profiles $S$ to the real numbers. Each player $j \in [m]$ selects a distribution over actions $\mathbf{x}_j \in \Delta_{S_j}$ with the goal of maximizing their reward $\mathbb{E}_{s \sim \prod_j \mathbf{x}_j}[A_j(s_1, \cdots, s_m)]$. We are interested in the following equilibrium notions:

**Definition 2.3.** An *$\epsilon$-coarse correlated equilibrium (CCE)* is a distribution $\mu \in \Delta_S$ so that, for every player $j$ and every deviation $d_j \in S_j$,

$$\mathbb{E}_{s \sim \mu}[A_j(d_j, s_{-j})] \leq \mathbb{E}_{s \sim \mu}[A_j(s_j, s_{-j})] + \epsilon. \tag{5}$$

**Definition 2.4.** An *$\epsilon$-correlated equilibrium (CE)* is a distribution $\mu \in \Delta_S$ so that, for every player $j$ and every deviation map $\pi_j : S_j \to S_j$

$$\mathbb{E}_{s \sim \mu}[A_j(\pi(s_j), s_{-j})] \leq \mathbb{E}_{s \sim \mu}[A_j(s_j, s_{-j})] + \epsilon. \tag{6}$$

Next, we define the notion of a sparse (C)CE:

**Definition 2.5** (Sparse equilibrium)**.** Let $k \in \mathbb{N}$. An $\epsilon$-CE (or $\epsilon$-CCE) $\mu \in \Delta_S$ is *$k$-sparse* if $\mu$ is a distribution over at most $k$ elements of $S$.

Typically we will consider the case that $S_j = [n]$ for each $j \in [m]$. Such a normal-form game is defined by its payoff matrices $A_j : [n]^m \to \mathbb{R}$ for $j \in [m]$.

### 2.3 Alternate models of computation

First, we consider the *query complexity* model of computation. Here, a normal-form game with payoff matrices $(A_1, \ldots, A_m)$ is fixed, but is *unknown* to the learning algorithm. The algorithm is allowed to make adaptive randomized queries to single elements of $A_1, \ldots, A_m$.

**Definition 2.6** (Query complexity)**.** Given a confidence level $\delta \in (0, 1)$, the *query complexity* of an equilibrium concept (e.g., $\epsilon$-CCE or $\epsilon$-CE) is the minimal $Q$ such that there exists an algorithm which, on any input, outputs the specified notion of equilibrium with probability at least $1 - \delta$, using only $Q$ queries.

Next, we consider the *communication complexity* model of computation. Here, each of the $m$ players $j \in [m]$ is given its own payoff matrix $A_j$, but does not know the payoff matrices of other agents. The agents are allowed to communicate in an adaptive randomized manner, according to some *communication protocol*, over an arbitrary number of rounds.

**Definition 2.7** (Communication complexity)**.** Given a confidence level $\delta \in (0, 1)$, the *communication complexity* of an equilibrium concept (e.g., $\epsilon$-CCE or $\epsilon$-CE) is the minimal $C$ such that there exists a communication protocol which, on any input, exchanges at most $C$ bits of communication between the players and terminates with each player having agreed upon the *same* distribution satisfying the equilibrium concept.

Typically, the scaling of the query and communication complexities of equilibrium concepts with respect to $\delta$ is $\log^{O(1)}(1/\delta)$; thus we often omit the parameter $\delta$ in our discussions.

**Miscellaneous notation.** We use the letter $C$ to denote absolute constants in our proofs. To avoid cluttering notation, we use the convention that the value of $C$ may change from line to line.

## 3 A new reduction from no-swap regret to no-external regret

In this section, we prove our main upper bound (Theorem 1.1, formally stated as Theorem 3.1 below), which gives a reduction from no-swap regret learning to no-external regret learning with no dependence on the number of the learner's actions. Following, we discuss several applications, including faster rates for swap regret in the experts setting with many experts (Section 3.1) and for infinite function classes (Section 3.2), upper bounds on the query and communication complexities of computing a correlated equilibrium in normal-form games (Section 3.3), and a nearly-tight bandit $\epsilon$-swap regret algorithm for constant $\epsilon$ (Section 3.5).

Let $\mathcal{X}$ be a set representing the set of actions of a learning algorithm, and let $\mathcal{F} \subset [0,1]^{\mathcal{X}}$ denote a class of utility functions for the learning algorithm, which is closed under convex combinations: for all $\mathbf{f}_1, \mathbf{f}_2 \in \mathcal{F}$ and all $\lambda \in [0,1]$, $\lambda \mathbf{f}_1 + (1-\lambda)\mathbf{f}_2 \in \mathcal{F}$. We assume that $\mathcal{F}$ admits a no-external regret learning algorithm:

**Assumption 1** (No-external regret algorithm)**.** For any $T \in \mathbb{N}$, there is an algorithm $\texttt{Alg} = \texttt{Alg}(T)$, together with functions $\texttt{Alg.act}, \texttt{Alg.update}$, which perform the following updates with an adaptive adversary over $T$ rounds. In each round $t \in [T]$:

- $\texttt{Alg}$ produces an action $\mathbf{x}^{(t)} := \texttt{Alg.act}()$, where $\mathbf{x}^{(t)} \in \Delta(\mathcal{X})$;
- The adversary observes $\mathbf{x}^{(t)}$ and chooses a function $\mathbf{f}^{(t)} \in \mathcal{F}$;
- $\texttt{Alg}$ updates its internal state according to $\texttt{Alg.update}(\mathbf{f}^{(t)})$.

We assume that the external regret of $\texttt{Alg}$ with respect to the functions $\mathbf{f}^{(t)}$ is bounded by $R_{\texttt{Alg}}(T)$:

$$\textbf{ExtRegret}\left(\mathbf{x}^{(1:T)}, \mathbf{f}^{(1:T)}\right) \le R_{\texttt{Alg}}(T).$$

**Algorithm 1** $\texttt{TreeSwap}(\mathcal{F}, \mathcal{X}, \texttt{Alg}, T, M, d)$

**Require:** Action set $\mathcal{X}$, utility class $\mathcal{F}$, no-external regret algorithm $\texttt{Alg}$, time horizon $T$, parameters $M, d$ with $T \le M^d$.

For each sequence $\sigma \in \bigcup_{h=1}^{d} \{0, 1, \dots, M-1\}^{h-1}$, initialize an instance of $\texttt{Alg}$ with time horizon $M$, denoted $\texttt{Alg}_\sigma$.
**for** $1 \le t \le T$ **do**
  Let $\sigma = (\sigma_1, \dots, \sigma_d)$ denote the base-$M$ representation of $t-1$.
  **for** $1 \le h \le d$ **do**
    **if** $\sigma_{h+1} = \cdots = \sigma_d = 0$ or $h = d$ **then**
      If $\sigma_h > 0$, call $\texttt{Alg}_{\sigma_{1:h-1}}.\texttt{update}\left(\frac{1}{M^{d-h}} \cdot \sum_{s=t-M^{d-h}}^{t-1} \mathbf{f}^{(s)}\right)$.
      $\texttt{Alg}_{\sigma_{1:h-1}}.\texttt{curAction} \leftarrow \texttt{Alg}_{\sigma_{1:h-1}}.\texttt{act}()$.
    **end if**
  **end for**
  Output the uniform mixture $\mathbf{x}^{(t)} := \frac{1}{d}\sum_{h=1}^{d} \texttt{Alg}_{\sigma_{1:h-1}}.\texttt{curAction}$, and observe $\mathbf{f}^{(t)}$.
**end for**

**Theorem 3.1.** *Suppose that an action set $\mathcal{X}$ and a utility function class $\mathcal{F}$ are given as above, together with an algorithm $\texttt{Alg}$ satisfying the conditions of Assumption 1. Suppose that $T, M, d \in$*

$\mathbb{N}$ *are given for which* $M \geq 2$ *and* $M^{d-1} \leq T \leq M^d$. *Then given an adversarial sequence* $\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(T)} \in \mathcal{F}$, `TreeSwap`$(\mathcal{F}, \mathcal{X}, \texttt{Alg}, T)$ *(Algorithm 1) produces a sequence of iterates* $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}$ *satisfying the following swap regret bound:*

$$\mathbf{SwapRegret}(\mathbf{x}^{(1:T)}, \mathbf{f}^{(1:T)}) \leq R_{\texttt{Alg}}(M) + \frac{3}{d}.$$

`TreeSwap` (Algorithm 1) simulates multiple instances of the no-external regret algorithm `Alg`. These instances of `Alg` are arranged in a depth-$d$, $M$-ary tree. For simplicity, suppose that $T = M^d$, so that there is one leaf of this tree for each time step. At each time step $t \in [T]$, consider the leaf of the tree corresponding to $t$. The root-to-leaf path for this leaf may be identified with the base-$M$ representation of $t$, which we denote by $\sigma_{1:d} = (\sigma_1, \ldots, \sigma_d) \in \{0, 1, \ldots, M-1\}^d$ (Line 3). In particular, for $h \in [d]$, $\sigma_h$ indexes the child taken at the $h$th step in this root-to-leaf path. Each node along this root-to-leaf path may therefore be identified with some prefix of $\sigma_{1:d}$, namely $\sigma_{1:h-1}$ for each $h \in [d]$.

For each $t \in [T]$, the distribution $\mathbf{x}^{(t)}$ played by `TreeSwap` (Line 10) is the uniform average over the distributions played by the instances of `Alg` (denoted by $\texttt{Alg}_{\sigma_{1:h-1}}$ for $h \in [d]$) at each node along the root-to-leaf path at step $t$. The instances $\texttt{Alg}_{\sigma_{1:h-1}}$ are updated in a lazy fashion, as follows: every $M^{d-h}$ rounds $t$ when $\sigma_{1:h-1}$ lies on the current root-to-leaf path, the utility functions $\mathbf{f}^{(t)}$ are averaged and fed to the `update` procedure of $\texttt{Alg}_{\sigma_{1:h-1}}$ (Line 6). Thus, each instance $\texttt{Alg}_{\sigma_{1:h-1}}$ is updated a total of $M$ times in the course of `TreeSwap`.

While the above discussion assumed that $T = M^d$ for simplicity, the proof below considers the setting of general values of $T$ treated by the statement of Theorem 3.1. In particular, the theorem will typically be applied in the following manner (see Sections 3.1 and 3.2): given some value of $T$, we choose $M$ as a function of $T$ and then let $d$ be chosen so as to satisfy $M^{d-1} \leq T \leq M^d$. As long as $T$ is sufficiently large, the resulting value of $3/d$ will be sufficiently small, which will ensure that $\mathbf{SwapRegret}(T)$ is small.

*Proof of Theorem 3.1.* For each $1 \leq h \leq d$, let $\Sigma_{h-1} := \{\sigma_{1:h-1} = (\sigma_1, \ldots, \sigma_{h-1}) : 1 + \sum_{g=1}^{h-1} \sigma_g \cdot M^{d-g} \leq T\}$ denote the set of sequence prefixes of length $h-1$ encountered over the course of $T$ rounds of `TreeSwap`. (Note that $\Sigma_{h-1}$ depends on $T$.) Consider any sequence $\sigma_{1:h-1} = (\sigma_1, \ldots, \sigma_{h-1}) \in \Sigma_{h-1}$ representing a node in the tree, and define $M'(\sigma_{1:h-1}) := \max\{\sigma_h \in \{0, \ldots, M-1\} : \sigma_{1:h} \in \Sigma_h\}$. Then let $\mathbf{x}^{(0)}_{\sigma_{1:h-1}}, \ldots, \mathbf{x}^{(M'(\sigma_{1:h-1}))}_{\sigma_{1:h-1}}$ denote the $M'$ actions taken by the algorithm $\texttt{Alg}_{\sigma_{1:h-1}}$ over the course of the $T$ rounds. Moreover, we let $\underline{\tau}(\sigma_{1:h-1}) := 1 + \sum_{g=1}^{h-1} \sigma_g \cdot M^{d-g}$ denote the first round when $\sigma_{1:h-1}$ is encountered, and $\overline{\tau}(\sigma_{1:h-1}) := \max\{T, \underline{\tau}(\sigma_{1:h-1}) + (M^{d-h+1} - 1)\}$ denote the last round when $\sigma_{1:h-1}$ is encountered.

We may then bound the total (unnormalized) utility of the learner over the $T$ rounds, as follows:

$$\sum_{t=1}^{T} \mathbf{f}^{(t)}(\mathbf{x}^{(t)}) = \frac{1}{d} \sum_{h=1}^{d} \sum_{\sigma_{1:h-1} \in \Sigma_{h-1}} \sum_{\sigma_h=0}^{M'(\sigma_{1:h-1})} \sum_{s=\underline{\tau}(\sigma_{1:h})}^{\overline{\tau}(\sigma_{1:h})} \mathbf{f}^{(s)}(\mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}}). \tag{7}$$

Now consider any function $\pi : \mathcal{X} \to \mathcal{X}$. We define $f \circ \pi : \mathcal{X} \to \mathbb{R}$ to be the function $(f \circ \pi)[s] = f[\pi(s)]$, for $s \in \mathcal{X}$. Then for $\mathbf{x} \in \Delta(\mathcal{X})$, we have $(f \circ \pi)(\mathbf{x}) = \mathbb{E}_{s \sim \mathbf{x}}[f[\pi(s)]]$. The learner's utility

under the swap function $\pi$ is given by

$$\begin{aligned}\sum_{t=1}^{T}(\mathbf{f}^{(t)}\circ\pi)(\mathbf{x}^{(t)}) =& \frac{1}{d}\sum_{h=1}^{d}\sum_{\sigma_{1:h-1}\in\Sigma_{h-1}}\sum_{\sigma_h=0}^{M'(\sigma_{1:h-1})}\sum_{s=\underline{\tau}(\sigma_{1:h})}^{\overline{\tau}(\sigma_{1:h})}(\mathbf{f}^{(s)}\circ\pi)(\mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}})) \\ \le& \frac{1}{d}\sum_{h=1}^{d}\sum_{\sigma_{1:h}\in\Sigma_h}\max_{\mathbf{x}^\star\in\mathcal{X}}\sum_{s\in\underline{\tau}(\sigma_{1:h})}^{\overline{\tau}(\sigma_{1:h})}\mathbf{f}^{(s)}(\mathbf{x}^\star). \end{aligned} \tag{8}$$

Subtracting (7) from (8) and using the fact that $\mathbf{f}(\mathbf{x})\in[0,1]$ for all $\mathbf{f},\mathbf{x}$, we see that

$$\begin{aligned}&\sum_{t=1}^{T}(\mathbf{f}^{(t)}\circ\pi)(\mathbf{x}^{(t)}) - \mathbf{f}^{(t)}(\mathbf{x}^{(t)}) \\ \le& \frac{1}{d}\sum_{h=2}^{d-1}\sum_{\sigma_{1:h-1}\in\Sigma_{h-1}}\max_{\mathbf{x}^\star\in\mathcal{X}}\left(\sum_{s\in\underline{\tau}(\sigma_{1:h-1})}^{\overline{\tau}(\sigma_{1:h-1})}\mathbf{f}^{(s)}(\mathbf{x}^\star) - \sum_{\sigma_h=0}^{M'(\sigma_{1:h-1})}\sum_{s=\underline{\tau}(\sigma_{1:h})}^{\overline{\tau}(\sigma_{1:h})}\mathbf{f}^{(s)}(\mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}})\right) + \frac{1}{d}\sum_{s=1}^{T}\max_{\mathbf{x}^\star\in\mathcal{X}}\mathbf{f}^{(s)}(\mathbf{x}^{(\star)}) \\ \le& \frac{1}{d}\sum_{h=2}^{d-1}\left(M^{d-h+1} + \sum_{\substack{\sigma_{1:h-1}\in\Sigma_{h-1}:\\ \overline{\tau}(\sigma_{1:h-1})-\underline{\tau}(\sigma_{1:h-1})=M^{d-h+1}-1}} M\cdot M^{d-h}\cdot R_{\mathtt{Alg}_{\sigma_{1:h-1}}}(M)\right) + \frac{1}{d}\sum_{s=1}^{T}\max_{\mathbf{x}^\star\in\mathcal{X}}\mathbf{f}^{(s)}(\mathbf{x}^{(s)}) \\ \le& T\cdot R_{\mathtt{Alg}}(M) + \frac{1}{d}\sum_{h=2}^{d-1}M^{d-h+1} + \frac{T}{d} \le T\cdot R_{\mathtt{Alg}}(M) + \frac{3T}{d}, \end{aligned} \tag{9}$$

where the second inequality above uses the external regret assumption of $\mathtt{Alg}_{\sigma_{1:h-1}}$ for each possible choice of $\sigma_{1:h-1}$ (Assumption 1), as well as the following observation: for each $2\le h\le d-1$, there is at most a single sequence $\sigma_{1:h-1}\in\Sigma_{h-1}$ for which $\overline{\tau}(\sigma_{1:h-1})-\underline{\tau}(\sigma_{1:h-1}) < M^{d-h+1}-1$. For such a sequence $\sigma_{1:h-1}$, we may trivially upper bound

$$\max_{\mathbf{x}^\star\in\mathcal{X}}\left(\sum_{s\in\underline{\tau}(\sigma_{1:h-1})}^{\overline{\tau}(\sigma_{1:h-1})}\mathbf{f}^{(s)}(\mathbf{x}^\star) - \sum_{\sigma_h=0}^{M'(\sigma_{1:h-1})}\sum_{s=\underline{\tau}(\sigma_{1:h})}^{\overline{\tau}(\sigma_{1:h})}\mathbf{f}^{(s)}(\mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}})\right) \le M^{d-h+1}.$$

The final inequality in the display (9) uses the fact that, since $M\ge 2$ and $T\ge M^{d-1}$ by assumption, $\sum_{h=2}^{d-1}M^{d-h+1}\le 2M^{d-1}\le 2T$.

Dividing the display (9) by $T$ gives the desired regret bound. □

### 3.1 Application: swap regret for $N$ experts

First, we show how Theorem 3.1 can be used to bound the swap regret in the "finite experts" setting: the number of rounds required to obtain sublinear swap regret is only polylogarithmic in the number of experts $N$, which improves upon the bound of $\Theta(N\log N)$ rounds from [BM07; SL05].

**Corollary 3.2.** *Fix $N\in\mathbb{N}$. Then, letting $\mathcal{X}=[N]$ and $\mathcal{F}=[0,1]^{[N]}$, for any $\epsilon\in(0,1)$, there is an algorithm for which the swap regret is bounded above as $\mathbf{SwapRegret}(T)\le\epsilon$ for any $T$ which satisfies $T\ge(\log(N)/\epsilon^2)^{O(1/\epsilon)}$.*

*Each iteration takes at most $O(N/\epsilon)$ time, and the amortized runtime over the $T$ iterations is $O(NT)$. The total space complexity is $O(N/\epsilon)$.*

*Proof.* Given $\epsilon \in (0,1)$ and $T$ as in the statement of the corollary, we may choose $M = \lceil \log(N)/\epsilon^2 \rceil$, and $d \geq \lceil 1/\epsilon \rceil$, so that $M^{d-1} \leq T \leq M^d$.

We then call $\texttt{TreeSwap}(\mathcal{F}, \mathcal{X}, \texttt{Alg}, T, M, d)$, where $\texttt{Alg}$ is the multiplicative weights algorithm (denoted $\texttt{MWU}$), which obtains a regret bound of $R_{\texttt{Alg}}(M) \leq C\sqrt{\log(N)/M}$, for some constant $C$. Then Theorem 3.1 guarantees that

$$\textbf{SwapRegret}(T) \leq O\left(\sqrt{\frac{\log N}{\log(N)/\epsilon^2}} + \frac{1}{d}\right) \leq O(\epsilon).$$

To get the desired bound of $\epsilon$, we can scale $\epsilon$ down by a constant factor.

To bound the time complexity, we need to consider two types of operations:

- *Maintaining the cumulative reward vector:* at any iteration $t \in [T]$ the algorithm would maintain the sum $\sum_{s=1}^{t} \mathbf{f}^{(s)}$. Since each $\mathbf{f}^{(s)}$ is an $N$-dimensional vector, this takes time $O(N)$ per iteration and $O(NT)$ total.
- *Number of MWU updates:* altogether, $\texttt{TreeSwap}$ has $O(T/M)$ instances of $\texttt{Alg} = \texttt{MWU}$, each making $M$ updates, so there are $O(T)$ updates to all instances of $\texttt{MWU}$ in total. We will show that each update takes time at most $O(N)$ below. To do so, note that an update consists of the following operations:
  - *Computing the average reward vector that is fed into the* $\texttt{MWU}$ *instance (Line 6):* In order to update each $\texttt{MWU}$ instance $\texttt{MWU}_{\sigma_{1:h-1}}$ in $\texttt{TreeSwap}$, we have to average the reward vectors over the $M^{d-h}$ preceding iterations. We say that an instance $\texttt{MWU}_{\sigma_{1:h-1}}$ is *live at round* $t$ if the base-$M$ representation of $t-1$ beings with $\sigma_{1:h-1}$ (i.e., if $\sigma_{1:h-1}$ is on the root-to-leaf path corresponding to the current iteration $t$). Note that at each round $t$, there are $d$ live instances of $\texttt{MWU}$. Since we maintain the current cumulative reward vector at each iteration, it suffices to save in memory, for each *live* execution of the MWU, the cumulative loss up to the time of its last update. Fix any such live instance, and let $t_0$ be the time of its last update: then for this instance, we store $\sum_{s=1}^{t_0} \mathbf{f}^{(s)}$. Then, the cumulative loss since the last update is just $\sum_{s=1}^{t} \mathbf{f}^{s} - \sum_{s=1}^{t_0} \mathbf{f}^{s}$. Since each of these summands is stored in memory as an $N$-dimensional vector, computing the difference takes $O(N)$ time.
  - *Computing the next action to take in a single* $\texttt{MWU.update}$ *call:* Each update step of $\texttt{MWU}$ takes time $O(N)$.
  - *Computing the uniform mixture over the actions suggested by the* $d$ *live* $\texttt{MWU}$ *instances at each round (Line 10):* To compute this change, we account for the change to each of the live $\texttt{MWU}$ instances which is updated for each iteration $t$: for each such update, we need to perform a $O(N)$-time operation.

  There are at most $d = O(1/\epsilon)$ updates in each iteration, which implies a worst-case runtime of $O(Nd) = O(N/\epsilon)$ and a total runtime of $O(NT)$, since there are at most $O(T)$ updates in total, as argued above.

This sums up to an $O(NT)$ runtime total and a maximum of $O(N/\epsilon)$ per-iteration runtime. To bound the space complexity, notice that, for each live execution of an $\texttt{MWU}$ instance, we need to store the cumulative loss since its last update. There are $d$ such instances, one for each level. Each

cumulative loss is an $N$-dimensional vector, which yields a total space of $O(Nd)$. Additionally, we need to store the current distribution over actions for each `MWU` instances. This takes space $O(N)$ per `MWU` instance and $O(Nd) = O(N/\epsilon)$ space overall. □

### 3.2 Application: Swap regret and approximate CE for infinite games

In abstract function classes, the existence of no-external regret learners depend on various *dimensions* of the class (see Appendix B). In general, for a bounded real-valued function class $\mathcal{H}$, there exist online learners with regret approaching 0 as $T \to \infty$ if (and only if) its *sequential Rademacher complexity* $\mathfrak{R}_T(\mathcal{H})$ converges to 0 as $T \to \infty$ (Theorem B.2). In the special case that $\mathcal{H}$ is binary-valued, the condition that $\mathfrak{R}_T(\mathcal{H}) \to 0$ is equivalent to finiteness of the *Littlestone dimension*, $\mathrm{LDim}(\mathcal{H})$, of $\mathcal{H}$. In the real-valued setting, $\mathfrak{R}_T(\mathcal{H}) \to 0$ if and only if the *sequential $\delta$-fat shattering dimension* of $\mathcal{H}$, $\mathrm{SFat}(\mathcal{H}, \delta)$, is finite for all $\delta > 0$ (see Proposition B.5).

Combining Theorem 3.1 and Theorem B.2, we can show the existence of a learning algorithm whose swap regret converges to 0 as long as the learner's action set has vanishing sequential Rademacher complexity. Since we denote the learner's action space by $\mathcal{X}$ and the space of reward functions by $\mathcal{F} \subset [0,1]^{\mathcal{X}}$, the relevant function class is the class $\{f \mapsto f(x) \ : \ x \in \mathcal{X}\}$; we denote this class by $\mathcal{X}$, with slight abuse of notation.

**Corollary 3.3.** *There is a constant $C > 0$ so that the following holds. Suppose that $\mathcal{X}, \mathcal{F}$ are given as above, and let $\epsilon \in (0,1)$ be given. Define $\tau_{\mathcal{X}}(\epsilon) := \inf_{\tau \in \mathbb{N}}\{\tau \ : \ \mathfrak{R}_\tau(\mathcal{X}) \le \epsilon\}$. Then for any $T \ge (\tau_{\epsilon/C}(\mathcal{X}))^{C/\epsilon}$ there is an algorithm for which the swap regret is bounded as* $\mathbf{SwapRegret}(T) \le \epsilon$.

*Proof.* We use Theorem 3.1 with action set given by $\mathcal{X}$ and the function class given by the convex hull of $\mathcal{F}$, denoted $\mathrm{co}(\mathcal{F})$. Accordingly, define $\bar{\mathcal{X}} \subset [0,1]^{\mathrm{co}(\mathcal{F})}$ by $\bar{\mathcal{X}} := \{\bar{f} \mapsto \bar{f}(x) \ : \ x \in \mathcal{X}\}$, where the domain for $\bar{\mathcal{X}}$ is $\mathrm{co}(\mathcal{F})$.

Theorem B.2 gives that Assumption 1 is satisfied by some external-regret algorithm `Alg` with $R_{\texttt{Alg}}(T) \le 2\mathfrak{R}_T(\bar{\mathcal{X}})$. As a straightforward consequence of Jensen's inequality and Definition B.1, we have that $\mathfrak{R}_T(\bar{\mathcal{X}}) \le \mathfrak{R}_T(\mathcal{X})$ (in fact, equality holds). Thus, applyying Theorem 3.1 with $M = \tau_{\mathcal{X}}(\epsilon/C)$ a choice of $d$ satisfying $M^{d-1} \le T \le M^d$, we obtain that, as long as $C$ is sufficiently large, $\texttt{TreeSwap}(\mathrm{co}(\mathcal{F}), \mathcal{X}, \texttt{Alg}, T)$ obtains $\mathbf{SwapRegret}(T) \le \epsilon$. □

By Proposition B.5, we obtain the following immediate corollary.

**Corollary 3.4.** *Consider $\mathcal{X}, \mathcal{F}$ as above, and suppose $\epsilon \in (0,1)$ is given. Then we have the following:*

- *If $\mathcal{F}$ is binary-valued and $\mathrm{LDim}(\mathcal{X}) = L$, then for all $T$ satisfying $T \ge (L/\epsilon)^{O(1/\epsilon)}$, there is an algorithm certifying* $\mathbf{SwapRegret}(T) \le \epsilon$.
- *If $\mathrm{SFat}(\mathcal{X}, \delta) \le O(\delta^{-p})$ for some $p \in [0,2)$, then letting $I = \int_0^1 \sqrt{\mathrm{SFat}(\mathcal{X}, \delta)}d\delta$, for $T \ge (I/\epsilon)^{O(1/\epsilon)}$, there is an algorithm satisfying* $\mathbf{SwapRegret}(T) \le \epsilon$.

*Remark* 3.5. Recall that our algorithm plays a *distribution* over actions in each round. In infinite action-spaces, distributions can be infinitely-supported. Yet, for the classes considered in Corollary 3.4, *uniform convergence* holds. This means that any infinitely-supported distribution can be replaced by a uniform distribution over a sufficiently-large i.i.d. sample.

Next, we utilize the fact that if each player uses a learning algorithm with swap regret bounded by $\epsilon$, then the empirical average of their joint strategy profiles is an $\epsilon$-approximate CE. Since there is a learning algorithm with respect to $\mathcal{X}, \mathcal{F}$ as above with swap regret converging to 0 as long as $\mathfrak{R}_T(\mathcal{X}) \to 0$ as $T \to \infty$ by Corollary 3.3 (which, in turn, by Proposition B.5 holds if $\mathrm{SFat}(\mathcal{X}, \delta) < \infty$ for all $\delta > 0$), we obtain the following as a further corollary:

**Corollary 3.6.** *Let $(S, A)$ be an $m$-player game. Suppose either of the below conditions hold:*

- $\mathfrak{R}_T(S, A) \to 0$ *as* $T \to \infty$ *(which, in turn, holds if* $\mathrm{SFat}(S, A, \delta) < \infty$ *for all* $\delta > 0$*);*
- $(S, A)$ *is binary-valued and* $\mathrm{LDim}(S, A) < \infty$.

*Then $(S, A)$ has an $\epsilon$-CE for all $\epsilon > 0$.*

### 3.3 Application: query and communication complexities of computing correlated equilibria

**Algorithm 2** $\texttt{CommCE}(A_1, \dots, A_m, \epsilon, \delta)$

**Require:** $m$-player, $N$-action normal-form game specified by payoff matrices $A_1, \dots, A_m : [N]^m \to [0, 1]$, real numbers $\epsilon, \delta \in (0, 1)$.

1: Set $L \leftarrow \frac{Cm \log(TNm/\delta)}{\epsilon^2}$, $M = \lceil \log(N)/\epsilon^2 \rceil$, $d = \lceil 1/\epsilon \rceil$, and $T = M^d$, for a sufficiently large constant $C$.
2: Let $\texttt{MWUSamp}$ denote the following variant of the multiplicative weights algorithm $\texttt{MWU}$ on action set $[N]$: it is identical to $\texttt{MWU}$, except when $\texttt{MWU}$ would play $\mathbf{y}^{(t)} \in \Delta_N$, $\texttt{MWUSamp}$ samples $a_1, \dots, a_N \sim \mathbf{y}^{(t)}$ and plays $\hat{\mathbf{y}}^{(t)} := \frac{1}{N} \sum_{j=1}^N e_{a_j} \in \Delta_N$.
3: Each player $i$ initializes an instance $\texttt{TreeSwap}_i$ of $\texttt{TreeSwap}(\mathcal{F}_i, \mathcal{X}, \texttt{MWUSamp}, T, M, d)$, where $\mathcal{X} = [N]$, $\mathcal{F}_i := \{a_i \mapsto \mathbb{E}_{a_{-i} \sim \mathbf{p}}[A_i(a_i, a_{-i})] \ : \ \mathbf{p} \in \Delta_{[N]^{m-1}}\}$.
4: **for** $1 \le t \le T$ **do**
5:  **for** Player $1 \le i \le m$ **do**
6:  Player $i$ receives a distribution over actions at round $t$, $\mathbf{x}_i^{(t)} \in \Delta_{\mathcal{X}}$, from $\texttt{TreeSwap}_i$.
7:  Send $\mathbf{x}_i^{(t)}$ to all other players.
8:  **end for**
9:  For $i \in [m]$, define $\mathbf{u}_i^{(t)} \in [0, 1]^N$ by, for $a \in [N]$, $\mathbf{u}_i^{(t)}[a] \leftarrow \mathbb{E}_{a_{i'} \sim \mathbf{x}_{i'}^{(t)} \ \forall i' \neq i}[A_i(a, a_{-i})]$.
10:  For $i \in [m]$, update $\texttt{TreeSwap}_i$ with the function $\mathbf{x} \mapsto \langle \mathbf{x}, \mathbf{u}_i^{(t)} \rangle$.
11: **end for**
12: **return** the distribution $\frac{1}{T} \sum_{t=1}^T (\mathbf{x}_1^{(t)} \times \cdots \times \mathbf{x}_m^{(t)}) \in \Delta_{[N]^m}$.

In this section, we discuss corollaries of Theorem 3.1 for communication and query-efficient protocols for computing correlated equilibria in normal-form games. We begin with communication complexity (Definition 2.7): we show that for $\epsilon = O(1)$ and $m = O(1)$, the communication complexity of computing an $\epsilon$-CE in a $m$-player normal-form game is $\mathrm{poly} \log N$, where $N$ denotes the number of actions per player. In contrast, the best known prior bound was $\tilde{O}(N)$, using the sublinear swap regret algorithm of [BM07] (see also [Bab20]).

**Corollary 3.7** (Communication complexity of CE)**.** *Let $N, m \in \mathbb{N}$ be fixed and suppose $A_1, \dots, A_m : [N]^m \to [0, 1]$ are the payoff matrices of an $m$-player $N$-action normal-form game $G$. Then for any*

$\epsilon, \delta \in (0,1)$, *there is an algorithm which outputs an* $\epsilon$*-CE of* $G$ *with probability* $1-\delta$*, for which the communication complexity is* $(\log(N)/\epsilon)^{O(1/\epsilon)} \cdot O(m^2 \log(m/\delta))$.

*Proof.* We show that the algorithm $\texttt{CommCE}(A_1, \ldots, A_m, \epsilon, \delta)$ (Algorithm 2) has the desired properties. Algorithm 2 works by having each player $i \in [m]$ run an instance $\texttt{TreeSwap}_i$ of $\texttt{TreeSwap}$ (Algorithm 1). Each instance $\texttt{TreeSwap}_i$ is used with action set equal to $\mathcal{X} = \Delta_N$, and with function class $\mathcal{F}_i = \left\{a_i \mapsto \mathbb{E}_{a_{-i} \sim \mathbf{p}}[A_i(a_i, a_{-i})] \ : \ \mathbf{p} \in \Delta_{[N]^{m-1}}\right\}$. Note that $\mathcal{F}_i$ is a convex hull of at most $N^{m-1}$ vectors. Moreover, the external regret algorithm used in the $\texttt{TreeSwap}_i$ instance is a variant of multiplicative weights which samples $L = O\left(\frac{m \log(TNm/\delta)}{\epsilon^2}\right)$ actions from the distribution played by $\texttt{MWU}$ at each round (see Lines 1 and 2 of $\texttt{CommCE}$). As in Algorithm 2, we let the distribution played by agent $i$'s instance $\texttt{TreeSwap}_i$ be denoted by $\mathbf{x}_i^{(t)} \in \Delta_{\mathcal{X}}$.

Fix $\epsilon, \delta \in (0,1)$. As in Algorithm 2 we define, for each $i \in [m]$ and $t \in [T]$, $\mathbf{u}_i^{(t)} \in [0,1]^N$ by $\mathbf{u}_i^{(t)}[a] := \mathbb{E}_{a_{i'} \sim \mathbf{x}_{i'}^{(t)} \ \forall i' \neq i}[A_i(a, a_{-i})]$ for $a \in [N]$. We choose $M = \lceil \log(N)/\epsilon^2 \rceil$, $d = \lceil 1/\epsilon \rceil$, and $T = M^d$. Recall that each algorithm $\texttt{TreeSwap}_i$ used in $\texttt{CommCE}$ is an instance of $\texttt{TreeSwap}(\mathcal{F}_i, \mathcal{X}, \texttt{MWUSamp}, T, M, d)$.

**Step 1: correctness.** First, we show that the instances of $\texttt{MWUSamp}(M)$, when used in the context of $\texttt{TreeSwap}_i$ for each player $i$, satisfy Assumption 1 with high probability. Note that each instance of $\texttt{MWUSamp}(M)$ used in player $i$'s instance of $\texttt{TreeSwap}$ is applied with action set $\mathcal{X} = [N]$ and function class $\mathcal{F}_i = \{a_i \mapsto \mathbb{E}_{a_{-i} \sim \mathbf{p}}[A_i(a_i, a_{-i})] \ : \ \mathbf{p} \in \Delta_{[N]^{m-1}}\}$. Consider the execution of $\texttt{MWUSamp}(M)$ with $\mathcal{X}, \mathcal{F}_i$ against an adaptive adversary: let $\hat{\mathbf{y}}^{(1)}, \ldots, \hat{\mathbf{y}}^{(M)} \in \Delta_N$ denote the actions of $\texttt{MWUSamp}(M)$, $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(M)}$ denote the induced utility vectors of the adversary (namely, if the adversary plays $\mathbf{p}^{(t)} \in \Delta_{[N]^{m-1}}$, then the induced utility vector is $\mathbf{u}^{(t)} = \mathbb{E}_{a_{-i} \sim \mathbf{p}^{(t)}}[A_i(\cdot, a_{-i})]$, and let $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(M)} \in \Delta_N$ denote the actions of the multiplicative weights algorithm from which $\hat{\mathbf{y}}^{(t)}$ are sampled. The guarantee of multiplicative weights ensures that $\max_{\mathbf{y}^\star \in \Delta_N} \sum_{t=1}^M \left(\langle \mathbf{y}^\star, \mathbf{u}^{(t)} \rangle - \langle \mathbf{y}^{(t)}, \mathbf{u}^{(t)} \rangle\right) \leq O(\sqrt{M \log N}) \leq O(\epsilon \cdot M)$ (with probability 1). Hoeffding's inequality together with the choice of $L$ in Line 1 of Algorithm 2 (as long as $C$ is sufficiently large) and a union bound over $t \in [M], a_{-i} \in [N]^{m-1}$ ensures that with probability at least $1 - \delta/(mT)$,

$$\max_{t \in [M]} \max_{a_{-i} \in [N]^{m-1}} \left| \langle \hat{\mathbf{y}}^{(t)} - \mathbf{y}^{(t)}, A_i(\cdot, a_{-i}) \rangle \right| \leq \epsilon.$$

Thus, under this event occurring with probability $1 - \delta/(mT)$, the external regret of the instance of $\texttt{MWUSamp}(M)$ may be bounded above as follows:

$$\max_{\mathbf{y}^\star \in \Delta_N} \frac{1}{M} \sum_{t=1}^M \left(\langle \mathbf{y}^\star, \mathbf{u}^{(t)} \rangle - \langle \mathbf{y}^{(t)}, \mathbf{u}^{(t)} \rangle\right) \leq O(\epsilon) + \frac{1}{M} \sum_{t=1}^M \left| \langle \hat{\mathbf{y}}^{(t)} - \mathbf{y}^{(t)}, \mathbf{u}^{(t)} \rangle \right| \leq O(\epsilon).$$

Note that the total number of instances of $\texttt{MWUSamp}(M)$ used in $\texttt{TreeSwap}_i$ is $1 + M + \cdots + M^{d-1} \leq M^d = T$. By a union bound over all these instances, under some event $\mathcal{E}_i$ occurring with probability $1 - \delta/m$, the external regret of all instances may be bounded above by: $R_{\texttt{MWUSamp}}(M) \leq O(\epsilon)$. Thus, Theorem 3.1 yields that, under $\mathcal{E}_i$, the swap regret of $\texttt{TreeSwap}_i$ may be bounded above as follows:

$$\max_{\pi : [N] \to [N]} \frac{1}{T} \sum_{t=1}^T \sum_{a=1}^N \mathbf{x}_i^{(t)}[a] \cdot (\mathbf{u}_i^{(t)}[\pi(a)] - \mathbf{u}_i^{(t)}[a]) \leq O(\epsilon).$$

Under the event $\mathcal{E}_1 \cap \cdots \cap \mathcal{E}_m$ (which occurs with probability $1-\delta$), it follows that, for each $i \in [m]$,

$$\max_{\pi:[N]\to[N]} \frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}_{a_{i'}\sim \mathbf{x}_{i'}^{(t)}\ \forall i'\in[m]}(A_i(\pi(a_i), a_{-i}) - A_i(a_1,\ldots,a_m))\right)$$
$$= \max_{\pi:[N]\to[N]} \frac{1}{T}\sum_{t=1}^{T}\sum_{a=1}^{N} \mathbf{x}_i^{(t)}[a]\cdot\left(\mathbf{u}_i^{(t)}[\pi(a)] - \mathbf{u}_i^{(t)}[a]\right) \le O(\epsilon),$$

which implies that $\frac{1}{T}\sum_{t=1}^{T}(\mathbf{x}_1^{(t)} \times \cdots \times \mathbf{x}_m^{(t)})$ is an $O(\epsilon)$-CE of $G$. The claimed result follows by rescaling $\epsilon$ by a constant factor.

**Step 2: communication cost.** Note that the distributions output by each instance of `MWUSamp` are $L$-sparse, and that the distributions $\mathbf{p}_i^{(t)}$ chosen by each instance $\texttt{TreeSwap}_i$ are mixtures of $d = O(1/\epsilon)$ outputs of instances of `MWUSamp` (see Line 10 of Algorithm 1). Thus, for each $i \in [m], t \in [T]$, $\mathbf{x}_i^{(t)}$ is $O(L/\epsilon)$-sparse, and thus can be communicated with $O(L\log(N/\epsilon)/\epsilon)$ bits.[9] Hence the total communication cost is $O(TmL\log(N/\epsilon)/\epsilon)$, which may be bounded as follows:

$$TmL\log(N/\epsilon)/\epsilon \le O\left(\frac{m^2\log(N/\epsilon)\log(TNm/\delta)}{\epsilon^3}\cdot\left(\frac{\log N}{\epsilon^2}\right)^{C/\epsilon}\right) \le O(m^2\log(m/\delta))\cdot\left(\frac{\log N}{\epsilon}\right)^{O(1/\epsilon)}.$$

□

As a corollary of Corollary 3.7, we obtain that there is a computationally efficient algorithm to compute a $\operatorname{poly}\log N$-sparse $\epsilon$-CE in $O(1)$-player normal-form games. Prior to the present work, the smallest sparsity obtained by any efficient algorithm was $\tilde{O}(N)$ [BBP14].

**Corollary 3.8** (Efficient computation of sparse CE)**.** *For any $N$-action $m$-player game and $\epsilon, \delta \in (0,1)$, there is an algorithm which computes a $\left(\frac{m\log(N/\delta)}{\epsilon}\right)^{O(m+\frac{1}{\epsilon})}$-sparse CE in time $\operatorname{poly}(N^m, (\log(N)/\epsilon)^{O(1/\epsilon)})$ with probability at least $1-\delta$.*

*Proof.* We simply run the algorithm `CommCE`, which clearly runs in the claimed time. The sparsity of the returned solution is

$$T\cdot(L/\epsilon)^m \le O\left(\left(\frac{Cm\log(TNm/\delta)}{\epsilon^2}\right)^m\cdot\left(\frac{\log(N)}{\epsilon^2}\right)^{C/\epsilon}\right) \le \left(\frac{m\log(N/\delta)}{\epsilon}\right)^{O(m+\frac{1}{\epsilon})}.$$

□

Next, we proceed to our corollary for query complexity (Definition 2.6): we show that for $\epsilon = O(1)$ and $m = O(1)$, the query complexity of computing an $\epsilon$-CE in an $m$-player normal-form game is $\operatorname{poly}\log N$; the best known prior bound was $\tilde{O}(N^2)$, using the sublinear swap regret algorithm of [BM07] (see also [Bab20]).

[9] We may assume that only $O(\log(1/\epsilon))$ bits of each nonzero entry of $\mathbf{x}_i^{(t)}$ are communicated; the loss from not communicating the lower order bits is bounded by $\epsilon$.

**Corollary 3.9** (Query complexity of CE)**.** *Let $N, m \in \mathbb{N}$ be fixed and suppose $A_1, \dots, A_m : [N]^m \to [0,1]$ are the payoff matrices of an $m$-player $N$-action normal-form game $G$. Then for any $\epsilon, \delta \in (0,1)$, there is an algorithm which outputs an $\epsilon$-CE of $G$ with probability $1-\delta$, which queries at most $(\log(N)/\epsilon)^{O(1/\epsilon)} \cdot O(mN \log(m/\delta))$ entries of the payoff matrices $A_i$.*

One approach to proving Corollary 3.9 is to use a minor modification of the `CommCE` algorithm; however, we can save a factor of $m$ in the query complexity by using a slightly different approach.

*Proof of Corollary 3.9.* Fix $\epsilon, \delta \in (0,1)$. We show that the algorithm $\texttt{QueryCE}(A_1, \dots, A_m, \epsilon, \delta)$ (Algorithm 3) has the desired properties. The algorithm `QueryCE` operates by having each player $i \in [m]$ run an instance of `TreeSwap`, denoted by $\texttt{TreeSwap}_i$. Each instance $\texttt{TreeSwap}_i$ is used with action set $\mathcal{X} = \Delta_N$, with function class $\mathcal{F} = [0,1]^{[N]} = \{a \mapsto \mathbf{u}[a] \ : \ \mathbf{u} \in [0,1]^N\}$, and with the external regret algorithm set to multiplicative weights, `MWU`. Let $\mathbf{x}_i^{(t)} \in \Delta_N$ denote the distribution played by each $\texttt{TreeSwap}_i$ instance at round $t$.

Define, for each $i \in [m]$ and $t \in [T]$, $\mathbf{u}_i^{(t)} \in [0,1]^N$ by $\mathbf{u}_i^{(t)}[a] := \mathbb{E}_{a_{i'} \sim \mathbf{x}_{i'}^{(t)} \ \forall i' \neq i}[A_i(a, a_{-i})]$ for $a \in [N]$. $\mathbf{u}_i^{(t)}$ represents the "ideal" utility vector that would be passed to each $\texttt{TreeSwap}_i$ instance at round $t$. Computing $\mathbf{u}_i^{(t)}$ exactly would take too many queries to the payoffs $A_i$, so instead `QueryCE` computes an approximation of them: in particular, it samples $L = O(\frac{\log(TNm/\delta)}{\epsilon^2})$ actions from each $\mathbf{x}_i^{(t)}$ and uses these samples to compute an empirical estimate $\hat{\mathbf{u}}_i^{(t)}$ of $\mathbf{u}_i^{(t)}$ (Line 9).

Hoeffding's inequality together with the choice of $L$ in Line 1 of Algorithm 3 (as long as $C$ is chosen sufficiently large) and a union bound over $t \in [T], i \in [m]$, and the $N$ coordinates of each utility vector, ensure that, under some event $\mathcal{E}$ occurring with probability $1-\delta$ over the execution of `CommCE`, for all $i \in [m]$ and $t \in [T]$, it holds that $\|\mathbf{u}_i^{(t)} - \hat{\mathbf{u}}_i^{(t)}\|_\infty \le \epsilon/4$. The guarantee for $\texttt{TreeSwap}_i$ (Theorem 3.1, which with the settings of the parameters used in `QueryCE` becomes exactly Corollary 3.2) together with the choice of $T, M, d$ in Line 1 ensures that, for each $i \in [m]$,

$$\max_{\pi:[N]\to[N]} \frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{N} \mathbf{x}_i^{(t)}[j] \cdot \left( \hat{\mathbf{u}}_i^{(t)}[\pi(j)] - \hat{\mathbf{u}}_i^{(t)}[j] \right) \le \epsilon/2.$$

Under the event $\mathcal{E}$, it then follows that, for each $i \in [m]$,

$$\begin{aligned} &\max_{\pi:[N]\to[N]} \frac{1}{T} \sum_{t=1}^{T} \left( \mathbb{E}_{a_{i'} \sim \mathbf{x}_{i'}^{(t)} \ \forall i' \in [m]} (A_i(\pi(a_i), a_{-i}) - A_i(a_1, \dots, a_m)) \right) \\ = &\max_{\pi:[N]\to[N]} \frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{N} \mathbf{x}_i^{(t)}[j] \cdot \left( \mathbf{u}_i^{(t)}[\pi(j)] - \mathbf{u}_i^{(t)}[j] \right) \le \epsilon, \end{aligned}$$

which implies that $\frac{1}{T} \sum_{t=1}^{T} (\mathbf{x}_1^{(t)} \times \cdots \times \mathbf{x}_m^{(t)})$ is an $\epsilon$-CE of $G$.

The total number of queries made in the course of `QueryCE` (in Line 7) is

$$NL \cdot Tm \le O\left( \frac{Nm \log(TNm/\delta)}{\epsilon^2} \cdot \left( \frac{\log N}{\epsilon^2} \right)^{C/\epsilon} \right) \le O(Nm \log(m/\delta)) \cdot \left( \frac{\log N}{\epsilon} \right)^{O(1/\epsilon)}.$$

□

**Algorithm 3** $\texttt{QueryCE}(A_1, \dots, A_m, \epsilon, \delta)$

**Require:** $m$-player, $N$-action normal-form game specified by payoff matrices $A_1, \dots, A_m : [N]^m \to [0,1]$, real numbers $\epsilon, \delta \in (0,1)$.

Set $M \leftarrow \frac{C \log N}{\epsilon^2}$, $d \leftarrow \lceil C/\epsilon \rceil$, $T \leftarrow M^d$, and $L \leftarrow \frac{C \log(TNm/\delta)}{\epsilon^2}$, for a sufficiently large constant $C$.
Each player $i$ initializes an instance $\texttt{TreeSwap}_i$ of $\texttt{TreeSwap}(\mathcal{F}, \mathcal{X}, \texttt{MWU}, T, M, d)$, where $\mathcal{X} = \Delta_N$, $\mathcal{F} := \{a \mapsto \mathbf{u}[a] \ : \ \mathbf{u} \in [0,1]^N\}$, and $\texttt{MWU}$ denotes the multiplicative weights algorithm.
**for** $1 \le t \le T$ **do**
    **for** Player $1 \le i \le m$ **do**
        Player $i$ receives a distribution over actions at round $t$, $\mathbf{x}_i^{(t)} \in \Delta_{\mathcal{X}}$, from $\texttt{TreeSwap}_i$.
        Sample $L$ actions from $\mathbf{x}_i^{(t)}$, denoted $a_{i,1}, \dots, a_{i,L}$.
        Define $\hat{\mathbf{u}}_i^{(t)} \in [0,1]^N$ by, $\hat{\mathbf{u}}_i^{(t)}[j] \leftarrow \frac{1}{N} \sum_{\ell=1}^{L} A_i(j, (a_{i',\ell})_{i' \neq i})$ by making $NL$ queries to $A_i$.
        Update $\texttt{TreeSwap}_i$ with the function $a_i \mapsto \hat{\mathbf{u}}_i^{(t)}[a_i]$.
    **end for**
**end for**
**return** the distribution $\frac{1}{T} \sum_{t=1}^{T} (\mathbf{x}_1^{(t)} \times \cdots \times \mathbf{x}_m^{(t)}) \in \Delta_{[N]^m}$.

### 3.4 Application: extensive-form games

In this section, we use Theorem 3.1 to efficiently compute $\epsilon$-CE in extensive-form games in polynomial time when $\epsilon = O(1)$. We begin by briefly introducing the terminology and notation for extensive-form games. An *$m$-player extensive-form game (EFG)* $G$ is specified by a tree $\mathcal{T}$, where the children of each node $h$ of $\mathcal{T}$ are indexed by a set of *actions* $A(h)$ at the node $h$. To each non-leaf node $h$ of $\mathcal{T}$ is associated some player in $[m] \cup \{\mathsf{ch}\}$, where $\mathsf{ch}$ denotes the *chance player*, which plays actions at each of its nodes according to some fixed probabilities. For each $i \in [m]$, player $i$'s nodes are partitioned into *information sets*: nodes in the same information set cannot be distinguished by player $i$. We assume *perfect recall*, which means that each player does not forget any information (i.e., distinct information sets cannot have as descendents two nodes in the same information set). Letting the set of leaves of $\mathcal{T}$ be denoted by $Z$, the specification of the EFG is completed by functions $u_i : Z \to \mathbb{R}$ for each $i \in [m]$, which describe each player's utility received upon reaching each leaf.

Let $\mathcal{I}_i$ denote the set of information sets of player $i$; for an information set $I \in \mathcal{I}_i$, let $A(I)$ denote the set of actions available at each node of $I$. A *policy* or (*normal-form plan*) for player $i$ in the EFG $G$ is a function $\pi_i$ which maps each $I \in \mathcal{I}_i$ to some element of $A(I)$. We denote the set of policies of player $i$ by $\Pi_i$. An EFG may be viewed as a normal form game where player $i$'s action set is $\Pi_i$, and her value function is $V_i(\pi_1, \dots, \pi_m) := \sum_{z \in Z} u_i(z) \cdot p^{\mathsf{ch}}(z) \cdot \prod_{j \in [m]} p^{\pi_j}(z)$, where $p^{\mathsf{ch}}(z)$ denotes the probability that the chance player takes a sequence of actions consistent with reaching $z$, and $p^{\pi_j}(z) \in \{0,1\}$ denotes the indicator of whether $\pi_j$ takes a sequence of actions consistent with reaching $z$.

**Sequence-form polytope.** A *sequence* for player $i$ consists of either (a) a pair $(I, a)$, where $I \in \mathcal{I}_i$ and $a \in A(I)$; or (b) the empty sequence $\emptyset$. The set of sequences for player $i$ is denoted by $\Sigma_i$. For an information set $I \in \mathcal{I}_i$, we let $\sigma_i(I) \in \Sigma_i$ denote the unique sequence for player $i$ which leads to $I$ (which is unique by perfect recall). Similarly, for a leaf $z \in Z$, let $\sigma_i(z) \in \Sigma_i$ denote the

unique sequence of player $i$ which leads to $z$. Existing external regret algorithms for learning in EFGs make use of the *sequence-form polytope* $\mathcal{Q}_i$ for each player $i$, defined below:

$$\mathcal{Q}_i := \left\{ \mathbf{q} \in \mathbb{R}^{\Sigma_i} \ : \ \mathbf{q}[\emptyset] = 1, \ \mathbf{q}[\sigma_i(I)] = \sum_{a \in A(I)} \mathbf{q}[(I,a)] \ \forall I \in \mathcal{I}_i \right\}.$$

Each element $\mathbf{q} \in \mathcal{Q}_i$ corresponds to the following distribution $P(\mathbf{q})$ over policies $\Pi_i$: at each information set $I$, sample action $a \in A(I)$ with probability $\frac{\mathbf{q}[(I,a)]}{\mathbf{q}(\sigma(I))}$, independently at each information set.[10] The key property of $P(\mathbf{q})$ is that for all $z \in Z$,

$$\mathbf{q}[\sigma_i(z)] = \mathbb{E}_{\pi_i \sim P(\mathbf{q})}[p^{\pi_i}(z)]. \tag{10}$$

We let $A_i := \max_{I \in \mathcal{I}_i} |A(I)|$. The following result establishes a guarantee for efficient external regret minimization in EFGs, thus verifying Assumption 1:

**Theorem 3.10** (Theorem 5.5 of [FLLK22])**.** *There is an algorithm running in time* $\mathrm{poly}(|\mathcal{I}_i|, A_i)$ *which, given sequentially an adversarial sequence* $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(T)} \in [0,1]^{\Sigma_i}$*, produces a sequence* $\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(T)} \in \mathcal{Q}_i$ *satisfying the following external regret guarantee:*

$$\max_{\mathbf{q}^\star \in \mathcal{Q}_i} \sum_{t=1}^{T} \left( \langle \mathbf{q}^\star, \mathbf{u}^{(t)} \rangle - \langle \mathbf{q}^{(t)}, \mathbf{u}^{(t)} \rangle \right) \le O\left( \sqrt{|\mathcal{I}_i| \log(A_i)/T} \right).$$

Theorem 3.10 improves the classical guarantee of *counterfactual regret minimization* [ZJBP07, Theorem 4], which obtains a regret guarantee of $O(|\mathcal{I}_i| \sqrt{A_i/T})$. We may now combine Theorems 3.1 and 3.10, as follows:

**Theorem 3.11.** *Given an* $m$*-player extensive-form game* $G$*, write* $I^\star := \max_{i \in [m]} |\mathcal{I}_i|$ *and* $A^\star := \max_{i \in [m]} A_i$*. For any* $\epsilon \in (0,1)$*, there is an algorithm running in time* $\mathrm{poly}(m, A^\star, (I^\star \log A^\star/\epsilon)^{1/\epsilon})$ *which outputs an* $\epsilon$*-approximate (normal-form) CE of* $G$*.*

*Proof.* Fix $T = (I^\star \log A^\star/\epsilon)^{C/\epsilon}$ for a sufficiently large constant $C$. For each $i \in [m]$, write $\mathcal{F}_i := \mathrm{co}(\{\pi_i \mapsto V_i(\pi_i, \pi_{-i}) \ : \ \pi_{-i} \in \prod_{i' \neq i} \Pi_{i'}\})$, where $\mathrm{co}(\cdot)$ denotes the convex hull. We let each player $i$ run the algorithm $\texttt{TreeSwap}(\mathcal{F}_i, \Pi_i, \texttt{Alg}_i, T)$, where $\texttt{Alg}_i$ is the algorithm of Theorem 3.10, with appropriate pre-processing and post-processing modifications (due to the fact that the algorithm of Theorem 3.10 does not technically speaking play actions in $\Delta(\Pi_i)$ nor does it receive as feedback functions in $\mathcal{F}_i$). After describing these modifications, we will apply the guarantee of Theorem 3.1 with $M = \frac{\max_i \{|\mathcal{I}_i| \log(A_i)\}}{\epsilon^2}$, and $d$ chosen so that $M^{d-1} \le T \le M^d$, so that $d \ge 1/\epsilon$.

To avoid confusion, we denote rounds of execution for each $\texttt{TreeSwap}_i$ instance by $s \in [T]$, and rounds of execution for each $\texttt{Alg}_i$ instance (used within $\texttt{TreeSwap}_i$) by $t \in [M]$. We next describe the pre-processing and post-processing modifications for $\texttt{Alg}_i$:

- *Post-processing for* $\texttt{Alg}_i$*:* At each round $t$ of execution of $\texttt{Alg}_i$, it produces a vector $\mathbf{q}_i^{(t)} \in \Delta(\mathcal{Q}_i)$; it passes $P(\mathbf{q}_i^{(t)}) \in \Delta(\Pi_i)$ for use in $\texttt{TreeSwap}$.

[10] This distribution over policies may have support which is exponential in $|\mathcal{I}_i|$; there is also an equivalent distribution over policies which is guaranteed to have polynomial-size support, which can be computed efficiently: see Theorem 4 of [CMBG19].

- *Post-processing for* $\texttt{TreeSwap}_i$*:* At each round $s$ of execution of $\texttt{TreeSwap}_i$, the algorithm $\texttt{TreeSwap}_i$ produces a distribution $P_i^{(s)} \in \Delta(\Pi_i)$, which, by the post-processing for $\texttt{Alg}_i$ described above and Line 10 of Algorithm 1, can be expressed as a uniform mixture of the form $P_i^{(s)} = \frac{1}{d}\sum_{h=1}^{d} P(\mathbf{q}_i^{(s,h)}) \in \Delta(\Pi_i)$, for vectors $\mathbf{q}_i^{(s,h)} \in \mathcal{Q}_i$. Given the distributions $P_i^{(s)}$ for each $i \in [m]$, we need to produce a function $\mathbf{f}_i^{(s)} \in \mathcal{F}_i$ to give as feedback for each $\texttt{TreeSwap}_i$ instance. To do so, we define $\bar{\mathbf{q}}_i^{(s)} := \frac{1}{d}\sum_{h=1}^{d} \mathbf{q}_i^{(s,h)}$, and then define

$$\mathbf{f}_i^{(s)}(\pi_i) := \mathbb{E}_{\pi_j \sim P(\bar{\mathbf{q}}_j^{(s)})\ \forall j \neq i}[V_i(\pi_i, \pi_{-i})]. \tag{11}$$

- *Pre-processing for* $\texttt{Alg}_i$*:* Each $\texttt{Alg}_i.\texttt{update}$ procedure in $\texttt{TreeSwap}$ is passed as input an average $\bar{\mathbf{f}}$ of elements $\mathbf{f}^{(s)} \in \mathcal{F}_i$ (in Line 6). For each of these elements $\mathbf{f}_i^{(s)}$, we compute some vector $\mathbf{u}_i^{(s)} \in [0,1]^{\Sigma_i}$, as defined below, and then average these vectors $\mathbf{u}_i^{(s)}$, producing some $\bar{\mathbf{u}} \in [0,1]^{\Sigma_i}$. The resulting average vector $\bar{\mathbf{u}}$ is then passed to the algorithm of Theorem 3.10.

  By construction, each $\mathbf{f}_i^{(s)}$ may be written of the form in (11). For each such $\mathbf{f}_i^{(s)}$, $\texttt{Alg}_i$ computes the utility vector $\mathbf{u}_i^{(s)} \in [0,1]^{\Sigma_i}$ defined by $\mathbf{u}_i^{(s)}[\sigma_i(z)] := u_i(z)p^{\mathsf{ch}}(z)\prod_{j\neq i}\mathbb{E}_{\pi_j \sim P(\bar{\mathbf{q}}_j^{(s)})}[p^{\pi_j}(z)]$ for leaves $z$, and $\mathbf{u}_i^{(s)}[\sigma_i] = 0$ for all other sequences $\sigma_i \in \Sigma_i$. Since each distribution $P(\bar{\mathbf{q}}_j^{(s)})$ randomizes independently at each information set, it is clear that $\mathbb{E}_{\pi_j \sim P(\bar{\mathbf{q}}_j^{(s)})}[p^{\pi_j}(z)]$, and thus $\mathbf{u}_i^{(s)}$, can be computed in polynomial time.

  Note that, for each $\mathbf{q}_i \in \mathcal{Q}_i$, we have

$$\begin{aligned}\langle \mathbf{q}_i, \mathbf{u}_i^{(s)} \rangle &= \sum_{z\in\mathcal{Z}} u_i(z)p^{\mathsf{ch}}(z)\mathbb{E}_{\pi_i\sim P(\mathbf{q}_i)}[p^{\pi_i}(z)] \cdot \prod_{j\neq i}\mathbb{E}_{\pi_j \sim P(\bar{\mathbf{q}}_j^{(s)})}[p^{\pi_j}(z)] \\ &= \mathbb{E}_{\pi_i\sim P(\mathbf{q}_i)}\mathbb{E}_{\pi_j \sim P(\bar{\mathbf{q}}_j^{(s)})\ \forall j\neq i}[V_i(\pi_1, \ldots, \pi_m)]. \end{aligned} \tag{12}$$

We claim that $\texttt{Alg}_i$, as defined above, satisfies the external regret bound of Assumption 1. To prove this, consider any instance of $\texttt{Alg}_i$ in $\texttt{TreeSwap}$, and consider an adversarial sequence $\bar{\mathbf{f}}_i^{(1)}, \ldots, \bar{\mathbf{f}}_i^{(M)} \in \mathcal{F}_i$ passed to $\texttt{Alg}_i$. Each $\bar{\mathbf{f}}_i^{(t)}$ may be written as an average of functions $\bar{\mathbf{f}}_i^{(t,h)}$, of the form $\bar{\mathbf{f}}_i^{(t,h)}(\pi_i) = \mathbb{E}_{\pi_j \sim P(\bar{\mathbf{q}}_i^{(t,h)})\ \forall j\neq i}[V_i(\pi_i, \pi_{-i})]$, where $\bar{\mathbf{q}}_i^{(t,h)} \in \mathcal{Q}_i$ (this is exactly the form of (11)). Accordingly, let $\bar{\mathbf{u}}^{(1)}, \ldots, \bar{\mathbf{u}}^{(M)} \in [0,1]^{\Sigma_i}$ denote the vectors produced by the above pre-processing procedure for $\texttt{Alg}_i$, and recall that, for $t \in [M]$, $P(\mathbf{q}_i^{(t)}) \in \Delta(\Pi_i)$ denotes the post-processed action distribution produced by $\texttt{Alg}_i$. Then we have

$$\max_{\pi_i^\star \in \Pi_i} \sum_{t=1}^{M} \left(\bar{\mathbf{f}}^{(t)}(\pi_i^\star) - \mathbb{E}_{\pi_i \sim P(\mathbf{q}_i^{(t)})}\bar{\mathbf{f}}^{(t)}(\pi_i)\right) = \max_{\mathbf{q}^\star \in \mathcal{Q}_i} \sum_{t=1}^{M} \left(\langle \mathbf{q}^\star, \bar{\mathbf{u}}^{(t)}\rangle - \langle \mathbf{q}_i^{(t)}, \bar{\mathbf{u}}^{(t)}\right) \leq O\left(\sqrt{|\mathcal{I}_i|\log(A_i)\cdot M}\right),$$

where the equality uses the fact that each $\pi_i^{(s)} \in \Pi_i$ can be expressed as $P(\mathbf{q}^\star)$ for some $\mathbf{q}^\star \in \mathcal{Q}_i$, as well as Equations (11) and (12), and the inequality uses Theorem 3.10. Thus, each $\texttt{Alg}_i$ satisfies Assumption 1 with $R_{\texttt{Alg}_i}(M) = O\left(\sqrt{|\mathcal{I}_i|\log(A_i)/M}\right)$.

Then by Theorem 3.1, each player's swap regret may be bounded as follows:

$$\max_{\phi:\Pi_i\to\Pi_i} \frac{1}{T}\sum_{s=1}^{T}\sum_{\pi_i\in\Pi_i} P_i^{(s)}(\pi_i)\cdot\left(\mathbb{E}_{\pi_j\sim P_j^{(s)}\ \forall j\neq i}[V_i(\phi(\pi_i),\pi_{-i})-V_i(\pi_i,\pi_{-i})]\right)$$

$$=\max_{\phi:\Pi_i\to\Pi_i} \frac{1}{T}\sum_{s=1}^{T}\sum_{\pi_i\in\Pi_i} P_i^{(s)}(\pi_i)\cdot\left(\mathbb{E}_{\pi_j\sim P(\bar{\mathbf{q}}_j^{(s)})\ \forall j\neq i}[V_i(\phi(\pi_i),\pi_{-i})-V_i(\pi_i,\pi_{-i})]\right)$$

$$=\max_{\phi:\Pi_i\to\Pi_i} \frac{1}{T}\sum_{s=1}^{T}\sum_{\pi_i\in\Pi_i} P_i^{(s)}(\pi_i)\cdot\left(\mathbf{f}_i^{(s)}[\phi(\pi_i)]-\mathbf{f}_i^{(s)}[\pi_i]\right)\le\frac{3}{d}+O\left(\max_i\sqrt{|\mathcal{I}_i|\log(A_i)/M}\right)\le O(\epsilon),$$

where the first equality uses the definition of $\bar{\mathbf{q}}_j^{(s)}$ and the fact that $\mathbb{E}_{\pi_j\sim P(\bar{\mathbf{q}}_j^{(s)})}[p^{\pi_j}(z)]=\mathbb{E}_{\pi_j\sim P_j^{(s)}}[p^{\pi_j}(z)]$ for all leaves $z\in Z$ (by (10)), the second equality uses (11), and the first inequality uses the guarantee of Theorem 3.1. By rescaling $\epsilon$ by a constant factor, we see that each player obtains swap regret of at most $\epsilon$, meaning that $\frac{1}{T}\sum_{s=1}^{T} P_1^{(s)}\times\cdots\times P_m^{(s)}\in\Delta(\Pi_1\times\cdots\times\Pi_m)$ is a (normal-form) $\epsilon$-approximate CE of the EFG. □

### 3.5 Application: bandit no-swap regret algorithm

**Algorithm 4** `BanditTreeSwap`$(N,T,M,d)$

**Require:** Action set $[N]$, time horizon $T$, parameters $M,d$ with $T/N\le M^d$.

1: For each $h\in[d]$ and sequence $\sigma\in\{0,1,\dots,M-1\}^{h-1}$, initialize an instance of `Exp3Multi`$(N,M,M^{-1/2},K^{-1}T^{-1/6},K)$ with $K=\frac{2NM^{d-h}}{d}$ and time horizon $M$, denoted `Exp3Multi`$_\sigma$. (*See Algorithm 5.*)
2: **for** $1\le t\le T$ **do**
3: Let $\sigma=(\sigma_1,\dots,\sigma_d)$ denote the base-$M$ representation of $\lfloor\frac{t-1}{N}\rfloor$.
4: **for** $1\le h\le d$ **do**
5: **if** $\sigma_{h+1}=\cdots=\sigma_d=0$ or $h=d$ **then**
6: If $\sigma_h>0$, set `Exp3Multi`$_{\sigma_{1:h-1}}$`.curDistribution` $\leftarrow$ `Exp3Multi`$_{\sigma_{1:h-1}}$`.update()`.
7: **end if**
8: **end for**
9: Sample $h^{(t)}\sim[d]$ uniformly at random.
10: Output an action $a^{(t)}\sim$ `Exp3Multi`$_{\sigma_{1:h^{(t)}-1}}$`.curDistribution`.
11: Observe reward $\mathbf{u}^{(t)}[a^{(t)}]$, and call `Exp3Multi`$_{\sigma_{1:h^{(t)}-1}}$`.storeSample`$(a^{(t)},\mathbf{u}^{(t)}[a^{(t)}])$.
12: **end for**

In this section, we discuss an application of our techniques in `TreeSwap` to the *bandit* setting. The bandit setting with a finite number of arms is identical to the "$N$ experts" setting of Section 3.1, with the exception that the learning algorithm has to choose a single action $a^{(t)}\in[N]$ at each round $t$. As feedback, the learner only sees the coordinate of the utility vector which it selected at round $t$: in particular, if the adversary plays $\mathbf{u}^{(t)}\in[0,1]^N$, the learner observes only $\mathbf{u}^{(t)}[a^{(t)}]$ (as opposed to the entire vector $\mathbf{u}^{(t)}$). In the bandit setting, the swap regret is defined exactly as in Definition 2.2

with the distributions $\mathbf{x}^{(t)}$ interpreted as singletons on $a^{(t)}$: explicitly, we have

$$\mathbf{SwapRegret}(a^{(1:T)}, \mathbf{u}^{(1:T)}) = \sup_{\pi:[N]\to[N]} \frac{1}{T}\sum_{t=1}^{T}\left(\mathbf{u}^{(t)}[\pi(a^{(t)})] - \mathbf{u}^{(t)}[a^{(t)}]\right).$$

When the context is clear, we abbreviate $\mathbf{SwapRegret}(T) = \mathbf{SwapRegret}(a^{(1:T)}, \mathbf{u}^{(1:T)})$. [BM07] showed that any algorithm in the bandit setting which achieves $\mathbf{SwapRegret}(T) \le \epsilon$ requires $T \ge \Omega\left(\frac{N}{\epsilon^2}\right)$ rounds; this bound was improved to $T \ge \Omega\left(\frac{N\log N}{\epsilon^2}\right)$ by [Ito20]. The best upper bounds were larger by a polynomial factor: [BM07] showed that it suffices to have $T \le O\left(\frac{N^3\log N}{\epsilon^2}\right)$ which was improved to $T \le O\left(\frac{N^2\log N}{\epsilon^2}\right)$ by [Ito20; JLWY22], still leaving a quadratic gap from the lower bound of [BM07].[11] Our upper bound in Theorem 3.12 below closes this quadratic gap up to $\operatorname{poly}\log N$ factors in the setting of constant $\epsilon$. Finally, for simplicity, we state and prove Theorem 3.12 for an *oblivious* adversary, as is somewhat standard in the adversarial bandit setting [LS20]. However, our techniques extend readily (with some more cumbersome notation) to the adaptive adversary setting.

**Theorem 3.12.** *Let $N \in \mathbb{N}$, $\epsilon \in (0,1)$ be given, and consider any $T \ge N \cdot (\log(N)/\epsilon)^{O(1/\epsilon)}$. Let $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(T)}$ be a fixed (deterministic) sequence of reward vectors (i.e., produced by an oblivious adversary). Then there is a bandit algorithm which, at each time step $t$, plays an action $a^{(t)}$ and observes only $\mathbf{u}^{(t)}[a^{(t)}]$, for which the expected swap regret may be bounded by*

$$\mathbb{E}[\mathbf{SwapRegret}(a^{(1:T)}, \mathbf{u}^{(1:T)})] \le \epsilon.$$

Theorem 3.12 is proved using a variant of `TreeSwap`, namely `BanditTreeSwap` (Algorithm 4). `BanditTreeSwap` operates in a similar manner to `TreeSwap`, namely by choosing $M, d$ so that $T \approx M^d$ and then constructing a $M$-ary tree of depth $d$, at each node of which lies a bandit no-external regret algorithm, which we instantiate as `Exp3Multi` (Algorithm 5), discussed below. Due to the challenges of the bandit setting, the semantics of the algorithm `Exp3Multi` are slightly different from those of the external regret minimizer `Alg` used in `TreeSwap`. In particular, each instance `Exp3Multi` operates over multiple rounds (when $T = M^d$ in the context of `BanditTreeSwap`, the number of rounds for each `Exp3Multi` instance will be $M$). Within each round $s$, `Exp3Multi` fixes a distribution $\mathbf{p}^{(s)} \in \Delta_N$ and draws several actions $a^{(s,k)} \sim \mathbf{p}^{(s)}$. It is then given samples of the form $(a^{(k,s)}, u^{(k,s)})$, for scalars $u^{(k,s)} \in \mathbb{R}$.

To process these samples, we assume that `Exp3Multi` has the following subprocedures:

- A procedure `Exp3Multi.storeSample`$(a, u)$, which stores the sample $(a, u) \in [N] \times \mathbb{R}$ in a memory buffer for the current round.
- A procedure `Exp3Multi.update`$(s)$, which takes as input the current round index $s$ (so that $1 \le s \le M$ in the context of `BanditTreeSwap`) and uses the samples stored in the current round $s$ to compute a distribution over actions $\mathbf{p}^{(s+1)} \in [N]$ to be played in the subsequent round. `Exp3Multi.update`$(s)$ then returns this distribution $\mathbf{p}^{(s+1)}$. Note that `Exp3Multi.update`$(s)$ always marks the end of the current round $s$ and the beginning of round $s+1$.

[11] We remark that the upper bound of [Ito20] bounds only the weaker notion of pseudo-swap regret.

**Instantiation of `Exp3Multi`.** The algorithm `Exp3Multi` (Algorithm 5) is a variant of the algorithm `EXP3-IX` which obtains sublinear regret for the adversarial bandit problem (see Chapter 12 of [LS20]).[12] The `update` procedure of `Exp3Multi` is similar to that of `EXP3-IX`: suppose `Exp3Multi` was given samples for round $s$ of the form $(a^{(s,k)}, u^{(s,k)})$ for $1 \le k \le K^{(s)}$, where $K^{(s)}$ denotes the total number of samples in round $s$. Each scalar $u^{(s,k)}$ should be interpreted as the $a^{(s,k)}$-th entry of some reward vector $\mathbf{u}^{(s,k)} \in [0,1]^N$, namely $u^{(s,k)} = \mathbf{u}^{(s,k)}[a^{(s,k)}]$. Then `Exp3Multi.update`$(s)$ constructs an importance-weighted estimator of $\sum_{k=1}^{K^{(s)}} \mathbf{u}^{(s,k)}$ (Line 11), and uses this importance weighted estimator to compute a multiplicative weights update to produce $\mathbf{p}^{(s+1)}$ (Line 13).

We will show that, for all instantiations of `Exp3Multi` in `BanditTreeSwap`, the value of $K^{(s)}$, for all rounds $s$, will be bounded below by $\Omega(N)$ with high probability. Thus, at a high level, one can think of each round of `Exp3Multi` as an attempt to "simulate" a full-information update of the exponential weights algorithm: the $K^{(s)} = \Omega(N)$ steps within round $s$ allow one to construct an estimator $\hat{\mathbf{u}}^{(s)}$ of the utility vector $\sum_{k=1}^{K^{(s)}} \mathbf{u}^{(k,s)}$ whose entries generally have variance $O(1)$. A key challenge in analyzing `Exp3Multi` is that the distribution $\mathbf{p}^{(s)}$ (from which the actions $a^{(k,s)}$ are drawn) is not uniform; thus, one must carefully account for the fact that some entries of the estimator $\hat{\mathbf{u}}^{(s)}$ may have large variance.

Below we state the main technical lemma in the proof of Theorem 3.12. In turn, its proof makes use of an external regret bound for `Exp3Multi`, which is stated below in Lemma 3.14.

**Lemma 3.13.** *There is a constant $C > 0$ so that the following holds. Let $N, T, M, d \in \mathbb{N}$ be given so that $T$ is a multiple of $N$, and $M^{d-1} \le T/N \le M^d$. Let $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(T)} \in [0,1]^N$ be a fixed (deterministic) sequence of reward vectors (i.e., produced by an oblivious adversary).*

*Then the expected swap regret of* `BanditTreeSwap`$(N, T, M, d)$ *(Algorithm 4), which produces a sequence of actions $a^{(1)}, \dots, a^{(T)} \in [N]$, may be bounded as follows:*

$$T \cdot \mathbb{E}[\mathbf{SwapRegret}(a^{(1:T)}, \mathbf{u}^{(1:T)})] \le \frac{3T}{d} + C\sqrt{TN\log(NT)} + C \cdot T \cdot \frac{\log^2(N)}{M^{1/6}} + \frac{CdT^2}{N} e^{-N/(3d)}.$$

First, we prove Theorem 3.12, assuming Lemma 3.13.

*Proof of Theorem 3.12.* Fix $\epsilon > 0$, and let $T = N \cdot (\log(N)/\epsilon)^{C_0/\epsilon}$, for a constant $C_0$ to be specified below. By increasing $T$ by at most $N$, we may assume without loss of generality that $T$ is a multiple of $N$. Moreover, note that [JLWY22, Corollary 25][13] establishes that there is an algorithm achieving expected swap regret $\mathbb{E}[\mathbf{SwapRegret}(T)] \le O(N\sqrt{\log(N)/T})$; in particular, given $\epsilon \in (0,1)$, if we take $T \ge C_1 \cdot N^2 \log(N)/\epsilon^2$, for a sufficiently large constant $C_1 > 0$, we obtain $\mathbb{E}[\mathbf{SwapRegret}(T)] \le \epsilon$. Thus, we may assume from here on that $\epsilon$ is chosen so that $T = N \cdot (\log(N)/\epsilon)^{C_0/\epsilon} < C_1 N^2 \log(N)/\epsilon^2$. As long as $C_0$ is sufficiently large, this inequality only holds when $\epsilon \ge 1/\log(N)$.

Let $C$ denote the constant in the statement of Lemma 3.13, define $M := \lceil C^6 \log^{12}(N)\epsilon^{-6} \rceil$, and choose $d \ge \lceil \epsilon^{-1} \rceil$ so that $M^{d-1} \le T/N \le M^d$ (such a choice of $d$ is possible by our choice of $T = N \cdot (\log(N)/\epsilon)^{C_0/\epsilon}$, as long as $C_0$ is chosen sufficiently large). Note that $d \le 1 + \log(T/N)/\log(M) \le 2C_0/\epsilon \cdot \log(\log(N)/\epsilon) \le 4C_0 \log(N) \cdot \log\log(N) < C_2\sqrt{N}/3 \le C_2 N$ for a sufficiently large constant $C_2$.

[12] We use `EXP3-IX` instead of the more well-known algorithm `EXP3` because we wish to obtain high-probability bounds, which `EXP3` does not guarantee [LS20, Exercise 11.6].

[13] In particular, set $H = 1$ in the statement of Corollary 25.

**Algorithm 5** `Exp3Multi`$(N, T, \eta, \gamma, K)$

**Require:** Action set $[N]$, time horizon $T$, step size $\eta > 0$, parameters $\gamma > 0$ and $K \in \mathbb{N}$.

Initialize $\hat{L}_a^{(0)} = 0$ for all $a \in [N]$ and $\mathbf{p}^{(1)} \leftarrow \mathsf{Unif}([N])$.
**for** round $1 \le t \le T$ **do**
    Let $K^{(t)} \in \mathbb{N}$ denote the number of samples in round $t$. ▷ *$K^{(t)}$ need not be known prior to the round.*
    **for** step $1 \le k \le K^{(t)}$ **do**
        Play action $a_k^{(t)} \sim \mathbf{p}^{(t)}$, and receive $u_k^{(t)} = \mathbf{u}_k^{(t)}[a_k^{(t)}]$.
        Call `Exp3Multi.storeSample`$(a_k^{(t)}, u_k^{(t)})$.
    **end for**
    Set $\mathbf{p}^{(t+1)} \leftarrow$ `Exp3Multi.update`$(t)$.
**end for**
**function** `Exp3Multi.update`$(t)$
    For $a \in [N]$, define $\hat{Y}_a^{(t)} := \frac{1}{K} \cdot \sum_{k=1}^{K^{(t)}} \left( \frac{\mathbb{1}\{a_k^{(t)} = a\} \cdot (1 - u_k^{(t)})}{\mathbf{p}^{(t)}[a] + \gamma} \right)$.
    Define $\hat{L}_a^{(t)} \leftarrow \hat{L}_a^{(t-1)} + \hat{Y}_a^{(t)}$ for all $a \in [N]$.
    **return** the distribution $\mathbf{p} \in \Delta_N$ defined by, for $a \in [N]$,

$$\mathbf{p}[a] = \frac{\exp\left(-\eta \hat{L}_a^{(t)}\right)}{\sum_{b=1}^{N} \exp\left(-\eta \hat{L}_b^{(t)}\right)}.$$

**end function**
**function** `storeSample`$(a, u)$
    Store $(a, u)$ in memory.
**end function**

Then by Lemma 3.13 with the chosen values of $M, d$, we obtain a (normalized) swap regret of

$$\begin{aligned}
&\frac{3}{d} + C\sqrt{N\log(NT)/T} + \frac{C\log^2(N)}{M^{1/6}} + \frac{CdT^2}{N}e^{-N/(3d)} \\
\leq& 4\epsilon + C\sqrt{\log(NT)}\cdot(T/N)^{-1/4} + CC_2(C_1N^2\log(N)/\epsilon^2)^2 e^{-N/(3d)} \\
\leq& 4\epsilon + C\sqrt{\log(NT)}\cdot(T/N)^{-1/4} + CC_2(C_1N^2\log^3(N))^2 e^{-C_2\sqrt{N}} \leq C_3\epsilon,
\end{aligned}$$

where the last inequality holds for a sufficiently large constant $C_3$ and it uses the fact that $1/\log(N) \leq \epsilon$. The theorem statement follows by rescaling $\epsilon$ by a factor of $C_3$. □

The proof of Lemma 3.13 proceeds in a similar manner to that of Theorem 3.1, with the added complication that we need to ensure that the empirical estimates derived from the sampled actions $a^{(t)}$ in `BanditTreeSwap` concentrate to their means.

*Proof of Lemma 3.13.* For each $1 \leq h \leq d$, let $\Sigma_{h-1} := \{\sigma_{1:h-1} = (\sigma_1, \ldots, \sigma_{h-1}) : 1 + N \cdot \sum_{g=1}^{h-1} \sigma_g \cdot M^{d-g} \leq T\}$ denote the set of prefixes of sequences of length $h-1$ encountered over the course of $T$ rounds of `BanditTreeSwap`. (Note that $\Sigma_{h-1}$ depends on $T$.) Consider any sequence $\sigma_{1:h-1} = (\sigma_1, \ldots, \sigma_{h-1}) \in \Sigma_{h-1}$ representing a node in the tree, and define $M'(\sigma_{1:h-1}) := \max\{\sigma_h \in \{0, \ldots, M-1\} : \sigma_{1:h} \in \Sigma_h\}$. Then let $\mathbf{x}^{(0)}_{\sigma_{1:h-1}}, \ldots, \mathbf{x}^{(M'(\sigma_{1:h-1}))}_{\sigma_{1:h-1}}$ denote the $M'(\sigma_{1:h-1})$ distributions chosen by the algorithm $\texttt{Exp3Multi}_{\sigma_{1:h-1}}$ over the course of the $T$ rounds (i.e., the $M'(\sigma_{1:h-1})$ different values taken by $\texttt{Exp3Multi}_{\sigma_{1:h-1}}.\texttt{curDistribution}$). For $h \geq 0$, we let $\underline{\tau}(\sigma_{1:h}) := 1 + N \cdot \sum_{g=1}^{h} \sigma_g \cdot M^{d-g}$ denote the first round when $\sigma_{1:h}$ is encountered, and $\overline{\tau}(\sigma_{1:h}) := \max\{T, \underline{\tau}(\sigma_{1:h}) + N \cdot M^{d-h} - 1\}$ denote the last round when $\sigma_{1:h}$ is encountered. Finally, write

$$\begin{aligned}
\mathcal{T}(\sigma_{1:h}) &= \left\{t \in [\underline{\tau}(\sigma_{1:h}), \overline{\tau}(\sigma_{1:h})] \ : \ h^{(t)} - 1 = h - 1\right\} \\
\overline{\mathcal{T}}(\sigma_{1:h-1}) &= \bigcup_{\sigma_h=0}^{M'(\sigma_{1:h-1})} \mathcal{T}(\sigma_{1:h}) = \left\{t \in [\underline{\tau}(\sigma_{1:h-1}), \overline{\tau}(\sigma_{1:h-1})] \ : \ h^{(t)} - 1 = h - 1\right\}.
\end{aligned}$$

Notice that the only randomness used in `BanditTreeSwap` is the draws of $h^{(t)}, a^{(t)}$ in Lines 9 and 10. Accordingly, let $\mathcal{F}^{(t)}$ denote the sigma algebra generated by $\{(h^{(s)}, a^{(s)})\}_{s=1}^{t}$. We let $\mathbb{E}_t[\cdot]$ denote expectation conditioned on $\mathcal{F}^{(t)}$. Given $t \in [T]$ for which the binary representation of $\lfloor \frac{t-1}{N} \rfloor$ is $\sigma = (\sigma_1, \ldots, \sigma_d)$, let $\mathbf{p}^{(t)} := \frac{1}{d}\sum_{h=1}^{d} \mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}}$, so that, conditioned on $\mathcal{F}^{(t-1)}$, $a^{(t)} \sim \mathbf{p}^{(t)}$. (This uses the sampling procedure for $a^{(t)}$ on Line 10 as well as the definition of $\texttt{Exp3Multi}_{\sigma_{1:h-1}}.\texttt{curDistribution}$ on Line 6.)

First, we expand the total (unnormalized) reward of the learner over the $T$ rounds, as follows:

$$\sum_{t=1}^{T} \mathbf{u}^{(t)}[a^{(t)}] = \sum_{h=1}^{d} \sum_{\sigma_{1:h-1} \in \Sigma_{h-1}} \sum_{\sigma_h=0}^{M'(\sigma_{1:h-1})} \sum_{s \in \mathcal{T}(\sigma_{1:h})} \mathbf{u}^{(s)}[a^{(s)}]. \tag{13}$$

Now consider any function $\pi : [N] \to [N]$ and $\delta \in (0,1)$. For $\mathbf{u} \in [0,1]^N$, define $(\pi \circ \mathbf{u}) \in [0,1]^N$ by $(\pi \circ \mathbf{u})[a] := \mathbf{u}[\pi(a)]$. The learner's total reward under the swap function $\pi$ is given by $\sum_{t=1}^{T} \mathbf{u}^{(t)}[\pi(a^{(t)})]$. For each $t \in [T]$, we have that, $\mathbb{E}_{t-1}\left[\mathbf{u}^{(t)}[\pi(a^{(t)})]\right] = \langle \mathbf{p}^{(t)}, \pi \circ \mathbf{u}^{(t)} \rangle$. Thus, by

the Azuma-Hoeffding inequality, we have that, for any $\delta \in (0,1)$, with probability $1-\delta$ over the randomness in `BanditTreeSwap`,

$$\left| \sum_{t=1}^{T} \mathbf{u}^{(t)}[\pi(a^{(t)})] - \langle \mathbf{p}^{(t)}, \pi \circ \mathbf{u}^{(t)} \rangle \right| \le C\sqrt{T \log(1/\delta)}, \tag{14}$$

for a sufficiently large constant $C$.

Next, we expand

$$\begin{aligned}
\sum_{t=1}^{T} \langle \mathbf{p}^{(t)}, \pi \circ \mathbf{u}^{(t)} \rangle =& \frac{1}{d} \sum_{h=1}^{d} \sum_{\sigma_{1:h-1} \in \Sigma_{h-1}} \sum_{\sigma_h=0}^{M'(\sigma_{1:h-1})} \sum_{s=\underline{\tau}(\sigma_{1:h})}^{\overline{\tau}(\sigma_{1:h})} \langle \mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}}, \pi \circ \mathbf{u}^{(s)} \rangle \\
=& \frac{1}{d} \sum_{h=1}^{d} \sum_{\sigma_{1:h} \in \Sigma_h} \sum_{s=\underline{\tau}(\sigma_{1:h})}^{\overline{\tau}(\sigma_{1:h})} \langle \mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}}, \pi \circ \mathbf{u}^{(s)} \rangle \\
=& \sum_{\substack{s \in [T] \\ \sigma_{1:d} := \lfloor \frac{s-1}{N} \rfloor}} \frac{1}{d} \sum_{h=1}^{d} \langle \mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}}, \pi \circ \mathbf{u}^{(s)} \rangle.
\end{aligned} \tag{15}$$

For each $\sigma_{1:h}$ and $s \in [\underline{\tau}(\sigma_{1:h}), \overline{\tau}(\sigma_{1:h})]$, we have that

$$\frac{1}{d} \langle \mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}}, \pi \circ \mathbf{u}^{(s)} \rangle = \mathbb{E}_{s-1} \left[ \mathbb{1}\{h^{(s)} - 1 \equiv h\} \cdot \langle \mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}}, \pi \circ \mathbf{u}^{(s)} \rangle \right],$$

where the statement $h^{(s)} - 1 \equiv h$ is to be interpreted modulo $d$ (i.e., we have $0 \equiv d$). Thus, for each $s \in [T]$, letting the binary representation of $\lfloor \frac{s-1}{N} \rfloor$ be $\sigma_{1:d}$, we have

$$\frac{1}{d} \sum_{h=1}^{d-1} \langle \mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}}, \pi \circ \mathbf{u}^{(s)} \rangle = \sum_{h=1}^{d-1} \mathbb{E}_{s-1} \left[ \mathbb{1}\{h^{(s)} - 1 \equiv h\} \cdot \langle \mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}}, \pi \circ \mathbf{u}^{(s)} \rangle \right].$$

Thus, by the Azuma-Hoeffding inequality, with probability $1-\delta$,

$$\left| \sum_{\substack{s \in [T] \\ \sigma_{1:d} := \lfloor \frac{s-1}{N} \rfloor}} \frac{1}{d} \sum_{h=1}^{d-1} \langle \mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}}, \pi \circ \mathbf{u}^{(s)} \rangle - \sum_{\substack{s \in [T] \\ \sigma_{1:d} := \lfloor \frac{s-1}{N} \rfloor}} \sum_{h=1}^{d-1} \mathbb{1}\{h^{(s)} - 1 = h\} \cdot \langle \mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}}, \pi \circ \mathbf{u}^{(s)} \rangle \right| \le C\sqrt{T \log(1/\delta)}, \tag{16}$$

for a sufficiently large constant $C$. We moreover have that

$$\begin{aligned}
&\sum_{\substack{s\in[T]\\ \sigma_{1:d}:=\lfloor\frac{s-1}{N}\rfloor}} \sum_{h=1}^{d-1} \mathbb{1}\{h^{(s)}-1=h\}\cdot\langle \mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}}, \pi\circ\mathbf{u}^{(s)}\rangle \\
&= \sum_{h=1}^{d-1}\sum_{\sigma_{1:h}\in\Sigma_h}\sum_{s=\underline{\tau}(\sigma_{1:h})}^{\overline{\tau}(\sigma_{1:h})} \mathbb{1}\{h^{(s)}-1=h\}\cdot\langle \mathbf{x}^{(\sigma_h)}_{\sigma_{1:h-1}}, \pi\circ\mathbf{u}^{(s)}\rangle \\
&= \sum_{h=2}^{d}\sum_{\sigma_{1:h-1}\in\Sigma_{h-1}}\sum_{s=\underline{\tau}(\sigma_{1:h-1})}^{\overline{\tau}(\sigma_{1:h-1})} \mathbb{1}\{h^{(s)}-1=h-1\}\cdot\langle \mathbf{x}^{(\sigma_{h-1})}_{\sigma_{1:h-2}}, \pi\circ\mathbf{u}^{(s)}\rangle \\
&= \sum_{h=2}^{d}\sum_{\sigma_{1:h-1}\in\Sigma_{h-1}}\sum_{s\in\overline{\mathcal{T}}(\sigma_{1:h-1})} \langle \mathbf{x}^{(\sigma_{h-1})}_{\sigma_{1:h-2}}, \pi\circ\mathbf{u}^{(s)}\rangle \\
&\le \sum_{h=2}^{d}\sum_{\sigma_{1:h-1}\in\Sigma_{h-1}}\max_{a^\star\in[N]}\sum_{s\in\overline{\mathcal{T}}(\sigma_{1:h-1})} \mathbf{u}^{(s)}[a^\star]. \qquad (17)
\end{aligned}$$

Fix any $\delta\in(0,1)$. It follows by a union bound over all $\pi:[N]\to[N]$ in (14) and (16) as well as (13), (15), and (17) that under some event $\mathcal{E}_1$ occurring with probability $1-\delta/3$, for all $\pi:[N]\to[N]$,

$$\begin{aligned}
&\sum_{t=1}^{T}\left(\mathbf{u}^{(t)}[\pi(a^{(t)})]-\mathbf{u}^{(t)}[a^{(t)}]\right) \\
&\le \sum_{t=1}^{t}\langle \mathbf{p}^{(t)}, \pi\circ\mathbf{u}^{(t)}\rangle - \sum_{h=1}^{d}\sum_{\sigma_{1:h-1}\in\Sigma_{h-1}}\sum_{\sigma_h=0}^{M'(\sigma_{1:h-1})}\sum_{s\in\mathcal{T}(\sigma_{1:h})}\mathbf{u}^{(s)}[a^{(s)}] + C\sqrt{TN\log(N/\delta)} \\
&\le \frac{T}{d} + \sum_{h=2}^{d}\sum_{\sigma_{1:h-1}\in\Sigma_{h-1}}\left(\max_{a^\star\in[N]}\sum_{s\in\overline{\mathcal{T}}(\sigma_{1:h-1})}\mathbf{u}^{(s)}[a^\star] - \sum_{s\in\overline{\mathcal{T}}(\sigma_{1:h-1})}\mathbf{u}^{(s)}[a^{(s)}]\right) + C\sqrt{TN\log(N/\delta)}, \qquad (18)
\end{aligned}$$

where we have also used that $\mathbf{u}^{(t)}\in[0,1]^N$ for all $t\in[T]$. (In particular, the first inequality above uses Equations (13) and (14), and the second inequality uses Equations (15) to (17).)

Note that the draws $h^{(t)}\sim[d]$ in Line 9 are all independent. Thus, for each sequence $\sigma_{1:h}\in\Sigma_h$, we have that $|\mathcal{T}(\sigma_{1:h})|\sim\mathrm{Bin}(NM^{d-h},1/d)$. Thus, for any such sequence $\sigma_{1:h}$, it holds by a Chernoff bound that

$$\Pr\left(|\mathcal{T}(\sigma_{1:h})|\le\frac{2NM^{d-h}}{d}\right)\ge 1-e^{-NM^{d-h}/(3d)}\ge 1-e^{-N/(3d)},$$

and in the event that $\overline{\tau}(\sigma_{1:h})-\underline{\tau}(\sigma_{1:h})=N\cdot M^{d-h}-1$,

$$\Pr\left(|\mathcal{T}(\sigma_{1:h})|\in\left[\frac{NM^{d-h}}{2d},\frac{2NM^{d-h}}{d}\right]\right)\ge 1-2e^{-NM^{d-h}/(3d)}\ge 1-2e^{-N/(3d)},$$

By a union bound over $h \in [d]$ and $\sigma_h \in \Sigma_h$ for which $\overline{\tau}(\sigma_{1:h}) - \underline{\tau}(\sigma_{1:h}) = N \cdot M^{d-h} - 1$, we have that, under some event $\mathcal{E}_2$ occurring with probability at least $1 - \frac{2dT}{N} e^{-N/(3d)}$, for all such sequences $\sigma_{1:h}$, $|\mathcal{T}(\sigma_{1:h})| \le \frac{2NM^{d-h}}{d}$, and for $\sigma_{1:h}$ satisfying $\overline{\tau}(\sigma_{1:h}) - \underline{\tau}(\sigma_{1:h}) = N \cdot M^{d-h} - 1$, $|\mathcal{T}(\sigma_{1:h})| \in [\frac{NM^{d-h}}{2d}, \frac{2NM^{d-h}}{d}]$.

Then by Lemma 3.14[14] with $T = M$ and $K = \frac{2NM^{d-h}}{d}$ (which are the parameters that the instance $\texttt{Exp3Multi}_{\sigma_{1:h-1}}$ was initialized with on Line 1 of Algorithm 4) and a union bound, there is some event $\mathcal{E}_3$ with probability at least $1 - \delta/3$ so that, under $\mathcal{E}_2 \cap \mathcal{E}_3$, for all $h \in [d]$ and $\sigma_{1:h-1} \in \Sigma_{h-1}$ for which $\overline{\tau}(\sigma_{1:h-1}) - \underline{\tau}(\sigma_{1:h-1}) = N \cdot M^{d-h+1} - 1$,

$$\max_{a^\star \in [N]} \sum_{s \in \overline{\mathcal{T}}(\sigma_{1:h-1})} \mathbf{u}^{(s)}[a^\star] - \sum_{s \in \overline{\mathcal{T}}(\sigma_{1:h-1})} \mathbf{u}^{(s)}[a^{(s)}] \le C \cdot \frac{NM^{d-h}}{d} \cdot M^{5/6} \cdot \log^2(TN/\delta) + N \cdot M^{5/6}. \tag{19}$$

Moreover, note that for each $2 \le h \le d-1$, there is at most a single sequence $\sigma_{1:h-1} \in \Sigma_{h-1}$ for which $\overline{\tau}(\sigma_{1:h-1}) - \underline{\tau}(\sigma_{1:h-1}) < N \cdot M^{d-h+1} - 1$. For any such sequence $\sigma_{1:h-1}$, we may bound

$$\max_{a^\star \in [N]} \sum_{s \in \overline{\mathcal{T}}(\sigma_{1:h-1})} \mathbf{u}^{(s)}[a^\star] - \sum_{s \in \overline{\mathcal{T}}(\sigma_{1:h-1})} \mathbf{u}^{(s)}[a^{(s)}] \le |\overline{\mathcal{T}}(\sigma_{1:h-1})| \le \frac{2NM^{d-h+1}}{d}, \tag{20}$$

where the final inequality holds under $\mathcal{E}_2$.

Combining (18), (19), and (20), we see that, under $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$ (which occurs with probability at least $1 - \delta - \frac{2dT}{N} e^{-N/(3d)}$),

$$\begin{aligned}
&\max_{\pi:[N]\to[N]} \sum_{t=1}^{T} \left( \mathbf{u}^{(t)}[\pi(a^{(t)})] - \mathbf{u}^{(t)}[a^{(t)}] \right) \\
&\le \frac{T}{d} + C\sqrt{TN \log(TN/\delta)} \\
&\quad + \sum_{h=2}^{d} \left( \frac{2NM^{d-h+1}}{d} + \sum_{\substack{\sigma_{1:h-1} \in \Sigma_{h-1}: \\ \overline{\tau}(\sigma_{1:h-1}) - \underline{\tau}(\sigma_{1:h-1}) = NM^{d-h+1}}} \left( C \cdot \frac{NM^{d-h}}{d} \cdot M^{5/6} \log^2(TN/\delta) + N \cdot M^{5/6} \right) \right) \\
&\le \frac{T}{d} + C\sqrt{TN \log(N/\delta)} + \frac{2NM^{d-1}}{d} + C \cdot T \cdot M^{-1/6} \log^2(TN/\delta) + TM^{-1/6} \cdot \sum_{h=2}^{d} M^{h-d} \\
&\le \frac{3T}{d} + C\sqrt{TN \log(TN/\delta)} + C \cdot T \cdot \frac{\log^2(TN/\delta)}{M^{1/6}},
\end{aligned}$$

where the second inequality uses that $\sum_{h=2}^{d} M^{d-h+1} \le M^{d-1}$, and the final inequality uses that $NM^{d-1} \le T$ by assumption and that $\sum_{h=2}^{d} M^{h-2} \le 2$. Choosing $\delta = 1/T$ yields that

$$T \cdot \left( \delta + \frac{2dT}{N} e^{-N/(3d)} \right) \le 1 + \frac{2dT^2}{N} e^{-N/(3d)},$$

[14] Note that technically, Lemma 3.14 applies to the main procedure in `Exp3Multi` (Algorithm 5), which is not directly called in `BanditTreeSwap`; however, note that this procedure is simulated exactly by the instances $\texttt{Exp3Multi}_{\sigma_{1:h-1}}$ in `BanditTreeSwap`, where the quantities $K^{(s)}$ in `Exp3Multi` correspond to the number of time steps during each round of each $\texttt{Exp3Multi}_{\sigma_{1:h-1}}$ instance that $h^{(t)} = h$.

and thus the claimed expected swap regret bound holds.

□

**Analysis of `Exp3Multi`.** Next we state the main external regret guarantee for `Exp3Multi`. It states that when each $K^{(t)}$ is within a constant factor of some parameter $K$, then the external regret of `Exp3Multi` is bounded by the product of $K$ and a quantity which is sublinear in the number of rounds $T$.

**Lemma 3.14.** *Let $N, T, K \in \mathbb{N}$ be given, and let $K^{(1)}, \dots, K^{(T)}$ be fixed so that for all $t \in [T]$, $K^{(t)} \in [K/4, K]$. let $(\mathbf{u}_k^{(t)} \in [0,1]^N)_{t\in[T], k\in[K^{(t)}]}$ be a fixed sequence of reward vectors. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$, the actions $a_k^{(t)}$ of `Exp3Multi`$(N, T, T^{-1/2}, K^{-1}T^{-1/6}, K)$ (Algorithm 5) satisfy*

$$\max_{a^\star \in [N]} \sum_{t=1}^{T} \sum_{k=1}^{K^{(t)}} \mathbf{u}_k^{(t)}[a^\star] - \mathbf{u}_k^{(t)}[a_k^{(t)}] \le K \cdot O\left(\log^2(TN/\delta) \cdot T^{5/6}\right) + N \cdot T^{5/6}.$$

*Proof.* Let $\eta = \frac{1}{\sqrt{T}}, \gamma = \frac{1}{KT^{1/6}}$ be the parameters passed to `Exp3Multi`.

Recall from the definitions in Algorithm 5 that $\hat{L}_a^{(t)} = \sum_{s=1}^{t} \hat{Y}_a^{(s)}$ and $\hat{Y}_a^{(t)} := \frac{1}{K} \cdot \sum_{k=1}^{K^{(t)}} \left( \frac{\mathbb{1}\{a_k^{(t)} = a\} \cdot (1 - u_k^{(t)})}{\mathbf{p}^{(t)}[a] + \gamma} \right)$ for each $t \in [T]$. We also define $\hat{L}^{(t)} = \sum_{s=1}^{t} \sum_{a=1}^{N} \mathbf{p}^{(s)}[a] \cdot \hat{Y}_a^{(s)}$. Additionally, we define $\mathbf{y}_k^{(t)} = \mathbf{1} - \mathbf{u}_k^{(t)}$, $y_k^{(t)} = 1 - u_k^{(t)}$, and

$$\mathbf{y}^{(t)} := \frac{1}{K} \sum_{k=1}^{K^{(t)}} \mathbf{y}_k^{(t)}, \qquad y^{(t)} = \frac{1}{K} \sum_{k=1}^{K^{(t)}} y_k^{(t)}$$

to denote the total *loss* vectors and realized *losses* in each time step $t \in [T]$, as well as

$$\tilde{L}^{(t)} = \sum_{s=1}^{t} y^{(s)}, \qquad L_a^{(t)} = \sum_{s=1}^{t} \mathbf{y}^{(s)}[a], \qquad R_a^{(t)} = \sum_{s=1}^{t} y^{(s)} - \sum_{s=1}^{t} \mathbf{y}^{(s)}[a] = \tilde{L}^{(t)} - L_a^{(t)}$$

to denote, respectively, the cumulative loss up to $t$ experienced by the learner, the cumulative loss up to $t$ experienced by taking action $a$, and the cumulative regret associated with action $a$ up to time step $t$. Finally, for each $k \in [K^{(t)}]$, write $\hat{Y}_a^{(t,k)} := \frac{\mathbb{1}\{a_k^{(t)} = a\} \cdot (1 - u_k^{(t)})}{\mathbf{p}^{(t)}[a] + \gamma}$, so that $\hat{Y}_a^{(t)} = \frac{1}{K} \sum_{k=1}^{K^{(t)}} \hat{Y}_a^{(t,k)}$.

**Step 1: Regret decomposition.** We consider the following decomposition of $\hat{R}_a^{(t)}$:

$$\hat{R}_a^{(t)} = (\tilde{L}^{(t)} - \hat{L}^{(t)}) + (\hat{L}^{(t)} - \hat{L}_a^{(t)}) + (\hat{L}_a^{(t)} - L_a^{(t)}). \tag{21}$$

We bound each of the terms in (21) in the below lemmas, whose proofs are provided in Section 3.5.1.

**Lemma 3.15.** *Let $\delta \in (0,1)$ be given. Then with probability at least $1-\delta$, for each $t \in [T]$ and $a \in [N]$,*

$$\hat{L}^{(t)} - \hat{L}_a^{(t)} \le \frac{\log N}{\eta} + 2\eta t + \frac{2\eta t \log^2(TN/\delta)}{(K\gamma)^2}.$$

**Lemma 3.16.** *There is a sufficiently large constant $C$ so that for any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\max_{a\in[N]}(\hat{L}_a^{(T)} - L_a^{(T)}) \le \frac{C\log(TN/\delta)}{K\gamma}\cdot\sqrt{T\log(N/\delta)}.$$

**Lemma 3.17.** *For all $t \in [T]$, it holds that $\tilde{L}^{(t)} - \hat{L}^{(t)} = \gamma\sum_{a=1}^{N}\hat{L}_a^{(t)}$.*

**Step 2: putting it all together.** Using Lemmas 3.15 to 3.17, we may now bound $\hat{R}_a^{(t)}$ using (21) as follows: with probability at least $1-2\delta$, we have that for all $a \in [N]$,

$$\begin{aligned}R_a^{(T)} \le& \frac{\log N}{\eta} + 2\eta T + \frac{2\eta T\log^2(TN/\delta)}{(K\gamma)^2} + \frac{C\log(TN/\delta)}{K\gamma}\cdot\sqrt{T\log(N/\delta)} + \gamma\sum_{a=1}^{N}\hat{L}_a^{(T)}\\ \le& \frac{\log N}{\eta} + 2\eta T + \frac{2\eta T\log^2(TN/\delta)}{(K\gamma)^2} + \frac{C\log(TN/\delta)}{K\gamma}\cdot\sqrt{T\log(N/\delta)} + \frac{C\log(TN/\delta)\sqrt{T\log(N/\delta)}}{K} + \gamma NT,\end{aligned}$$

where the final inequality uses Lemma 3.16 again and the fact that $L_a^{(T)} \le T$ for all $a \in [N]$. By our choices of $\eta = 1/\sqrt{T}$ and $\gamma = \frac{1}{KT^{1/6}}$, we obtain that, with probability $1-2\delta$,

$$R_a^{(t)} \le C\log^2(TN/\delta)\cdot T^{5/6} + T^{5/6}\cdot\frac{N}{K}.$$

The proof is completed by noting that $R_a^{(t)} = \frac{1}{K}\cdot\sum_{t=1}^{T}\sum_{k=1}^{K^{(t)}}\mathbf{u}_k^{(t)}[a] - \mathbf{u}_k^{(t)}[a_k^{(t)}]$. ∎

### 3.5.1 Proofs of lemmas

In this section, we prove Lemmas 3.15 to 3.17. We begin with the following lemma establishing concentration for the $\hat{Y}_a^{(t)}$ values in each round $t$.

**Lemma 3.18.** *Fix $\delta \in (0,1)$. Then with probability at least $1-\delta$, for all $s \in [T]$ and $a \in [N]$,*

$$\hat{Y}_a^{(s)} - \mathbf{y}^{(s)}[a] \le \frac{\log(TN/\delta)}{2K\gamma}.$$

*Proof.* For $s \in [T]$ and $k \in [K^{(s)}]$, let $\mathcal{F}^{(s,k)}$ denote the sigma-algebra generated by all random variables prior to the beginning of step $k$ of round $s$ in `Exp3Multi` (Algorithm 5; so in particular, $\mathbf{p}^{(s)}$ is $\mathcal{F}^{(s,1)}$-measurable). Consider any fixed $s \in [T]$ and $a \in [N]$. Then for each $k \in [K^{(s)}]$,

$$\mathbb{E}\left[\frac{\mathbb{1}\{a_k^{(s)} = a\}\cdot y_k^{(s)}}{\mathbf{p}^{(s)}[a]} \mid \mathcal{F}^{(s,k)}\right] = \frac{\mathbf{p}^{(s)}[a]\cdot\mathbf{y}^{(s)}[a]}{\mathbf{p}^{(s)}[a]} = \mathbf{y}^{(s)}[a]. \tag{22}$$

We next apply Lemma 3.19 with $T = K^{(s)}$ to the sequence of random variables $\left(\frac{\mathbb{1}\{a_k^{(s)}=a\}\cdot\mathbf{y}_k^{(s)}[a]}{\mathbf{p}^{(s)}[a]}\right)_{k=1}^{K^{(s)}}$. The values of the parameters are defined as follows: all values of $y^{(t)}$ are set to $\mathbf{y}^{(s)}[a]$, all parameters $\lambda^{(t)}$ are set to $\frac{\gamma}{\mathbf{p}^{(s)}[a]}$, and all parameters $\alpha^{(t)}$ are set to $2\gamma$. The precondition is satisfied by

(22). Then under some event $\mathcal{E}_{a,s}$ that occurs with probability $1-\delta/(NT)$,

$$\sum_{k=1}^{K^{(s)}}(\hat{Y}_a^{(s,k)}-\mathbf{y}_k^{(s)}[a]) = \sum_{k=1}^{K^{(s)}}\left(\frac{1}{1+\frac{\gamma}{\mathbf{p}^{(s)}[a]}}\cdot\frac{\mathbb{1}\{a_k^{(s)}=a\}\cdot\mathbf{y}_k^{(s)}[a]}{\mathbf{p}^{(s)}[a]}-\mathbf{y}_k^{(s)}[a]\right)$$
$$\leq \frac{\log(TN/\delta)}{2\gamma}.$$

The lemma statement follows by a union bound over $s$ and $a$, as well as the fact that $\hat{Y}_a^{(s)} = \frac{1}{K}\sum_{k=1}^{K}\hat{Y}_a^{(s,k)}$ and $\mathbf{y}^{(s)}[a] = \frac{1}{K}\sum_{k=1}^{K}\mathbf{y}_k^{(s)}[a]$. □

Next we prove Lemma 3.16, which establishes concentration of the values $\hat{L}_a^{(t)}$ to $L_a^{(t)}$.

*Proof of Lemma 3.16.* Write $R := \frac{\log(TN/\delta)}{2K\gamma}$. We have $\hat{L}_a^{(T)} = \sum_{t=1}^{T}\hat{Y}_a^{(t)}$ and $L_a^{(T)} = \sum_{t=1}^{T}\mathbf{y}^{(t)}[a]$. Let $\mathbb{I}_a^{(t)} := \mathbb{1}\left\{\hat{Y}_a^{(t)}-\mathbf{y}^{(t)}[a]\leq R\right\}$. Note that $\hat{Y}_a^{(t)}\leq K^{(t)}/\gamma$ for each $a\in[N], t\in[T]$. The Azuma-Hoeffding inequality gives that, with probability at least $1-\delta/N$,

$$\begin{aligned}\hat{L}_a^{(t)}-L_a^{(t)} &= \sum_{t=1}^{T}(\hat{Y}_a^{(t)}-\mathbf{y}^{(t)}[a])\\ &\leq \sum_{t=1}^{T}\mathbb{I}_a^{(t)}\cdot(\hat{Y}_a^{(t)}-\mathbf{y}^{(t)}[a]) + \sum_{t=1}^{T}\frac{K^{(t)}}{\gamma}\cdot(1-\mathbb{I}_a^{(t)})\\ &\leq CR\sqrt{T\log(N/\delta)} + \sum_{t=1}^{T}\frac{K^{(t)}}{\gamma}\cdot(1-\mathbb{I}_a^{(t)}),\end{aligned} \tag{23}$$

for a sufficiently large constant $C$. Azuma-Hoeffding is applied in the following manner: let $\mathcal{F}^{(t)}$ denote the sigma-algebra generated by all random variables up to the end of round $t$. Then $\mathbb{I}_a^{(t)}\cdot(\hat{Y}_a^{(t)}-\mathbf{y}^{(t)}[a])\in[-1,R]$ almost surely, and for each $t$,

$$\begin{aligned}\mathbb{E}\left[\mathbb{I}_a^{(t)}\cdot(\hat{Y}_a^{(t)}-\mathbf{y}^{(t)}[a])\mid\mathcal{F}^{(t-1)}\right] &\leq \mathbb{E}\left[\hat{Y}_a^{(t)}-\mathbf{y}^{(t)}[a]\mid\mathcal{F}^{(t-1)}\right]\\ &\leq \frac{1}{K}\sum_{k=1}^{K^{(t)}}\left(\mathbb{E}\left[\frac{\mathbb{1}\{a_k^{(t)}=a\}\cdot\mathbf{y}_k^{(t)}[a]}{\mathbf{p}^{(t)}[a]}\mid\mathcal{F}^{(t-1)}\right]-\mathbf{y}_k^{(t)}[a]\right)=0.\end{aligned}$$

The above verifies that the assumptions of Azuma-Hoeffing are satisfied for the random variables $\mathbb{I}_a^{(t)}\cdot(\hat{Y}_a^{(t)}-\mathbf{y}^{(t)}[a])$ (i.e., they form a supermartingale with respect to the filtration $\mathcal{F}^{(t)}$). Taking a union bound in (23) over all $a\in[N]$ and using the fact that, by Lemma 3.18, $\mathbb{I}_a^{(t)}=1$ for all $a$ and $t$ with probability at least $1-\delta$, we conclude that with probability at least $1-2\delta$, $\max_a(\hat{L}_a^{(t)}-L_a^{(t)})\leq CR\sqrt{T\log(N/\delta)}$, concluding the proof of the lemma. □

*Proof of Lemma 3.17.* We compute

$$\begin{aligned}
\tilde{L}^{(t)} - \hat{L}^{(t)} =& \sum_{s=1}^{t} y^{(s)} - \sum_{s=1}^{t}\sum_{a=1}^{N} \hat{\mathbf{p}}^{(s)}[a] \cdot \hat{Y}_a^{(s)} \\
=& \frac{1}{K}\sum_{s=1}^{t}\left(\sum_{k=1}^{K^{(s)}} y_k^{(s)} - \sum_{a=1}^{N} \hat{\mathbf{p}}^{(s)}[a] \cdot \sum_{k=1}^{K^{(s)}}\left(\frac{\mathbb{1}\{a_k^{(s)} = a\} \cdot y_k^{(s)}}{\mathbf{p}^{(s)}[a] + \gamma}\right)\right) \\
=& \frac{1}{K}\sum_{s=1}^{t}\left(\sum_{k=1}^{K^{(s)}} y_k^{(s)} - \sum_{a=1}^{N}\left(1 - \frac{\gamma}{\mathbf{p}^{(s)}[a] + \gamma}\right) \cdot \sum_{k=1}^{K^{(s)}}\left(\mathbb{1}\{a_k^{(s)} = a\} \cdot y_k^{(s)}\right)\right) \\
=& \frac{\gamma}{K}\sum_{s=1}^{t}\sum_{a=1}^{N}\sum_{k=1}^{K^{(s)}} \frac{\mathbb{1}\{a_k^{(s)} = a\} \cdot y_k^{(s)}}{\mathbf{p}^{(s)}[a] + \gamma} \\
=& \gamma \sum_{s=1}^{t}\sum_{a=1}^{N} \hat{Y}_a^{(s)} = \gamma \sum_{a=1}^{N} \hat{L}_a^{(t)},
\end{aligned}$$

as desired. ∎

Finally, we prove Lemma 3.15, which uses the definition of the exponential weights updates to establish that $\hat{L}^{(t)} - \hat{L}_a^{(t)}$ is bounded for each $a, t$.

*Proof of Lemma 3.15.* For each $a \in [N], t \in [T]$, define $\hat{X}_a^{(t)} = 1 - \hat{Y}_a^{(t)}$, $\hat{S}_a^{(t)} = \sum_{s=1}^{t} \hat{X}_a^{(s)}$, and $\hat{S}^{(t)} = \sum_{s=1}^{t}\sum_{a=1}^{N} \mathbf{p}^{(s)}[a] \cdot \hat{X}_a^{(s)}$. Note that, for all $t \in [T]$ and $a \in [N]$, $\hat{S}_a^{(t)} = t - \hat{L}_a^{(t)}$. Then from Line 13 of Algorithm 5, we have

$$\mathbf{p}^{(t)}[a] = \frac{\exp\left(\eta \hat{S}_a^{(t-1)}\right)}{\sum_{b=1}^{N} \exp\left(\eta \hat{S}_b^{(t-1)}\right)}.$$

Define $W^{(t)} := \sum_{a=1}^{N} \exp(\eta \hat{S}_a^{(t)})$, so that $W^{(0)} = N$. For each $t \in [T]$, we have

$$\begin{aligned}
\frac{W^{(t)}}{W^{(t-1)}} =& \sum_{a=1}^{N} \frac{\exp(\eta \hat{S}_a^{(t-1)}) \cdot \exp(\eta \hat{X}_a^{(t)})}{W^{(t-1)}} \\
=& \sum_{a=1}^{N} \mathbf{p}^{(t)}[a] \cdot \exp(\eta) \cdot \exp(\eta(\hat{X}_a^{(t)} - 1)) \\
\leq& \exp(\eta) \cdot \sum_{a=1}^{N} \mathbf{p}^{(t)}[a] \cdot \left(1 + \eta(\hat{X}_a^{(t)} - 1) + \frac{\eta^2}{2}(\hat{X}_a^{(t)} - 1)^2\right) \\
\leq& \exp(\eta) \cdot \exp\left(\sum_{a=1}^{N} \eta \mathbf{p}^{(t)}[a](\hat{X}_a^{(t)} - 1) + \sum_{a=1}^{N} \frac{\eta^2}{2} \mathbf{p}^{(t)}[a](\hat{X}_a^{(t)} - 1)^2\right) \\
=& \exp\left(\sum_{a=1}^{N} \eta \mathbf{p}^{(t)}[a] X_a^{(t)} + \sum_{a=1}^{N} \frac{\eta^2}{2} \mathbf{p}^{(t)}[a](\hat{X}_a^{(t)} - 1)^2\right)
\end{aligned}$$

where the first inequality uses that $\eta(\hat{X}_a^{(t)} - 1) \le 0$ since $\hat{X}_a^{(t)} = 1 - \hat{Y}_a^{(t)} \le 1$, and the fact that $\exp(x) \le 1 + x + \frac{x^2}{2}$ for $x \le 0$. Then

$$\begin{aligned}\exp\left(\eta \hat{S}_a^{(t)}\right) \le W^{(t)} =& W^{(0)} \cdot \frac{W^{(1)}}{W^{(0)}} \cdots \frac{W^{(t)}}{W^{(t-1)}} \le N \cdot \exp\left(\eta \sum_{s=1}^{t}\sum_{a=1}^{N} \mathbf{p}^{(s)}[a]\hat{X}_a^{(s)} + \eta^2 \sum_{s=1}^{t}\sum_{a=1}^{N} \mathbf{p}^{(s)}[a](\hat{X}_a^{(s)} - 1)^2\right) \\ =& N \cdot \exp\left(\eta \hat{S}^{(t)} + \eta^2 \sum_{s=1}^{t}\sum_{a=1}^{N} \mathbf{p}^{(s)}[a](\hat{X}_a^{(s)} - 1)^2\right).\end{aligned}$$

Taking the logarithm and rearranging gives that

$$\hat{L}^{(t)} - \hat{L}_a^{(t)} = \hat{S}_a^{(t)} - \hat{S}^{(t)} \le \frac{\log N}{\eta} + \eta \sum_{s=1}^{t}\sum_{a=1}^{N} \mathbf{p}^{(s)}[a](\hat{Y}_a^{(s)})^2. \tag{24}$$

Note that $\hat{Y}_a^{(s,k)} \ge 0$ for all $a, s, k$ and thus $\hat{Y}_a^{(s)} \ge 0$ for all $a, s$. Then under the event $\mathcal{E}$ of Lemma 3.18, which occurs with probability at least $1 - \delta$, for all $s \in [T]$ and $a \in [N]$,

$$\mathbf{p}^{(s)}[a] \cdot (\hat{Y}_a^{(s)})^2 \le \mathbf{p}^{(s)}[a] \cdot \left(2(\mathbf{y}^{(s)}[a])^2 + \frac{2\log^2(TN/\delta)}{(K\gamma)^2}\right).$$

Using that $\mathbf{y}^{(s)}[a] \in [0, 1]$ for all $s, a$ and combining the above display with (24) yields that, under $\mathcal{E}$,

$$\hat{L}^{(t)} - \hat{L}_a^{(t)} \le \frac{\log N}{\eta} + \eta \sum_{s=1}^{t}\sum_{a=1}^{N} \mathbf{p}^{(s)}[a] \cdot \left(2 + \frac{2\log^2(TN/\delta)}{(K\gamma)^2}\right) \le \frac{\log N}{\eta} + 2\eta t + \frac{2\eta t \log^2(TN/\delta)}{(K\gamma)^2},$$

as desired. □

The following lemma establishes concentration of a martingale with potentially heavy tails.

**Lemma 3.19** (Lemma 12.2 of [LS20]). *Let $(y^{(t)})_{t=1}^T$ denote a fixed sequence of real numbers, let $(\mathcal{F}^{(t)})_{t=1}^T$ be a filtration, and let $\tilde{Y}^{(t)}$ be a real-valued sequence adapted to the filtration $\mathcal{F}^{(t)}$. Suppose that $\mathbb{E}[\tilde{Y}^{(t)} \mid \mathcal{F}^{(t-1)}] = y^{(t)}$ for all $T \in [T]$.*

*Moreover, let $(\alpha^{(t)})_{t\in[T]}$ and $(\lambda^{(t)})_{t\in[T]}$ be real-valued $\mathcal{F}^{(t)}$-predictable sequences of random variables so that for all $a, t$, it holds that $0 \le \alpha^{(t)}\tilde{Y}^{(t)} \le 2\lambda^{(t)}$. Then for all $\delta \in (0, 1)$,*

$$\mathbb{P}\left(\sum_{t=1}^{T} \alpha^{(t)} \cdot \left(\frac{\tilde{Y}^{(t)}}{1 + \lambda^{(t)}} - y^{(t)}\right) \ge \log(1/\delta)\right) \le \delta.$$

# 4 An oblivious lower bound on swap regret

In this section, we make a slight adjustment to notation, considering an adversary that selects, at each time step $t$, a reward vector $\mathbf{u}^{(t)} \in [0,1]^N$ as opposed to a reward function $\mathbf{f}^{(t)} : [N] \to [0,1]$. These are functionally identical, and it's a change we make only to match the notational choice of classic results in this space.

**Theorem 4.1.** *Given $T$ rounds of online learning over $N$ actions, there exists a randomized oblivious adversarial strategy $\mathcal{D} \in \Delta\left([0,1]_N^T\right)$ that forces all learners to incur $\tilde{\Omega}\left(\min\left\{1, \sqrt{N/T}\right\}\right)$ swap regret, in expectation over the sampled adversarial actions $\mathbf{u}^{(1:T)} \sim \mathcal{D}$. Any learner strategy in this setting can be viewed as a collection of maps that, at each time step $t$, maps the observed $\mathbf{u}^{(1:t-1)}$ to an action. That is, for all $t \in [T]$, and all $\mathbf{x}^{(t)} : [0,1]_N^{t-1} \to \Delta_N$, we have*

$$\mathbb{E}_{\mathbf{u}^{(1:T)} \sim \mathcal{D}}\left[\mathbf{SwapRegret}\left(\mathbf{x}^{(t \in 1:T)}\left(\mathbf{u}^{(1:t-1)}\right), \mathbf{u}^{(1:T)}\right)\right] = \Omega\left(\min\left(\frac{1}{\log^5(T)}, \frac{\sqrt{N/T}}{\log^5(N)}\right)\right)$$

## 4.1 Construction of the adversary

The adversary will produce rewards in $2^D$ batches of size $B$. Each batch of rewards will correspond to a root-to-leaf path in a complete binary tree as follows. Consider a complete binary tree containing $2^D$ leaf nodes. That is, a tree of depth $D$, having $2^{D+1} - 1$ total nodes. We will associate each node $v$ of this tree with 2 unique actions: $a_v, \dot{a}_v \in [N]$. This necessitates $2(2^{D+1} - 1) \leq N$. We label the children of each internal node as 'left child' and 'right child'. This will yield an enumeration of the leaves of the tree (according to the natural DFS from the root, which explores the left child first). According to this enumeration, we denote the leaves by $\ell_1, \cdots, \ell_{2^D}$. In each batch $b \in [2^D]$, the rewards $\mathbf{u}^{(t)}$ will be supported on actions $a_v, \dot{a}_v$ with $v$ on the root-to-$\ell_b$ path. We denote this path by $P_b = (v_{(b,0)} - v_{(b,1)} - \cdots - v_{(b,D)})$, where $v_{(b,0)}$ is the root and $v_{(b,D)}$ is the leaf $\ell_b$. The reward of all other actions will be 0. In fact, for each internal node $v \in P_b$, $v \neq \ell_b$, only one of the two actions $a_v, \dot{a}_v$ will have non-zero reward during the batch. We denote this rewarded action $\mathring{a}_v(b) \in \{a_v, \dot{a}_v\}$, and it will be determined in terms of a Bernoulli random variable $r_v$ as follows:

$$\mathring{a}_v(b) = \begin{cases} a_v & \text{if } \ell_b \text{ is a left descendant of } v \\ a_v & \text{if } \ell_b \text{ is a right descendant of } v \text{ and } r_v = 0 \\ \dot{a}_v & \text{if } \ell_b \text{ is a right descendant of } v \text{ and } r_v = 1 \end{cases}$$

where $\Pr[r_v = 1] = 1/2$ i.i.d. for all internal nodes $v$. Informally, the rewarded action will be $a_v$ throughout all batches $b$ where the leaf $\ell_b$ is a left descendant of $v$, and it will switch to $\dot{a}_v$ with probability $1/2$ once $\ell_b$ becomes a right descendant of $v$.

Similarly for the leaves, at each time step $t$, only one of the actions $\mathring{a}_{\ell_b}(t) \in \{a_{\ell_b}, \dot{a}_{\ell_b}\}$ receives reward. However, unlike the internal nodes, this rewarded action is chosen independently at each time step $t$, rather than being constant for the entirety of batch $b$. $\mathring{a}_{\ell_b}(t)$ is determined by a Bernoulli random variable $r_{(\ell_b,t)}$ with $\Pr[r_{(\ell_b,t)} = 1] = 1/2$ i.i.d. as follows

$$\mathring{a}_{\ell_b}(t) = \begin{cases} a_{\ell_b} & \text{if } r_{(\ell_b,t)} = 0 \\ \dot{a}_{\ell_b} & \text{if } r_{(\ell_b,t)} = 1 \end{cases}$$

All of these Bernoulli random variables will be sampled before the first round of learning; their values will initially be unknown to the learner; and they will be the only source of randomness the adversary relies on.

Now, we are ready to define the rewards chosen by the adversary. For all batches $b$, for each time step $t$ in batch $b$, we have

$$\begin{aligned} \mathbf{u}^{(t)}[\mathring{a}_{v_{(b,d)}}(b)] &= \frac{d}{2D} && \text{for } d < D \text{ (internal nodes in } P_b) \\ \mathbf{u}^{(t)}[\mathring{a}_{\ell_b}(t)] &= 1 && \text{(the leaf of } P_b) \\ \mathbf{u}^{(t)}[a] &= 0 && \text{(for all other actions)} \end{aligned} \tag{25}$$

Lastly, we must have $4 \cdot 2^D - 2 \leq N$ and also $2^D \leq B \cdot 2^D \leq T$. So, we choose

$$D = \left\lfloor \log_2 \min\left(T, \frac{N+2}{4}\right) \right\rfloor \qquad\qquad B = \left\lfloor \frac{T}{2^D} \right\rfloor$$

Note, $B = 1$ for $T \leq 4N - 2$. Additionally, for $t > T' = B2^D$, we simply have $\mathbf{u}^{(t)} = \mathbf{0}$. Note, $T' \geq \frac{B}{B+1}T \geq T/2$, so not utilizing these rounds will only impact the swap regret by a constant factor.

Also, importantly, at every single time step $t \leq T'$, $\left\|\mathbf{u}^{(t)}\right\|_1 = \sum_{d=1}^{D-1} \frac{d}{2D} + 1 = \frac{D+3}{4}$. Running the same algorithm scaling each reward by $\frac{4}{D+3}$ gives an algorithm for which $\left\|\mathbf{u}^{(t)}\right\|_1 \leq 1$ for all $t$ that achieves the following:

**Corollary 4.2.** *Given $T$ rounds of online learning over $N$ actions, there exists a randomized oblivious adversarial strategy $\mathcal{D} \in \Delta\left(S_N^T\right)$ where $S_N = \left\{\mathbf{u} \in [0,1]^N \middle| \|\mathbf{u}\|_1 \leq 1\right\}$ that forces all learners to incur $\tilde{\Omega}\left(\min\left\{1, \sqrt{N/T}\right\}\right)$ swap regret, in expectation over the sampled adversarial actions $\mathbf{u}^{(1:T)} \sim \mathcal{D}$. That is, for all $t \in [T]$, $\mathbf{x}^{(t)} : S_N^{t-1} \to \Delta_N$, we have*

$$\mathbb{E}_{\mathbf{u}^{(1:T)} \sim \mathcal{D}}\left[\textbf{SwapRegret}\left(\mathbf{x}^{(t \in 1:T)}\left(\mathbf{u}^{(1:t-1)}\right), \mathbf{u}^{(1:T)}\right)\right] = \Omega\left(\min\left(\frac{1}{\log^6(T)}, \frac{\sqrt{N/T}}{\log^6(N)}\right)\right)$$

## 4.2 Proof of Theorem 4.1

### 4.2.1 Outline

We introduce the following notation for the time steps of learning. For a node $v$, define $\underline{t}_v$ to be the first time step of the first batch in which $v$ appears on the root-to-$\ell_b$ path. If that is batch $b$, then $\underline{t}_v = (b-1)B + 1$. Similarly, let $\bar{t}_v$ be the last time step of the last batch in which $v$ appears on the root-to-$\ell_b$ path. If that is batch $b$, then $\bar{t}_v = bB$.

We also introduce the following string-concatenation notation to index the nodes of the tree. For a non-leaf node $v$, let $vL$ and $vR$ denote the left and right child of $v$ respectively. This enables us to index grandchildren as such: $vLL, vLR, vRL, vRR$. Under this notation, we also refer to the root by the empty string: $\emptyset$. As a sanity check, realize $\underline{t}_v = \underline{t}_{vL}$ and $\bar{t}_v = \bar{t}_{vR}$. Also observe that, for $t \leq \bar{t}_{vL}$, the reward $\mathbf{u}^{(t)}$ is independent from Bernoulli random variable $r_v$. $\bar{t}_{vL}$ is the final time step before we switch to a batch $b$ where $\ell_b$ is a right descendant of $v$, and $\mathring{a}_v(b)$ is dependent on $r_v$. At every time step $t$, the learner selects $\mathbf{x}^{(t)}$ as a function of the observed rewards $\mathbf{u}^{(1:t-1)}$.

The rewards $\mathbf{u}^{(1:t-1)}$ only depend on the set of Bernoulli random variables that have been observed before time $t$. The set $r^{(t)}$ contains all these variables, defined formally as:

$$r^{(t)} = \{r_v | \text{ non-leaf } v \text{ s.t. } \bar{t}_{vL} < t\} \cup \left\{r_{(\ell,\tau)} \middle| \text{ leaves } \ell \text{ and time } \tau < t\right\}$$

Fixing the strategy of the learner, for every $t$, we can view the learner's action $\mathbf{x}^{(t)}_{r^{(t)}} = \mathbf{x}^{(t)}(\mathbf{u}^{(1:t-1)}(r^{(t)}))$ as a function of the random variables $r^{(t)}$. This function is deterministic since there is no reason for the learner to randomize against an oblivious adversary.

We have

$$\mathbb{E}_{r^{(T)}}\left[\textbf{SwapRegret}\left(\mathbf{x}^{(t\in 1:T)}_{r^{(t)}}, \mathbf{u}^{(1:T)}\right)\right] = \frac{1}{T}\sum_{a\in[N]} \mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]} \text{Swap}(a \to a')\right]$$

where

$$\text{Swap}(a \to a') = \sum_{t=1}^{T} \mathbf{x}^{(t)}_{r^{(t)}}[a]\left(\mathbf{u}^{(t)}[a'] - \mathbf{u}^{(t)}[a]\right) \ .$$

For each action $a \in [N]$, we establish a lower bound on $\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]} \text{Swap}(a \to a')\right]$ in terms of the amount of time that $a$ is played in rounds $[1 : T']$ (recall $T' = 2^D B$ is the last iteration where the adversary gives rewards). We split into 6 cases:

1. $a = a_v$ where $\text{depth}(v) \le D - 2$ (Lemma A.1)
2. $a = \dot{a}_v$ where $\text{depth}(v) \le D - 2$ (Lemma A.2)
3. $a = a_v$ where $\text{depth}(v) = D - 1$ (Lemma A.3)
4. $a = \dot{a}_v$ where $\text{depth}(v) = D - 1$ (Lemma A.4)
5. $a = a_v$ or $a = \dot{a}_v$ where $\text{depth}(v) = D$ (Lemma A.5)
6. $a \in [N]$ such that $a \neq a_v, \dot{a}_v$ for all nodes $v$ in the tree — these actions receive no reward (Lemma A.7)

For cases 1,2,3,4, and 6, we prove

$$\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]} \text{Swap}(a \to a')\right] \ge \Omega\left(\frac{1}{D^5}\right) \mathbb{E}_{r^{(T)}}\left[\sum_{t=1}^{T'} \mathbf{x}^{(t)}_{r^{(t)}}[a]\right] \tag{26}$$

For case 5 where $v$ is a leaf, we prove

$$\begin{aligned}\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]} \text{Swap}(a_v \to a')\right] + \mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]} \text{Swap}(\dot{a}_v \to a')\right] \\ \ge \Omega\left(\frac{1}{D^4\sqrt{B}}\right)\left(\mathbb{E}_{r^{(T)}}\left[\sum_{t=1}^{T'} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right] - \frac{B}{2}\right)\end{aligned} \tag{27}$$

Summing these bounds over all $a \in [N]$ gives

$$\mathbb{E}_{r^{(T)}}\left[\textbf{SwapRegret}\left(\mathbf{x}^{(t\in 1:T)}_{r^{(t)}}, \mathbf{u}^{(1:T)}\right)\right] \ge \Omega\left(\frac{1}{D^5\sqrt{BT}}\right)\left(\mathbb{E}_{r^{(T)}}\left[\sum_{t=1}^{T'}\sum_{a\in[N]} \mathbf{x}^{(t)}_{r^{(t)}}[a]\right] - (2^D)\frac{B}{2}\right)$$

subtracting a $B/2$ term for each of the $2^D$ leaves

$$= \Omega\left(\frac{1}{D^5\sqrt{B}T}\right)(T'/2)$$

since $\sum_{a\in[N]} \mathbf{x}^{(t)}_{r^{(t)}}[a] = 1$ in each iteration

$$= \Omega\left(\frac{1}{D^5\sqrt{B}}\right) = \Omega\left(\min\left(\frac{1}{\log^5(T)}, \frac{\sqrt{N/T}}{\log^5(N)}\right)\right)$$

since $T' \geq T/2$, as desired. In Appendix A, we establish lemmas proving equations (26) and (27) for each of these 6 cases respectively. Below, we will outline the proof for these cases, while getting into more detail in only some of them.

#### 4.2.2 Case 1: $v$ is an internal node and $a = a_v$

Let's consider the first case where $a = a_v$ with $v$ an internal node. The interval of time over which this action is potentially rewarded, $[\underline{t}_v, \bar{t}_v]$, can be broken into 4 intervals: $[\underline{t}_{vLL}, \bar{t}_{vLL}]$, $[\underline{t}_{vLR}, \bar{t}_{vLR}]$, $[\underline{t}_{vRL}, \bar{t}_{vRL}]$ and $[\underline{t}_{vRR}, \bar{t}_{vRR}]$, also referred to as first interval, second interval etc. Denote by $z_1$ the total weight put on $a_v$ during the first interval (a random variable in $r^{(t)}$):

$$z_1 = \sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v].$$

Similarly, denote by $z_2, z_3$ and $z_4$ the weight in the second, third and fourth intervals. Additionally, denote by $z_5$ the weight on $a_v$ outside of these intervals, namely, when $v \notin P_b$:

$$z_5 = \sum_{t=1}^{\underline{t}_v - 1} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \sum_{t=\bar{t}_v+1}^{T} \mathbf{x}^{(t)}_{r^{(t)}}[a_v].$$

We will prove the following claims:

- The regret for swapping $a_v$ to $a_{vL}$ will be large, if $a_v$ is played a lot during the first interval:

$$\mathbb{E}\left[\max_{a'\in[N]} \text{Swap}(a_v \to a')\right] \geq \mathbb{E}\left[\max\left(\text{Swap}(a_v \to a_{vL}), 0\right)\right] \geq \mathbb{E}\left[\frac{z_1}{C_1' D}\right] \quad , \tag{28}$$

  where $C_1' > 0$ is a bounded constant. Notice that the first inequality is due to that fact that the regret for swapping from $a_v$ to itself is always zero. The second inequality is elaborated below.

- The regret for swapping to $a_{vL}$ or $\dot{a}_{vL}$ is large if $a_v$ is played a lot during the second quarter but not during the first quarter:

$$\begin{aligned}\mathbb{E}\left[\max_{a'\in[N]} \text{Swap}(a_v \to a')\right] &\geq \mathbb{E}\left[\max\left(\text{Swap}(a_v \to a_{vL}), \text{Swap}(a_v \to \dot{a}_{vL}), 0\right)\right] \\ &\geq \mathbb{E}\left[\frac{z_2}{C_2' D} - C_2 z_1\right] \quad ,\end{aligned} \tag{29}$$

  where $C_2$ and $C_2'$ are bounded constants (and similarly for the constants in the other cases).

- Similarly, we lower bound the swap regret for swapping to $a_{vR}$:

$$\mathbb{E}\left[\max_{a'\in[N]} \text{Swap}(a_v \to a')\right] \geq \mathbb{E}\left[\max\left(\text{Swap}(a_v \to a_{vR}), 0\right)\right] \geq \mathbb{E}\left[\frac{z_3}{C_3' D} - C_3(z_1+z_2)\right] \quad , \quad (30)$$

- The swap regret to $\dot{a}_{vR}$:

$$\begin{aligned}\mathbb{E}\left[\max_{a'\in[N]} \text{Swap}(a_v \to a')\right] &\geq \mathbb{E}\left[\max\left(\text{Swap}(a_v \to a_{vR}), \text{Swap}(a_v \to \dot{a}_{vR}), 0\right)\right] \quad (31)\\ &\geq \mathbb{E}\left[\frac{z_4}{C_4' D} - C_4(z_1+z_2+z_3)\right] \quad ,\end{aligned}$$

- Lastly, the swap regret to the root, denoted here as root:

$$\begin{aligned}\mathbb{E}\left[\max_{a'\in[N]} \text{Swap}(a_v \to a')\right] &\geq \mathbb{E}\left[\max\left(\text{Swap}(a_v \to a_{\text{root}}), \text{Swap}(a_v \to \dot{a}_{\text{root}}), 0\right)\right] \quad (32)\\ &\geq \mathbb{E}\left[\frac{z_5}{C_5' D} - C_5(z_1+z_2+z_3+z_4)\right] \quad ,\end{aligned}$$

Next, we add the above equations with appropriate coefficients:

$$\begin{aligned}&(28) + \frac{1}{5C_1' C_2 D}(29) + \frac{1}{5C_1' C_2 C_2' C_3 D^2}(30) + \frac{1}{5C_1' C_2 C_2' C_3 C_3' C_4 D^3}(31)\\ &\quad + \frac{1}{5C_1' C_2 C_2' C_3 C_3' C_4 C_4' C_5 D^4}(32)\end{aligned}$$

The left hand side will be a constant times the maximal swap regret from $a_v$ to any other action. The right hand side will be $\Omega(\mathbb{E}\left[z_1 + \cdots + z_5\right]/D^5)$. This yields Eq. (26) as desired. Below, we outline the first two inequalities, Eq. (28) and Eq. (29), in more depth, and the other inequalities will be discussed more lightly.

**Proving Eq. (28).** To compute $\text{Swap}(a_v \to a_{vL})$, we need to understand how much reward is given to each of $a_v$ and $a_{vL}$, as a function $r^{(T)}$, in each of the intervals $[\underline{t}_{vLL}, \bar{t}_{vLL}]$, $[\underline{t}_{vLR}, \bar{t}_{vLR}]$, $[\underline{t}_{vRL}, \bar{t}_{vRL}]$ and $[\underline{t}_{vRR}, \bar{t}_{vRR}]$. Assume that $v$ is of depth $d$, and notice:

- In the first interval, the reward of action $a_v$ is $d/(2D)$, while the reward of $a_{vL}$ is $(d+1)/2D$.
- In the second interval, if $r_{vL} = 0$ then the reward of $a_v$ is $d/(2D)$ and the reward of $a_{vL}$ is $(d+1)/2D$. We will not care about what happens when $r_{vL} = 1$.
- In the third and fourth intervals, if $r_v = 1$ then both actions get a reward of zero. We will not care what happens when $r_v = 0$.
- In iterations outside these intervals, when $v, vL \notin P_b$, both actions receive a reward of zero.

This implies that if $r_{vL} = 0$ and $r_v = 1$ then the total reward for action $r_v$ over all the $T$ rounds is $\frac{d}{2D}(z_1+z_2)$, since $z_1+z_2$ is the total weight put on action $a_v$ in these two intervals. The reward that would be obtained from playing $a_{vL}$ instead is $\frac{d+1}{2D}(z_1+z_2)$. Therefore, if $r_{vL} = 1$ and $r_v = 0$:

$$\text{Swap}(a_v \to a_{vL}) \geq \frac{d+1}{2D}(z_1+z_2) - \frac{d}{2D}(z_1+z_2) = \frac{z_1+z_2}{2D} \geq \frac{z_1}{2D}.$$

Next, we use the fact that $z_1$ is independent from $r_{vL}$ and $r_v$, since $z_1$ is the weight on action $a_v$ in the first interval, before the learner observes $r_v$ and $r_{vL}$. We derive that:

$$\begin{aligned}
\mathbb{E}_{r^{(T)}}\left[\max\left(\mathrm{Swap}(a_v \to a_{vL}), 0\right)\right] &\geq \mathbb{E}_{r^{(T)}}\left[\max\left(\mathrm{Swap}(a_v \to a_{vL}), 0\right) \mathbb{1}\left[r_v = 1, r_{vL} = 0\right]\right] \\
&\geq \mathbb{E}_{r^{(T)}}\left[\mathrm{Swap}(a_v \to a_{vL}) \mathbb{1}\left[r_v = 1, r_{vL} = 0\right]\right] \\
&\geq \mathbb{E}_{r^{(T)}}\left[\frac{z_1}{2D} \mathbb{1}\left[r_v = 1, r_{vL} = 0\right]\right] \\
&\geq \mathbb{E}_{r^{(T)}}\left[\frac{z_1}{2D}\right] \mathbb{E}\left[\mathbb{1}\left[r_v = 1, r_{vL} = 0\right]\right] \\
&\geq \frac{\mathbb{E}[z_1]}{8D}.
\end{aligned}$$

This is what we wanted to prove.

**Proving Equation** (29) We divide into cases according to $r_{vL}$. As before, we will condition on $r_v = 1$. First, if $r_{vL} = 0$, then, as argued in the proof of Eq. (28),

$$\mathrm{Swap}(a_v \to a_{vL}) \geq \frac{z_1 + z_2}{2D} \geq \frac{z_2}{2D}.$$

For the case that $r_{vL} = 1$, we lower bound the swap regret to $\dot{a}_{vL}$. We notice that:

- In the first interval, the reward of $a_v$ is $d/(2D)$ and the reward of $\dot{a}_{vL} = 0$.
- In the second interval, the reward of $a_v$ is still $d/(2D)$ while the reward of $\dot{a}_{vL}$ is $(d+1)/2D$.
- In the remaining iterations, both $a_v$ and $\dot{a}_{vL}$ have reward of 0.

Consequently, the reward of $a_v$ is $\frac{d}{2D}(z_1 + z_2)$, whereas the reward of swapping $a_v$ to $\dot{a}_{vL}$ is $\frac{d+1}{2d} z_2$. Hence,

$$\mathrm{Swap}(a_v \to \dot{a}_{vL}) \geq \frac{z_2}{2D} - \frac{dz_1}{2D}.$$

By combining the two cases above, we obtain that whenever $a_v = 1$,

$$\max_{a' \in [N]} \mathrm{Swap}(a_v \to a') \geq \frac{z_2}{2D} - \frac{dz_1}{2D}\ .$$

Hence, we obtain that

$$\begin{aligned}
\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \mathrm{Swap}(a_v \to a')\right] &\geq \mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \mathrm{Swap}(a_v \to a') \mathbb{1}[r_v = 1]\right] \\
&\geq \mathbb{E}_{r^{(T)}}\left[\left(\frac{z_2}{2D} - \frac{dz_1}{2D}\right) \mathbb{1}[r_v = 1]\right] \\
&= \mathbb{E}_{r^{(T)}}\left[\left(\frac{z_2}{2D} - \frac{dz_1}{2D}\right)\right] \mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_v = 1]\right] \\
&= \frac{1}{4D} \mathbb{E}_{r^{(T)}}\left[z_2 - dz_1\right],
\end{aligned}$$

again, using independence of $r_v$ with $z_1$ and $z_2$. This is what we wanted to prove.

**Proving Equations (30) to (32):** Eq. (30) and Eq. (31) are proved similarly to Eq. (28) and Eq. (29), with the difference that now, we also need to consider the case that $r_v = 0$. Eq. (32) is also proved similarly. Here, we consider weight put on $a_v$ during iterations where $a_v$ is not rewarded at all, and we bound the regret of swapping to the root.

#### 4.2.3 Cases 2,3,4,6: other cases where $v$ is not a leaf

For the case that $v$ is an internal node of depth at most $D-2$, and $a = \dot{a}_v$, the proof is very similar to the case that $a = a_v$. Similarly, the cases where $v$ is an internal node of depth $D-1$ and $a = a_v, \dot{a}_v$ are very similar to the cases of depth at most $D-2$. They are separate cases simply because the child nodes in this case are leaves, altering the equations. For the case when $a$ is associated with no nodes at all and receives 0 reward, we obtain our lower bound by simply considering the swap to the root.

#### 4.2.4 Case 5: $v$ is a leaf and $a = a_v$ or $\dot{a}_v$

For Case 5, when $v = \ell_b$ is a leaf of the tree, there are Bernoulli random variables $r_{(v,t)}$ for every time step $t$ in $[\underline{t}_v, \bar{t}_v]$. By definition,

$$\mathbf{u}^{(t)}[a_v] = \mathbb{1}\left[r_{(v,t)} = 0\right] \qquad\qquad \mathbf{u}^{(t)}[\dot{a}_v] = \mathbb{1}\left[r_{(v,t)} = 1\right]$$

for $t \in [\underline{t}_v, \bar{t}_v]$. We have,

$$\begin{aligned}
&\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \mathrm{Swap}(a_v \to a')\right] + \mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \mathrm{Swap}(\dot{a}_v \to a')\right] \\
&\geq \mathbb{E}_{r^{(T)}}\left[\max\left\{\sum_{t=\underline{t}_v}^{\bar{t}_v}\left(\mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right)\mathbb{1}\left[r_{(v,t)} = 0\right], \sum_{t=\underline{t}_v}^{\bar{t}_v}\left(\mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right)\mathbb{1}\left[r_{(v,t)} = 1\right]\right\}\right] \\
&- \frac{1}{2}\mathbb{E}_{r^{(T)}}\left[\sum_{t=\underline{t}_v}^{\bar{t}_v}\left(\mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right)\right] \\
&= \frac{1}{2}\mathbb{E}_{r^{(T)}}\left[\left|\sum_{t=\underline{t}_v}^{\bar{t}_v}\left(\mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right)Z^{(t)}\right|\right] \qquad \text{where } Z^{(t)} = \begin{cases} 1 & \text{if } r_{(v,t)} = 1 \\ -1 & \text{if } r_{(v,t)} = 0 \end{cases}
\end{aligned}$$

If we denote by $X^{(t)} = \sum_{\tau=\underline{t}_v}^{t}\left(\mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right)Z^{(t)}$, then $X^{(\underline{t}_v)}, \cdots, X^{(\bar{t}_v)}$ is a martingale due to the independence of $\mathbf{x}^{(t)}_{r^{(t)}}[a_v], \mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]$ and $Z^{(t)}$ for all $t$. We use the following lemma which applies Azuma's inequality,

**Lemma 4.3.** *Consider an algorithm which, at any round $t = 1, \ldots, B$, selects a parameter $x_t \in [0,1]$, possibly at random. Then, it observes the outcome of a random variable $Z_t \sim \mathrm{Uniform}(\{-1,1\})$. Assume that $\mathbb{E}\left[\sum_{t=1}^{B} x_t^2\right] \geq \epsilon B$ for some $\epsilon > 0$. Then,*

$$\mathbb{E}\left[\left|\sum_{t=1}^{B} x_t Z_t\right|\right] \geq \frac{\epsilon\sqrt{B}}{4\sqrt{\log(1/\epsilon)}}\,.$$

Using this lemma, we establish

$$\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\mathrm{Swap}(a_v\to a')\right]+\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\mathrm{Swap}(\dot{a}_v\to a')\right]$$
$$\geq\frac{1}{1024D^3\sqrt{B}}\left(\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]+\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right]\right)-\frac{\sqrt{B}}{8192D^4}$$

# A Proofs remaining for Theorem 4.1

**Lemma A.1** (Case 1)**.** *Let $v$ be a node in the tree of depth $d\leq D-2$. Then,*

$$184D^5\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\mathrm{Swap}(a_v\to a')\right]\geq\sum_{t=1}^{T'}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]\tag{33}$$

*Proof of Lemma A.1.* We have

$$\begin{aligned}\mathrm{Swap}(a_v\to a_{vL})&=\sum_{t=1}^{T}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\left(\mathbf{u}^{(t)}[a_{vL}]-\mathbf{u}^{(t)}[a_v]\right)\\&=\frac{d+1}{2D}\left(\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]+\mathbb{1}\left[r_{vL}=0\right]\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right)\\&\quad-\frac{d}{2D}\left(\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]+\mathbb{1}\left[r_v=0\right]\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right)\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbb{1}[r_{vL}=0\wedge r_v=1]\cdot\mathrm{Swap}(a_v\to a_{vL})&=\frac{1}{2D}\mathbb{1}[r_{vL}=0\wedge r_v=1]\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\\&=\frac{1}{2D}\mathbb{1}[r_{vL}=0\wedge r_v=1]\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]&(34)\\&\quad+\frac{1}{2D}\mathbb{1}[r_v=1]\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=0]&(35)\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_{vL}=0\wedge r_v=1]\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]&=\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_{vL}=0\wedge r_v=1]\right]\mathbb{E}_{r^{(T)}}\left[\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]\\&=\frac{1}{4}\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]&(36)\end{aligned}$$

due to the independence of $r_{vL}, r_v$ from $r^{(t)}$ for $t \leq \bar{t}_{vLL}$. And,

$$\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_v = 1] \sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL} = 0]\right] = \mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_v = 1]\right]\mathbb{E}_{r^{(T)}}\left[\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL} = 0]\right]$$
$$= \frac{1}{2}\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL} = 0]\right] \tag{37}$$

due to the independence of $r_v$ from $r_{vL}, r^{(t)}$ for $t \leq \bar{t}_{vLR}$. So, combining equations (34), (35), (36), and (37), we have

$$\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_{vL} = 0 \wedge r_v = 1] \cdot \mathrm{Swap}(a_v \to a_{vL})\right]$$
$$= \frac{1}{8D}\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] + \frac{1}{4D}\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL} = 0]\right]$$

Lastly, since $\mathrm{Swap}(a_v \to a_v) = 0$,

$$\begin{aligned}\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \mathrm{Swap}(a_v \to a')\right] &\geq \mathbb{E}_{r^{(T)}}\left[\max\left\{\mathrm{Swap}(a_v \to a_{vL}), 0\right\}\right] \\ &\geq \mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_{vL} = 0 \wedge r_v = 1] \cdot \max\left\{\mathrm{Swap}(a_v \to a_{vL}), 0\right\}\right] \\ &\geq \mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_{vL} = 0 \wedge r_v = 1] \cdot \mathrm{Swap}(a_v \to a_{vL})\right] \\ &= \frac{1}{8D}\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] + \frac{1}{4D}\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL} = 0]\right]\end{aligned} \tag{38}$$

Similarly,

$$\mathrm{Swap}(a_v \to \dot{a}_{vL}) = \sum_{t=1}^{T} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\left(\mathbf{u}^{(t)}[\dot{a}_{vL}] - \mathbf{u}^{(t)}[a_v]\right)$$
$$= \frac{d+1}{2D}\left(\mathbb{1}\left[r_{vL} = 1\right]\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right) - \frac{d}{2D}\left(\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbb{1}\left[r_v = 0\right]\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right)$$

Therefore,

$$\mathbb{1}[r_v = 1] \cdot \mathrm{Swap}(a_v \to \dot{a}_{vL}) = \frac{1}{2D}\mathbb{1}[r_v = 1]\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL} = 1] \tag{39}$$
$$- \frac{d}{2D}\mathbb{1}[r_v = 1]\left(\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL} = 0]\right) \tag{40}$$

and

$$\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_v=1]\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=1]\right]=\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_v=1]\right]\mathbb{E}_{r^{(T)}}\left[\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=1]\right]$$
$$=\frac{1}{2}\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=1]\right] \tag{41}$$

due to the independence of $r_v$ from $r_{vL}, r^{(t)}$ for $t \le \bar{t}_{vLR}$. And,

$$\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_v=1]\left(\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]+\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=0]\right)\right]$$
$$=\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_v=1]\right]\mathbb{E}_{r^{(T)}}\left[\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]+\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=0]\right]$$
$$=\frac{1}{2}\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]+\frac{1}{2}\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=0]\right] \tag{42}$$

due to the independence of $r_v$ from $r^{(t)}$ for $t \le \bar{t}_{vLL}$. So, combining equations (39), (40), (41), and (42), we have

$$\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_v=1]\cdot\mathrm{Swap}(a_v\to\dot{a}_{vL})\right]$$
$$=\frac{1}{4D}\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=1]\right]-\frac{d}{4D}\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]-\frac{d}{4D}\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=0]\right]$$
$$\ge\frac{1}{4D}\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=1]\right]-\frac{1}{4}\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]-\frac{1}{4}\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=0]\right]$$

since $d \le D$. Lastly, since $\mathrm{Swap}(a_v \to a_v) = 0$,

$$\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\mathrm{Swap}(a_v\to a')\right]$$
$$\ge\mathbb{E}_{r^{(T)}}\left[\max\left\{\mathrm{Swap}(a_v\to\dot{a}_{vL}),0\right\}\right]$$
$$\ge\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_v=1]\cdot\max\left\{\mathrm{Swap}(a_v\to\dot{a}_{vL}),0\right\}\right]$$
$$\ge\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_v=1]\cdot\mathrm{Swap}(a_v\to\dot{a}_{vL})\right]$$
$$\ge\frac{1}{4D}\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=1]\right]-\frac{1}{4}\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]-\frac{1}{4}\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=0]\right] \tag{43}$$

Similarly,

$$\begin{aligned}
&\text{Swap}(a_v \to a_{vR}) = \sum_{t=1}^{T} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \left(\mathbf{u}^{(t)}[a_{vR}] - \mathbf{u}^{(t)}[a_v]\right) \\
&= \frac{d+1}{2D} \left( \sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbb{1}\left[r_{vR} = 0\right] \sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right) - \frac{d}{2D} \left( \sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbb{1}\left[r_v = 0\right] \sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right) \\
&\geq \frac{d+1}{2D} \left( \sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbb{1}\left[r_{vR} = 0\right] \sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right) - \frac{d}{2D} \left( \sum_{t=\underline{t}_{v}}^{\bar{t}_{v}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right)
\end{aligned}$$

bounding $\mathbb{1}[r_v = 0] \leq 1$. Therefore,

$$\mathbb{1}[r_{vR} = 0] \cdot \text{Swap}(a_v \to a_{vR}) \geq \frac{1}{2D} \mathbb{1}[r_{vR} = 0] \sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \tag{44}$$

$$+ \frac{1}{2D} \sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \mathbb{1}[r_{vR} = 0] \tag{45}$$

$$- \frac{d}{2D} \mathbb{1}[r_{vR} = 0] \sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \tag{46}$$

and

$$\begin{aligned}
\mathbb{E}_{r^{(T)}} \left[ \mathbb{1}[r_{vR} = 0] \sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] &= \mathbb{E}_{r^{(T)}} \left[\mathbb{1}[r_{vR} = 0]\right] \mathbb{E}_{r^{(T)}} \left[ \sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] \\
&= \frac{1}{2} \sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] \qquad (47)
\end{aligned}$$

due to the independence of $r_{vR}$ from $r^{(t)}$ for $t \leq \bar{t}_{vRL}$. And,

$$\begin{aligned}
\mathbb{E}_{r^{(T)}} \left[ \mathbb{1}[r_{vR} = 0] \sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] &= \mathbb{E}_{r^{(T)}} \left[\mathbb{1}[r_{vR} = 0]\right] \mathbb{E}_{r^{(T)}} \left[ \sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] \\
&= \frac{1}{2} \sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] \qquad (48)
\end{aligned}$$

due to the independence of $r_{vR}$ from $r^{(t)}$ for $t \leq \bar{t}_{vL}$. So, combining equations (44), (45), (46), (47),

and (48), we have

$$\begin{aligned}
&\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_{vR}=0]\cdot \mathrm{Swap}(a_v \to a_{vR})\right] \\
&\geq \frac{1}{4D}\sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] + \frac{1}{2D}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vR}=0]\right] - \frac{d}{4D}\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] \\
&\geq \frac{1}{4D}\sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] + \frac{1}{2D}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vR}=0]\right] - \frac{1}{4}\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]
\end{aligned}$$

since $d \leq D$. Lastly, since $\mathrm{Swap}(a_v \to a_v) = 0$,

$$\begin{aligned}
&\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]} \mathrm{Swap}(a_v \to a')\right] \\
&\geq \mathbb{E}_{r^{(T)}}\left[\max\left\{\mathrm{Swap}(a_v \to a_{vR}), 0\right\}\right] \\
&\geq \mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_{vR}=0]\cdot \max\left\{\mathrm{Swap}(a_v \to a_{vR}), 0\right\}\right] \\
&\geq \mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_{vR}=0]\cdot \mathrm{Swap}(a_v \to a_{vR})\right] \\
&\geq \frac{1}{4D}\sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] + \frac{1}{2D}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vR}=0]\right] - \frac{1}{4}\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]
\end{aligned} \tag{49}$$

Similarly,

$$\begin{aligned}
&\mathrm{Swap}(a_v \to \dot{a}_{vR}) = \sum_{t=1}^{T} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\left(\mathbf{u}^{(t)}[\dot{a}_{vR}] - \mathbf{u}^{(t)}[a_v]\right) \\
&= \frac{d+1}{2D}\left(\mathbb{1}\left[r_{vR}=1\right]\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right) - \frac{d}{2D}\left(\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbb{1}\left[r_v=0\right]\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right) \\
&\geq \frac{d+1}{2D}\left(\mathbb{1}\left[r_{vR}=1\right]\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right) - \frac{d}{2D}\left(\sum_{t=\underline{t}_{v}}^{\bar{t}_{v}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right)
\end{aligned}$$

bounding $\mathbb{1}[r_v=0] \leq 1$

$$= \frac{1}{2D}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}\left[r_{vR}=1\right] - \frac{d}{2D}\left(\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vRL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}\left[r_{vR}=0\right]\right)$$

So, we have

$$\begin{aligned}
&\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\mathrm{Swap}(a_v\to a')\right] \geq \mathbb{E}_{r^{(T)}}\left[\mathrm{Swap}(a_v\to \dot{a}_{vR})\right]\\
&\geq \frac{1}{2D}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}\left[r_{vR}=1\right]\right] - \frac{d}{2D}\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vRL}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] - \frac{d}{2D}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}\left[r_{vR}=0\right]\right]\\
&\geq \frac{1}{2D}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}\left[r_{vR}=1\right]\right] - \frac{1}{2}\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vRL}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] - \frac{1}{2}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}\left[r_{vR}=0\right]\right]
\end{aligned} \tag{50}$$

since $d \leq D$.

Similarly, recalling our notation that the empty string $\emptyset$ represents the root of the tree,

$$\begin{aligned}
&\mathrm{Swap}(a_v\to a_\emptyset) = \sum_{t=1}^{T}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\left(\mathbf{u}^{(t)}[a_\emptyset]-\mathbf{u}^{(t)}[a_v]\right)\\
&= \frac{1}{2D}\left(\sum_{t=\underline{t}_{\emptyset L}}^{\bar{t}_{\emptyset L}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbb{1}\left[r_\emptyset=0\right]\sum_{t=\underline{t}_{\emptyset R}}^{\bar{t}_{\emptyset R}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right) - \frac{d}{2D}\left(\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbb{1}\left[r_v=0\right]\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right)\\
&\geq \frac{1}{2D}\left(\sum_{t=\underline{t}_{\emptyset L}}^{\bar{t}_{\emptyset L}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbb{1}\left[r_\emptyset=0\right]\sum_{t=\underline{t}_{\emptyset R}}^{\bar{t}_{\emptyset R}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right) - \frac{d}{2D}\left(\sum_{t=\underline{t}_{v}}^{\bar{t}_{v}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right)
\end{aligned}$$

bounding $\mathbb{1}[r_v=0]\leq 1$. Also,

$$\begin{aligned}
&\mathrm{Swap}(a_v\to \dot{a}_\emptyset) = \sum_{t=1}^{T}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\left(\mathbf{u}^{(t)}[\dot{a}_\emptyset]-\mathbf{u}^{(t)}[a_v]\right)\\
&= \frac{1}{2D}\left(\mathbb{1}\left[r_\emptyset=1\right]\sum_{t=\underline{t}_{\emptyset R}}^{\bar{t}_{\emptyset R}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right) - \frac{d}{2D}\left(\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbb{1}\left[r_v=0\right]\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right)\\
&\geq \frac{1}{2D}\left(\mathbb{1}\left[r_\emptyset=1\right]\sum_{t=\underline{t}_{\emptyset R}}^{\bar{t}_{\emptyset R}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right) - \frac{d}{2D}\left(\sum_{t=\underline{t}_{v}}^{\bar{t}_{v}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right)
\end{aligned}$$

So,

$$\begin{aligned}
\mathrm{Swap}(a_v\to a_\emptyset)+\mathrm{Swap}(a_v\to \dot{a}_\emptyset) &\geq \frac{1}{2D}\left(\sum_{t=1}^{T'}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right) - \frac{d}{D}\left(\sum_{t=\underline{t}_{v}}^{\bar{t}_{v}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right)\\
&\geq \frac{1}{2D}\left(\sum_{t=1}^{T'}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right) - \left(\sum_{t=\underline{t}_{v}}^{\bar{t}_{v}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right)
\end{aligned}$$

since $d \leq D$. Thus,

$$\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \text{Swap}(a_v \to a')\right] \geq \frac{1}{2}\mathbb{E}_{r^{(t)}}\left[\text{Swap}(a_v \to a_\emptyset) + \text{Swap}(a_v \to \dot{a}_\emptyset)\right]$$
$$\geq \frac{1}{4D}\sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] - \frac{1}{2}\sum_{t=\underline{t}_v}^{\bar{t}_v} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] \tag{51}$$

Collecting equations (38), (43), (49), (50), and (51),

$$\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \text{Swap}(a_v \to a')\right]$$
$$\geq \frac{1}{8D}\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] + \frac{1}{4D}\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=0]\right] \tag{38}$$

$$\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \text{Swap}(a_v \to a')\right]$$
$$\geq \frac{1}{4D}\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=1]\right] - \frac{1}{4}\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] - \frac{1}{4}\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=0]\right] \tag{43}$$

$$\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \text{Swap}(a_v \to a')\right]$$
$$\geq \frac{1}{4D}\sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] + \frac{1}{2D}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vR}=0]\right] - \frac{1}{4}\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] \tag{49}$$

$$\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \text{Swap}(a_v \to a')\right]$$
$$\geq \frac{1}{2D}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}\left[r_{vR}=1\right]\right] - \frac{1}{2}\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vRL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] - \frac{1}{2}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}\left[r_{vR}=0\right]\right] \tag{50}$$

$$\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \text{Swap}(a_v \to a')\right]$$
$$\geq \frac{1}{4D}\sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] - \frac{1}{2}\sum_{t=\underline{t}_v}^{\bar{t}_v} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] \tag{51}$$

Summing the inequalities

$$4D(51) + 4D^2(50) + 16D^3(49) + 32D^4(43) + 128D^5(38)$$

gives

$$
\begin{aligned}
&184D^5\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\mathrm{Swap}(a_v\to a')\right]\\
&\geq\sum_{t=1}^{T'}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]\\
&+(-2D-2D^2-4D^3-8D^4+16D^4)\sum_{t=\underline{t}_{vLL}}^{\bar{t}_{vLL}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]\\
&+(-2D-2D^2-4D^3-8D^4+32D^4)\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=0]\right]\\
&+(-2D-2D^2-4D^3+8D^3)\sum_{t=\underline{t}_{vLR}}^{\bar{t}_{vLR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vL}=1]\right]\\
&+(-2D-2D^2+4D^2)\sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]\\
&+(-2D-2D^2+8D^2)\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vR}=0]\right]\\
&+(-2D+2D)\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}[r_{vR}=1]\right]\\
&\geq\sum_{t=1}^{T'}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]
\end{aligned}
$$

which gives (33), as desired. □

**Lemma A.2** (Case 2)**.** *Let $v$ be a node in the tree of depth $d\leq D-2$. Then,*

$$
24D^3\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\mathrm{Swap}(\dot{a}_v\to a')\right]\geq\sum_{t=1}^{T'}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right]\tag{52}
$$

*Proof of Lemma A.2.* We have

$$
\begin{aligned}
\mathrm{Swap}(\dot{a}_v\to a_{vR})&=\sum_{t=1}^{T}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\left(\mathbf{u}^{(t)}[a_{vR}]-\mathbf{u}^{(t)}[\dot{a}_v]\right)\\
&=\frac{d+1}{2D}\left(\sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]+\mathbb{1}\left[r_{vR}=0\right]\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right)-\frac{d}{2D}\left(\mathbb{1}\left[r_v=1\right]\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right)\\
&=\frac{d+1}{2D}\left(\sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]+\mathbb{1}\left[r_{vR}=0\right]\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right)-\frac{d}{2D}\left(\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right)
\end{aligned}
$$

bounding $\mathbb{1}[r_v = 1] \leq 1$. Therefore,

$$\mathbb{1}[r_{vR} = 0] \cdot \text{Swap}(\dot{a}_v \to a_{vR}) \geq \frac{1}{2D}\mathbb{1}[r_{vR} = 0] \sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] + \frac{1}{2D} \sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\mathbb{1}[r_{vR} = 0] \quad (53)$$

and

$$\begin{aligned}\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_{vR} = 0] \sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\right] &= \mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_{vR} = 0]\right] \mathbb{E}_{r^{(T)}}\left[\sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\right] \\ &= \frac{1}{2} \sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\right] \quad (54)\end{aligned}$$

due to the independence of $r_{vR}$ from $r^{(t)}$ for $t \leq \bar{t}_{vRL}$. So, combining equations (53), and (54), we have

$$\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_{vR} = 0] \cdot \text{Swap}(\dot{a}_v \to a_{vR})\right] \geq \frac{1}{4D} \sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\right] + \frac{1}{2D} \sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\mathbb{1}[r_{vR} = 0]\right]$$

Lastly, since $\text{Swap}(\dot{a}_v \to \dot{a}_v) = 0$,

$$\begin{aligned}\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \text{Swap}(\dot{a}_v \to a')\right] &\geq \mathbb{E}_{r^{(T)}}\left[\max\left\{\text{Swap}(\dot{a}_v \to a_{vR}), 0\right\}\right] \\ &\geq \mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_{vR} = 0] \cdot \max\left\{\text{Swap}(\dot{a}_v \to a_{vR}), 0\right\}\right] \\ &\geq \mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_{vR} = 0] \cdot \text{Swap}(\dot{a}_v \to a_{vR})\right] \\ &\geq \frac{1}{4D} \sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\right] + \frac{1}{2D} \sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\mathbb{1}[r_{vR} = 0]\right] \quad (55)\end{aligned}$$

Similarly,

$$\begin{aligned}&\text{Swap}(\dot{a}_v \to \dot{a}_{vR}) = \sum_{t=1}^{T} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\left(\mathbf{u}^{(t)}[\dot{a}_{vR}] - \mathbf{u}^{(t)}[\dot{a}_v]\right) \\ &= \frac{d+1}{2D}\left(\mathbb{1}\left[r_{vR} = 1\right] \sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\right) - \frac{d}{2D}\left(\mathbb{1}\left[r_v = 1\right] \sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\right) \\ &\geq \frac{d+1}{2D}\left(\mathbb{1}\left[r_{vR} = 1\right] \sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\right) - \frac{d}{2D}\left(\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\right)\end{aligned}$$

bounding $\mathbb{1}[r_v = 1] \leq 1$

$$= \frac{1}{2D} \sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\mathbb{1}\left[r_{vR} = 1\right] - \frac{d}{2D}\left(\sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] + \sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]\mathbb{1}\left[r_{vR} = 0\right]\right)$$

So, we have

$$\begin{aligned}
&\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\mathrm{Swap}(\dot{a}_v\to a')\right] \geq \mathbb{E}_{r^{(T)}}\left[\mathrm{Swap}(\dot{a}_v\to \dot{a}_{vR})\right] \\
&\geq \frac{1}{2D}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}\left[r_{vR}=1\right]\right] - \frac{d}{2D}\sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right] - \frac{d}{2D}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}\left[r_{vR}=0\right]\right] \\
&\geq \frac{1}{2D}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}\left[r_{vR}=1\right]\right] - \frac{1}{2}\sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right] - \frac{1}{2}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}\left[r_{vR}=0\right]\right]
\end{aligned} \tag{56}$$

since $d \leq D$.

Similarly, recalling our notation that the empty string $\emptyset$ represents the root of the tree,

$$\begin{aligned}
\mathrm{Swap}(\dot{a}_v\to a_\emptyset) &= \sum_{t=1}^{T}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\left(\mathbf{u}^{(t)}[a_\emptyset]-\mathbf{u}^{(t)}[\dot{a}_v]\right) \\
&= \frac{1}{2D}\left(\sum_{t=\underline{t}_{\emptyset L}}^{\bar{t}_{\emptyset L}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v] + \mathbb{1}\left[r_\emptyset=0\right]\sum_{t=\underline{t}_{\emptyset R}}^{\bar{t}_{\emptyset R}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right) - \frac{d}{2D}\left(\mathbb{1}\left[r_v=1\right]\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right) \\
&\geq \frac{1}{2D}\left(\sum_{t=\underline{t}_{\emptyset L}}^{\bar{t}_{\emptyset L}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v] + \mathbb{1}\left[r_\emptyset=0\right]\sum_{t=\underline{t}_{\emptyset R}}^{\bar{t}_{\emptyset R}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right) - \frac{d}{2D}\left(\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right)
\end{aligned}$$

bounding $\mathbb{1}[r_v=1]\leq 1$. Also,

$$\begin{aligned}
\mathrm{Swap}(\dot{a}_v\to \dot{a}_\emptyset) &= \sum_{t=1}^{T}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\left(\mathbf{u}^{(t)}[\dot{a}_\emptyset]-\mathbf{u}^{(t)}[\dot{a}_v]\right) \\
&= \frac{1}{2D}\left(\mathbb{1}\left[r_\emptyset=1\right]\sum_{t=\underline{t}_{\emptyset R}}^{\bar{t}_{\emptyset R}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right) - \frac{d}{2D}\left(\mathbb{1}\left[r_v=1\right]\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right) \\
&\geq \frac{1}{2D}\left(\mathbb{1}\left[r_\emptyset=1\right]\sum_{t=\underline{t}_{\emptyset R}}^{\bar{t}_{\emptyset R}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right) - \frac{d}{2D}\left(\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right)
\end{aligned}$$

So,

$$\begin{aligned}
\mathrm{Swap}(\dot{a}_v\to a_\emptyset)+\mathrm{Swap}(\dot{a}_v\to \dot{a}_\emptyset) &\geq \frac{1}{2D}\left(\sum_{t=1}^{T'}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right) - \frac{d}{D}\left(\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right) \\
&\geq \frac{1}{2D}\left(\sum_{t=1}^{T'}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right) - \left(\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right)
\end{aligned}$$

since $d \le D$. Thus,

$$\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \text{Swap}(\dot{a}_v \to a')\right] \ge \frac{1}{2}\mathbb{E}_{r^{(t)}}\left[\text{Swap}(\dot{a}_v \to a_\emptyset) + \text{Swap}(\dot{a}_v \to \dot{a}_\emptyset)\right]$$
$$\ge \frac{1}{4D}\sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right] - \frac{1}{2}\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right] \tag{57}$$

Collecting equations (55), (56), and (57),

$$\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \text{Swap}(\dot{a}_v \to a')\right]$$
$$\ge \frac{1}{4D}\sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right] + \frac{1}{2D}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}[r_{vR}=0]\right] \tag{55}$$
$$\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \text{Swap}(\dot{a}_v \to a')\right]$$
$$\ge \frac{1}{2D}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}\left[r_{vR}=1\right]\right] - \frac{1}{2}\sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right] - \frac{1}{2}\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}\left[r_{vR}=0\right]\right] \tag{56}$$
$$\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \text{Swap}(\dot{a}_v \to a')\right]$$
$$\ge \frac{1}{4D}\sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right] - \frac{1}{2}\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right] \tag{57}$$

Summing the inequalities

$$4D(57) + 4D^2(56) + 16D^3(55)$$

gives

$$24D^3\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \text{Swap}(\dot{a}_v \to a')\right] \ge \sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right]$$
$$+ (-2D - 2D^2 + 4D^2)\sum_{t=\underline{t}_{vRL}}^{\bar{t}_{vRL}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right]$$
$$+ (-2D - 2D^2 + 8D^2)\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}[r_{vR}=0]\right]$$
$$+ (-2D + 2D)\sum_{t=\underline{t}_{vRR}}^{\bar{t}_{vRR}} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}[r_{vR}=1]\right]$$
$$\ge \sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right] \tag{58}$$

which gives (52), as desired. □

**Lemma A.3** (Case 3)**.** *Let $v$ be a node in the tree of depth $d = D - 1$. Then,*

$$24D^3\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\mathrm{Swap}(a_v \to a')\right] \geq \sum_{t=1}^{T'}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] \tag{59}$$

*Proof of Lemma A.3.* In this case, we have nodes $vL$ and $vR$ both leaves. So, there are Bernoulli random variables $r_{(vL,t)}$ and $r_{(vR,t)}$ for every time step $t$ in $[\underline{t}_{vL}, \bar{t}_{vL}]$ and $[\underline{t}_{vR}, \bar{t}_{vR}]$ respectively. By definition,

$$\mathbf{u}^{(t)}[a_{vL}] = \mathbb{1}\left[r_{(vL,t)} = 0\right] \qquad \mathbf{u}^{(t)}[\dot{a}_{vL}] = \mathbb{1}\left[r_{(vL,t)} = 1\right]$$
$$\mathbf{u}^{(t)}[a_{vR}] = \mathbb{1}\left[r_{(vR,t)} = 0\right] \qquad \mathbf{u}^{(t)}[\dot{a}_{vR}] = \mathbb{1}\left[r_{(vR,t)} = 1\right]$$

for $t$ in $[\underline{t}_{vL}, \bar{t}_{vL}]$ and $[\underline{t}_{vR}, \bar{t}_{vR}]$ respectively. We have

$$\begin{aligned}\mathrm{Swap}(a_v \to a_{vL}) + \mathrm{Swap}(a_v \to \dot{a}_{vL}) &= \sum_{t=1}^{T}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\left(\mathbf{u}^{(t)}[a_{vL}] + \mathbf{u}^{(t)}[\dot{a}_{vL}] - 2\mathbf{u}^{(t)}[a_v]\right)\\ &= \left(\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right) - \frac{2(D-1)}{2D}\left(\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbb{1}\left[r_v = 0\right]\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right)\end{aligned}$$

Therefore,

$$\mathbb{1}[r_v = 1]\cdot\left(\mathrm{Swap}(a_v \to a_{vL}) + \mathrm{Swap}(a_v \to \dot{a}_{vL})\right) = \frac{1}{D}\mathbb{1}\left[r_v = 1\right]\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]$$

and

$$\mathbb{E}_{r^{(T)}}\left[\mathbb{1}\left[r_v = 1\right]\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] = \mathbb{E}_{r^{(T)}}\left[\mathbb{1}\left[r_v = 1\right]\right]\mathbb{E}_{r^{(T)}}\left[\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] = \frac{1}{2}\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]$$

due to the independence of $r_v$ from $r^{(t)}$ for $t \leq \bar{t}_{vL}$. Lastly, since $\mathrm{Swap}(a_v \to a_v) = 0$,

$$\begin{aligned}\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\mathrm{Swap}(a_v \to a')\right] &\geq \mathbb{E}_{r^{(T)}}\left[\max\left\{\frac{1}{2}\left(\mathrm{Swap}(a_v \to a_{vL}) + \mathrm{Swap}(a_v \to \dot{a}_{vL})\right), 0\right\}\right]\\ &\geq \mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_v = 1]\cdot\max\left\{\frac{1}{2}\left(\mathrm{Swap}(a_v \to a_{vL}) + \mathrm{Swap}(a_v \to \dot{a}_{vL})\right), 0\right\}\right]\\ &\geq \frac{1}{2}\mathbb{E}_{r^{(T)}}\left[\mathbb{1}[r_v = 1]\cdot\left(\mathrm{Swap}(a_v \to a_{vL}) + \mathrm{Swap}(a_v \to \dot{a}_{vL})\right)\right]\\ &\geq \frac{1}{4D}\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]\end{aligned} \tag{60}$$

Similarly

$$\begin{aligned}
&\mathrm{Swap}(a_v \to a_{vR}) + \mathrm{Swap}(a_v \to \dot{a}_{vR}) = \sum_{t=1}^{T} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \left( \mathbf{u}^{(t)}[a_{vR}] + \mathbf{u}^{(t)}[\dot{a}_{vR}] - 2\mathbf{u}^{(t)}[a_v] \right) \\
&= \left( \sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right) - \frac{2(D-1)}{2D} \left( \sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbb{1}[r_v = 0] \sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right) \\
&\geq \frac{1}{D} \sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] - \sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]
\end{aligned}$$

bounding $\mathbb{1}[r_v = 0] \leq 1$. Thus,

$$\begin{aligned}
\mathbb{E}_{r^{(T)}} \left[ \max_{a' \in [N]} \mathrm{Swap}(a_v \to a') \right] &\geq \mathbb{E}_{r^{(T)}} \left[ \frac{1}{2} \left( \mathrm{Swap}(a_v \to a_{vR}) + \mathrm{Swap}(a_v \to \dot{a}_{vR}) \right) \right] \\
&\geq \frac{1}{2D} \sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] - \frac{1}{2} \sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] \qquad (61)
\end{aligned}$$

Similarly

$$\begin{aligned}
&\mathrm{Swap}(a_v \to a_{\emptyset}) + \mathrm{Swap}(a_v \to \dot{a}_{\emptyset}) = \sum_{t=1}^{T} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \left( \mathbf{u}^{(t)}[a_{\emptyset}] + \mathbf{u}^{(t)}[\dot{a}_{\emptyset}] - 2\mathbf{u}^{(t)}[a_v] \right) \\
&= \frac{1}{2D} \left( \sum_{t=1}^{T'} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right) - \frac{2(D-1)}{2D} \left( \sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] + \mathbb{1}[r_v = 0] \sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right) \\
&\geq \frac{1}{2D} \sum_{t=1}^{T'} \mathbf{x}^{(t)}_{r^{(t)}}[a_v] - \sum_{t=\underline{t}_{v}}^{\bar{t}_{v}} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]
\end{aligned}$$

bounding $\mathbb{1}[r_v = 0] \leq 1$. Thus,

$$\begin{aligned}
\mathbb{E}_{r^{(T)}} \left[ \max_{a' \in [N]} \mathrm{Swap}(a_v \to a') \right] &\geq \mathbb{E}_{r^{(T)}} \left[ \frac{1}{2} \left( \mathrm{Swap}(a_v \to a_{\emptyset}) + \mathrm{Swap}(a_v \to \dot{a}_{\emptyset}) \right) \right] \\
&\geq \frac{1}{4D} \sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] - \frac{1}{2} \sum_{t=\underline{t}_{v}}^{\bar{t}_{v}} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] \qquad (62)
\end{aligned}$$

Collecting equations (60), (61), and (62),

$$\mathbb{E}_{r^{(T)}} \left[ \max_{a' \in [N]} \mathrm{Swap}(a_v \to a') \right] \geq \frac{1}{4D} \sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] \qquad (60)$$

$$\mathbb{E}_{r^{(T)}} \left[ \max_{a' \in [N]} \mathrm{Swap}(a_v \to a') \right] \geq \frac{1}{2D} \sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] - \frac{1}{2} \sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] \qquad (61)$$

$$\mathbb{E}_{r^{(T)}} \left[ \max_{a' \in [N]} \mathrm{Swap}(a_v \to a') \right] \geq \frac{1}{4D} \sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] - \frac{1}{2} \sum_{t=\underline{t}_{v}}^{\bar{t}_{v}} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}^{(t)}_{r^{(t)}}[a_v] \right] \qquad (62)$$

Summing the inequalities

$$4D(62) + 4D^2(61) + 16D^3(60)$$

gives

$$\begin{aligned}
24D^3\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\text{Swap}(a_v \to a')\right] &\geq \sum_{t=1}^{T'}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] \\
&\quad + (-2D - 2D^2 + 4D^2)\sum_{t=\underline{t}_{vL}}^{\bar{t}_{vL}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] \\
&\quad + (-2D + 2D)\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] \\
&\geq \sum_{t=1}^{T'}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]
\end{aligned}$$

which gives (59), as desired. □

**Lemma A.4** (Case 4)**.** *Let $v$ be a node in the tree of depth $d = D - 1$. Then,*

$$8D^2\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\text{Swap}(\dot{a}_v \to a')\right] \geq \sum_{t=1}^{T'}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right] \tag{63}$$

*Proof of Lemma A.4.* In this case, we have that node $vR$ is a leaf. So, there are Bernoulli random variables $r_{(vR,t)}$ for every time step $t \in [\underline{t}_{vR}, \bar{t}_{vR}]$. By definition,

$$\mathbf{u}^{(t)}[a_{vR}] = \mathbb{1}\left[r_{(vR,t)} = 0\right] \qquad \mathbf{u}^{(t)}[\dot{a}_{vR}] = \mathbb{1}\left[r_{(vR,t)} = 1\right]$$

for $t \in [\underline{t}_{vR}, \bar{t}_{vR}]$. We have

$$\begin{aligned}
\text{Swap}(\dot{a}_v \to a_{vR}) + \text{Swap}(\dot{a}_v \to \dot{a}_{vR}) &= \sum_{t=1}^{T}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\left(\mathbf{u}^{(t)}[a_{vR}] + \mathbf{u}^{(t)}[\dot{a}_{vR}] - 2\mathbf{u}^{(t)}[\dot{a}_v]\right) \\
&= \left(\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right) - \frac{2(D-1)}{2D}\left(\mathbb{1}\left[r_v = 1\right]\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right) \\
&\geq \frac{1}{D}\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]
\end{aligned}$$

bounding $\mathbb{1}\left[r_v = 1\right] \leq 1$. Thus,

$$\begin{aligned}
\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\text{Swap}(\dot{a}_v \to a')\right] &\geq \mathbb{E}_{r^{(T)}}\left[\frac{1}{2}\left(\text{Swap}(\dot{a}_v \to a_{vR}) + \text{Swap}(\dot{a}_v \to \dot{a}_{vR})\right)\right] \\
&\geq \frac{1}{2D}\sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right]
\end{aligned} \tag{64}$$

Similarly

$$\begin{aligned}
&\text{Swap}(\dot{a}_v \to a_\emptyset) + \text{Swap}(\dot{a}_v \to \dot{a}_\emptyset) = \sum_{t=1}^{T} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] \left( \mathbf{u}^{(t)}[a_\emptyset] + \mathbf{u}^{(t)}[\dot{a}_\emptyset] - 2\mathbf{u}^{(t)}[\dot{a}_v] \right) \\
&= \frac{1}{2D} \left( \sum_{t=1}^{T'} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] \right) - \frac{2(D-1)}{2D} \left( \mathbb{1}[r_v = 1] \sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] \right) \\
&\geq \frac{1}{2D} \sum_{t=1}^{T'} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] - \sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v]
\end{aligned}$$

bounding $\mathbb{1}[r_v = 1] \leq 1$. Thus,

$$\begin{aligned}
\mathbb{E}_{r^{(T)}} \left[ \max_{a' \in [N]} \text{Swap}(\dot{a}_v \to a') \right] &\geq \mathbb{E}_{r^{(T)}} \left[ \frac{1}{2} \left( \text{Swap}(\dot{a}_v \to a_\emptyset) + \text{Swap}(\dot{a}_v \to \dot{a}_\emptyset) \right) \right] \\
&\geq \frac{1}{4D} \sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] \right] - \frac{1}{2} \sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] \right] \qquad (65)
\end{aligned}$$

Collecting equations (64) and (65),

$$\mathbb{E}_{r^{(T)}} \left[ \max_{a' \in [N]} \text{Swap}(a_v \to a') \right] \geq \frac{1}{2D} \sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] \right] \qquad (64)$$

$$\mathbb{E}_{r^{(T)}} \left[ \max_{a' \in [N]} \text{Swap}(a_v \to a') \right] \geq \frac{1}{4D} \sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] \right] - \frac{1}{2} \sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] \right] \qquad (65)$$

Summing the inequalities

$$4D(64) + 4D^2(65)$$

gives

$$\begin{aligned}
8D^2 \mathbb{E}_{r^{(T)}} \left[ \max_{a' \in [N]} \text{Swap}(\dot{a}_v \to a') \right] &\geq \sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] \right] \\
&\quad + (-2D + 2D) \sum_{t=\underline{t}_{vR}}^{\bar{t}_{vR}} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] \right] \\
&\geq \sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] \right]
\end{aligned}$$

which gives (63), as desired. □

**Lemma A.5** (Case 5)**.** *Let $v$ be a leaf node in the tree (depth $d = D$). Then,*

$$\begin{aligned}
&4100D^4\sqrt{B} \left( \mathbb{E}_{r^{(T)}} \left[ \max_{a' \in [N]} \text{Swap}(a_v \to a') \right] + \mathbb{E}_{r^{(T)}} \left[ \max_{a' \in [N]} \text{Swap}(\dot{a}_v \to a') \right] \right) \\
&\qquad \geq \sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}} \left[ \mathbf{x}_{r^{(t)}}^{(t)}[a_v] + \mathbf{x}_{r^{(t)}}^{(t)}[\dot{a}_v] \right] - \frac{B}{2} \qquad (66)
\end{aligned}$$

*Proof of Lemma A.5.* Since $v$ is a leaf node, there are Bernoulli random variables $r_{(v,t)}$ for every time step $t$ in $[\underline{t}_v, \bar{t}_v]$. By definition,

$$\mathbf{u}^{(t)}[a_v] = \mathbb{1}\left[r_{(v,t)} = 0\right] \qquad \mathbf{u}^{(t)}[\dot{a}_v] = \mathbb{1}\left[r_{(v,t)} = 1\right]$$

for $t \in [\underline{t}_v, \bar{t}_v]$. We have

$$\begin{aligned}
\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \mathrm{Swap}(a_v \to a')\right] &\geq \mathbb{E}_{r^{(T)}}\left[\frac{1}{2}\left(\mathrm{Swap}(a_v \to a_\emptyset) + \mathrm{Swap}(a_v \to \dot{a}_\emptyset)\right)\right] \\
&= \frac{1}{2}\sum_{t=1}^{T} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\left(\mathbf{u}^{(t)}[a_\emptyset] + \mathbf{u}^{(t)}[\dot{a}_\emptyset] - 2\mathbf{u}^{(t)}[a_v]\right)\right] \\
&\geq \frac{1}{4D}\sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] - \sum_{t=\underline{t}_v}^{\bar{t}_v} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] \qquad (67)
\end{aligned}$$

Similarly,

$$\begin{aligned}
\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \mathrm{Swap}(\dot{a}_v \to a')\right] &\geq \mathbb{E}_{r^{(T)}}\left[\frac{1}{2}\left(\mathrm{Swap}(\dot{a}_v \to a_\emptyset) + \mathrm{Swap}(\dot{a}_v \to \dot{a}_\emptyset)\right)\right] \\
&= \frac{1}{2}\sum_{t=1}^{T} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\left(\mathbf{u}^{(t)}[a_\emptyset] + \mathbf{u}^{(t)}[\dot{a}_\emptyset] - 2\mathbf{u}^{(t)}[\dot{a}_v]\right)\right] \\
&\geq \frac{1}{4D}\sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right] - \sum_{t=\underline{t}_v}^{\bar{t}_v} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right] \qquad (68)
\end{aligned}$$

Now,

$$\begin{aligned}
&\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \mathrm{Swap}(a_v \to a')\right] \geq \mathbb{E}_{r^{(T)}}\left[\max\left\{\mathrm{Swap}(a_v \to a_v), \mathrm{Swap}(a_v \to \dot{a}_v)\right\}\right] \\
&= \mathbb{E}_{r^{(T)}}\left[\max\left\{\sum_{t=1}^{T} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\left(\mathbf{u}^{(t)}[a_v] - \mathbf{u}^{(t)}[a_v]\right), \sum_{t=1}^{T} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\left(\mathbf{u}^{(t)}[\dot{a}_v] - \mathbf{u}^{(t)}[a_v]\right)\right\}\right] \\
&= \mathbb{E}_{r^{(T)}}\left[\max\left\{\sum_{t=\underline{t}_v}^{\bar{t}_v} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}\left[r_{(v,t)} = 0\right], \sum_{t=\underline{t}_v}^{\bar{t}_v} \mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}\left[r_{(v,t)} = 1\right]\right\}\right] - \sum_{t=\underline{t}_v}^{\bar{t}_v} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}\left[r_{(v,t)} = 0\right]\right]
\end{aligned}$$

Similarly,

$$\begin{aligned}
&\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \mathrm{Swap}(\dot{a}_v \to a')\right] \\
&\geq \mathbb{E}_{r^{(T)}}\left[\max\left\{\sum_{t=\underline{t}_v}^{\bar{t}_v} \mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}\left[r_{(v,t)} = 0\right], \sum_{t=\underline{t}_v}^{\bar{t}_v} \mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}\left[r_{(v,t)} = 1\right]\right\}\right] - \sum_{t=\underline{t}_v}^{\bar{t}_v} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}\left[r_{(v,t)} = 1\right]\right]
\end{aligned}$$

Now,

$$\begin{aligned}
\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}\left[r_{(v,t)} = 0\right]\right] &= \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]\mathbb{E}_{r^{(T)}}\left[\mathbb{1}\left[r_{(v,t)} = 0\right]\right] \\
&= \frac{1}{2}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]
\end{aligned}$$

and

$$\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}\left[r_{(v,t)}=1\right]\right]=\frac{1}{2}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right]$$

Also, using as shorthand $\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]=\mathbf{x}^{(t)}_{r^{(t)}}[a_v]+\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]$, we have

$$\begin{aligned}
&\max\left\{\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}\left[r_{(v,t)}=0\right],\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\mathbb{1}\left[r_{(v,t)}=1\right]\right\}\\
&+\max\left\{\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}\left[r_{(v,t)}=0\right],\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\mathbb{1}\left[r_{(v,t)}=1\right]\right\}\\
&\geq\max\left\{\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\mathbb{1}\left[r_{(v,t)}=0\right],\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\mathbb{1}\left[r_{(v,t)}=1\right]\right\}
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\mathrm{Swap}(a_v\to a')\right]+\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\mathrm{Swap}(\dot{a}_v\to a')\right]\\
&\geq\mathbb{E}_{r^{(T)}}\left[\max\left\{\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\mathbb{1}\left[r_{(v,t)}=0\right],\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\mathbb{1}\left[r_{(v,t)}=1\right]\right\}\right]\\
&-\frac{1}{2}\mathbb{E}_{r^{(T)}}\left[\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\right]\\
&=\frac{1}{2}\mathbb{E}_{r^{(T)}}\left[\left|\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]Z^{(t)}\right|\right]
\end{aligned}$$

where $Z^{(t)}=\begin{cases}1 & \text{if } r_{(v,t)}=1\\ -1 & \text{if } r_{(v,t)}=0\end{cases}$. If we denote by $X^{(t)}=\sum_{\tau=\underline{t}_v}^{t}\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]Z^{(t)}$, then $X^{(\underline{t}_v)},\cdots,X^{(\bar{t}_v)}$ is a martingale due to the independence of $\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]$ and $Z^{(t)}$ for all $t$. We use the following lemma.

**Lemma A.6.** *Consider an algorithm which, at any round $t=1,\dots,B$, selects a parameter $x_t\in[0,1]$, possibly at random. Then, it observes the outcome of a random variable $Z_t\sim\mathrm{Uniform}(\{-1,1\})$. Assume that $\mathbb{E}\left[\sum_{t=1}^{B}x_t^2\right]\geq\epsilon B$ for some $\epsilon>0$. Then,*

$$\mathbb{E}\left[\left|\sum_{t=1}^{B}x_tZ_t\right|\right]\geq\frac{\epsilon\sqrt{B}}{4\sqrt{\log(1/\epsilon)}}$$

We want to apply the lemma with $x_t=\frac{1}{2}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\right]$. We have $\frac{1}{2}\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\leq 1$ for all $t$. If we assume,

$$\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\right]\geq\frac{B}{8D}\tag{69}$$

Jensen's inequality gives

$$\sum_{t=\underline{t}_v}^{\bar{t}_v}\left(\frac{1}{2}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\right]\right)^2 \geq \frac{B}{4}\left(\frac{1}{B}\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\right]\right)^2 \geq \frac{B}{256D^2}$$

and the preconditions of Lemma (A.6) hold for $\epsilon = \frac{1}{256D^2}$. Thus,

$$\begin{aligned}\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\text{Swap}(a_v\to a')\right]+\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\text{Swap}(\dot{a}_v\to a')\right] &\geq \mathbb{E}_{r^{(T)}}\left[\left|\sum_{t=\underline{t}_v}^{\bar{t}_v}\left(\frac{1}{2}\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\right)Z^{(t)}\right|\right]\\ &\geq \frac{\sqrt{B}}{1024D^3}\end{aligned}$$

for $D\geq 3$. Equivalently, incorporating assumption (69) into the equation,

$$\begin{aligned}&\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\text{Swap}(a_v\to a')\right]+\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\text{Swap}(\dot{a}_v\to a')\right]\\ &\geq \frac{\sqrt{B}}{1024D^3}\mathbb{1}\left[\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\right]\geq\frac{B}{8D}\right]\\ &\geq \frac{\sqrt{B}}{1024D^3}\frac{1}{B}\left(\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\right]\right)\mathbb{1}\left[\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\right]\geq\frac{B}{8D}\right]\end{aligned}$$

because $\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\right]\leq B$

$$\begin{aligned}&\geq \frac{1}{1024D^3\sqrt{B}}\left(\left(\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\right]\right)-\frac{B}{8D}\right)\\ &= \frac{1}{1024D^3\sqrt{B}}\left(\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v+\dot{a}_v]\right]\right)-\frac{\sqrt{B}}{8192D^4}\end{aligned}\tag{70}$$

Collecting equations (67), (68), and (70),

$$\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\text{Swap}(a_v\to a')\right]\geq\frac{1}{4D}\sum_{t=1}^{T'}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]-\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right]\tag{67}$$

$$\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\text{Swap}(\dot{a}_v\to a')\right]\geq\frac{1}{4D}\sum_{t=1}^{T'}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right]-\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right]\tag{68}$$

$$\begin{aligned}&\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\text{Swap}(a_v\to a')\right]+\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\text{Swap}(\dot{a}_v\to a')\right]\\ &\qquad\geq\frac{1}{1024D^3\sqrt{B}}\left(\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]+\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right]\right)-\frac{\sqrt{B}}{8192D^4}\end{aligned}\tag{70}$$

Summing the inequalities

$$4D(67) + 4D(68) + 4096D^4\sqrt{B}(70)$$

gives

$$\begin{aligned}
&4100D^4\sqrt{B}\left(\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\mathrm{Swap}(a_v\to a')\right]+\mathbb{E}_{r^{(T)}}\left[\max_{a'\in[N]}\mathrm{Swap}(\dot{a}_v\to a')\right]\right)\\
&\geq\sum_{t=1}^{T'}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]+\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right]+(-4D+4D)\sum_{t=\underline{t}_v}^{\bar{t}_v}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]+\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right]-\frac{B}{2}\\
&=\sum_{t=1}^{T'}\mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]+\mathbf{x}^{(t)}_{r^{(t)}}[\dot{a}_v]\right]-\frac{B}{2}
\end{aligned}$$

which gives (66), as desired. □

*Proof of Lemma A.6.* Denote by $X=\sum_{t=1}^{B}x_tZ_t$. First, it holds that

$$\mathbb{E}\left[X^2\right]=\mathbb{E}\left[\sum_{t=1}^{B}x_t^2\right]$$

Next, from Azuma's inequality, we know that for any $C>0$, $\Pr[|X|\geq C]\leq 2\exp(-C^2/(2B))$. For any $R>0$, we can write

$$\begin{aligned}
\epsilon B&\leq\mathbb{E}\left[\sum_{t=1}^{B}x_t^2\right]\\
&=\mathbb{E}[X^2]\\
&=\mathbb{E}[X^2\mathbb{1}(|X|\leq R)]+\mathbb{E}[X^2\mathbb{1}(|X|>R)]\\
&\leq\mathbb{E}[|X|R\mathbb{1}(|X|\leq R)]+\mathbb{E}[X^2\mathbb{1}(|X|>R)]\\
&\leq R\mathbb{E}[|X|]+\mathbb{E}[X^2\mathbb{1}(|X|>R)].
\end{aligned}$$

Consequently,

$$\mathbb{E}[|X|]\geq\left(\epsilon B-\mathbb{E}[X^2\mathbb{1}(|X|>R)]\right)/R.$$

Notice that

$$\begin{aligned}
\mathbb{E}[X^2\mathbb{1}(|X|>R)]&=\int_R^\infty y\Pr[|X|\geq y]dy\\
&\leq\int_R^\infty ye^{-y^2/(2B)}dy\\
&=2Be^{-R^2/(2B)}.
\end{aligned}$$

Substituting above, we obtain that

$$\mathbb{E}[|X|]\geq\frac{B}{R}\left(\epsilon-2e^{-R^2/(2B)}\right)$$

Substituting $R = \lambda\sqrt{B}$, we obtain

$$\mathbb{E}[|X|] \geq \frac{\sqrt{B}}{\lambda}\left(\epsilon - 2e^{-\lambda^2/2}\right) \geq \frac{\epsilon\sqrt{B}}{4\sqrt{\log(1/\epsilon)}}$$

where the last inequality holds for $\epsilon \leq 0.25$ when setting $\lambda = 2\sqrt{\log(1/\epsilon)}$. □

**Lemma A.7** (Case 6)**.** *Let $a \in [N]$ such that $a \neq a_v, \dot{a}_v$ for any nodes $v$ in the tree. Then,*

$$4D\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \mathrm{Swap}(a \to a')\right] \geq \sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a_v]\right] \tag{71}$$

*Proof of Lemma A.7.* We have

$$\begin{aligned}
\mathbb{E}_{r^{(T)}}\left[\max_{a' \in [N]} \mathrm{Swap}(a \to a')\right] &\geq \mathbb{E}_{r^{(T)}}\left[\frac{1}{2}\left(\mathrm{Swap}(a \to a_\emptyset) + \mathrm{Swap}(a \to \dot{a}_\emptyset)\right)\right] \\
&= \frac{1}{2}\sum_{t=1}^{T} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a]\left(\mathbf{u}^{(t)}[a_\emptyset] + \mathbf{u}^{(t)}[\dot{a}_\emptyset] - 2\mathbf{u}^{(t)}[a]\right)\right] \\
&= \frac{1}{4D}\sum_{t=1}^{T'} \mathbb{E}_{r^{(T)}}\left[\mathbf{x}^{(t)}_{r^{(t)}}[a]\right]
\end{aligned}$$

which gives (71), as desired. □

# B Dimensions of games and function classes

In this section, we review the definitions of sequential complexity measures for real-valued *function classes* $\mathcal{H}$, namely a set of *concepts* $h : \mathcal{Z} \to \mathcal{Y}$, where $\mathcal{Z}$ is called the *domain set* and $\mathcal{Y}$ is called the *label set*.

**Trees.** For a set $\mathcal{Z}$, an *$\mathcal{Z}$-valued tree* $\mathcal{T}$ of *depth* $d$ is a complete rooted binary tree $\mathcal{T}$ each of whose nodes are labeled by an internal node. Each node of the tree is associated to a sequence $(\epsilon_1, \ldots, \epsilon_t) \in \{-1, 1\}^{t-1}$, describing the root to leaf path for that node (i.e., with $+1$ corresponding to 'right' and $-1$ corresponding to 'left'). Accordingly, the tree $\mathcal{T}$ may be specified by a sequence $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_d)$ of functions $\mathbf{z}_t : \{-1, 1\}^{t-1} \to \mathcal{Z}$, for $t \in [d]$, where $\mathbf{z}_t(\epsilon_{1:t-1})$ denotes the label of the node $\epsilon_{1:t-1}$.

**Definition B.1** (Sequential Rademacher complexity)**.** For a real-valued function class $\mathcal{H} : \mathcal{Z} \to \mathbb{R}$ and an integer $T \in \mathbb{N}$, its *sequential Rademacher complexity* (at depth $T$) is defined as

$$\mathfrak{R}_T(\mathcal{H}) := \sup_{\mathbf{z}} \mathbb{E}\left[\sup_{h \in \mathcal{H}} \frac{1}{T}\sum_{t=1}^{T} \epsilon_t h(\mathbf{z}_t(\epsilon_{1:t-1}))\right], \tag{72}$$

where the expectation is over i.i.d. Rademacher sequences $\epsilon_1, \ldots, \epsilon_T \sim \mathsf{Unif}(\{-1, 1\})$, and the supremum is over all $\mathcal{Z}$-valued trees $\mathbf{z}$ of depth $T$.

The sequential Rademacher complexity is known to tightly characterize the sample complexity of (agnostically) online learning a class $\mathcal{H}$. In particular, the minimax external regret can be upper bounded as follows:

**Theorem B.2** (Theorem 7 of [RST14][15]). *For any function class $\mathcal{H} \subset \mathbb{R}^{\mathcal{Z}}$, there is a (randomized) algorithm which, for any adaptive adversary choosing a sequence $\mathbf{z}^1, \dots, \mathbf{z}^{(T)} \in \mathcal{Z}$, produces a sequence of distributions $\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(T)} \in \Delta(\mathcal{H})$ so that*

$$\textbf{ExtRegret}(\mathbf{q}^{(1:T)}, \mathbf{z}^{(1:T)}) = \sup_{h^\star \in \mathcal{H}} \frac{1}{T} \sum_{t=1}^{T} \left( h^\star(\mathbf{z}^{(t)}) - \mathbb{E}_{h^{(t)} \sim \mathbf{q}^{(t)}}[h(\mathbf{z}^{(t)})] \right) \le 2\mathfrak{R}_T(\mathcal{H}). \tag{73}$$

**Combinatorial complexity measures.** As a corollary of Theorem B.2, the external regret for a function class may be upper bounded by combinatorial complexity measures. We first consider the binary case, for which the relevant complexity measure is the *Littlestone dimension*.

**Definition B.3** (Littlestone Dimension). For a function class $\mathcal{H}$ with domain $\mathcal{Z}$ and binary label set $\mathcal{Y} = \{0, 1\}$, define its Littlestone Dimension $\mathrm{LDim}(\mathcal{F})$ to be the maximum depth $d$ of a $\mathcal{Z}$-valued tree $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_d)$ so that, for all $\epsilon_{1:d} \in \{-1, 1\}^d$, there is some $h \in \mathcal{H}$ so that $h(\mathbf{z}_t(\epsilon_{1:t-1})) = \epsilon_t$ for each $t \in [d]$.

The analogue of Littlestone dimension for real-valued function classes, is the *sequential fat-shattering dimension*:

**Definition B.4** ($\delta$-Sequential Fat Shattering Dimension). For a function class $\mathcal{H}$ with domain $\mathcal{Z}$ and label set $\mathcal{Y} = \mathbb{R}$, denote its $\delta$-sequential fat shattering dimension $\mathrm{SFat}(\mathcal{H}, \delta)$ is the maximum integer $d$ so that there are complete binary trees $\mathbf{s}, \mathbf{z}$ of depth $d$ so that for all $\epsilon_{1:d} \in \{-1, 1\}^d$, there is some $h \in \mathcal{H}$ so that

$$\epsilon_t \cdot (h(\mathbf{z}_t(\epsilon_{1:t-1})) - \mathbf{s}_t(\epsilon_{1:t-1})) \ge \delta.$$

The following result, which upper bounds sequential Rademacher complexity in terms of the Littlestone and sequential fat-shattering dimensions, may be combined with Theorem B.2 to obtain an upper bound on the external regret in terms of the respective combinatorial complexity measures:

**Proposition B.5** (Proposition 18 of [BDR21] & Proposition 9 of [RST14]). *Consider a function class $\mathcal{H} \subset \mathbb{R}^{\mathcal{Z}}$. Then:*

- *If $\mathcal{H}$ is binary-valued (i.e., $\mathcal{H} \subset \{0, 1\}^{\mathcal{Z}}$), then $\mathfrak{R}_T(\mathcal{H}) \le \sqrt{\mathrm{LDim}(\mathcal{H})/T}$.*
- *If $\mathrm{SFat}(\mathcal{H}, \delta) \le O(\delta^{-p})$ for some $p \in [0, 2)$, then $\mathfrak{R}_T(\mathcal{H}) \le O(1/\sqrt{T}) \cdot \int_0^1 \sqrt{\mathrm{SFat}(\mathcal{H}, \delta)} d\delta$.*
- *In general, $\mathfrak{R}_T(\mathcal{H}) \le O\left( \alpha + \frac{1}{\sqrt{T}} \cdot \int_\alpha^1 \sqrt{\mathrm{SFat}(\mathcal{H}, \delta) \log(T/\delta)} d\delta \right)$ for any $\alpha \in (0, 1)$.*

[15] The minimax regret as defined in [RST14] is defined slightly differently to the expression in (73), in that the adversary in [RST14] can observe the draws $h^{(t)} \sim \mathbf{q}^{(t)}$ before choosing $\mathbf{z}^{(t+1)}$. It is straightforward to see that the two are equivalent.

**Complexity measures for games.** The complexity measures for function classes introduced above may be extended to games in the intuitive way. Consider an $m$-player game $(S, A)$, where $S = S_1 \times \cdots \times S_m$ denotes the joint action set, and $A_j : S \to \mathbb{R}$ denotes player $j$'s payoff function.

For each player $j$, we define a function class $\mathcal{X}_j : S_j \to \mathbb{R}$ as follows: for each action profile of the other players $s_{-j} \in S_{-j}$, define $f_{s_{-j}}(s_j) := A_j(s_j, s_{-j})$, for $s_j \in S_j$. Then set

$$\mathcal{X}_j = \{f_{s_{-j}} \ : \ s_{-j} \in S_{-j}\},$$

so that $\mathcal{X}_j$ is indexed by $S_{-j}$. For a given complexity measure, we define its value for the game $(S, A)$ to be its maximum value over the functions classes $\mathcal{X}_1, \ldots, \mathcal{X}_m$.

**Definition B.6** (Complexity measures for multiplayer games)**.** Let $(S, A)$ be an $m$-player game. If $A_j : S \to \{0, 1\}$ is binary-valued for all players, we define

$$\mathrm{LDim}(S, A) = \max_j \mathrm{LDim}(\mathcal{X}_j)$$

and if the game is real-valued, define

$$\mathrm{SFat}(S, A, \delta) = \max_j \mathrm{SFat}(\mathcal{X}_j, \delta)$$
$$\mathfrak{R}_T(S, A) = \max_j \mathfrak{R}_T(\mathcal{X}_j).$$

# C An adaptive lower bound on the swap regret

In this section, we present Theorem C.1, which gives an alternative lower bound on the swap regret. For simplicity, we focus on the setting when $N \geq T$, for which we obtain a lower bound of $\Omega(\log^{-3} T)$ on the swap regret; our techniques readily extend to obtain a lower bound that scales as $\Omega\left(\frac{\sqrt{N/T}}{\log^{O(1)} N}\right)$ when $T > N$.

Compared to Theorem 4.1, the adversary established by Theorem C.1 is *adaptive* and does not satisfy the property that its reward vectors $\mathbf{u}^{(t)}$ have $\ell_1$ norm bounded above by 1. On the other hand, the bound obtained by Theorem C.1 is quantitatively stronger than Theorem 4.1: in the regime $T \leq N$, Theorem 4.1 obtains a lower bound of $\Omega(\log^{-5} T)$, which is smaller than the bound of $\Omega(\log^{-3} T)$ of Theorem C.1.

**Theorem C.1.** *Fix any $T \in \mathbb{N}$. Then, for any learning algorithm on $N = T$ actions, there is an adaptive adversary guaranteeing that the swap regret is bounded below by*

$$\mathbf{SwapRegret}(T) \geq \Omega\left(\frac{1}{\log^3 T}\right).$$

**Proof overview for Theorem C.1.** Roughly speaking, the adversary constructs a sequence of "template" utility vectors $\mathbf{u}_{\mathsf{base}}^{(t)} \in [-1, 1]^N$, for $t \in [T]$ (defined formally in (74)). The vector $\mathbf{u}_{\mathsf{base}}^{(1)}$ is a monotonically decreasing vector, where the entries decrease by $\Delta = O(1/\log N)$ on a logarithmic scale: in particular, the first 2 entries are equal to 1, the next $2 \cdot 2^0$ entries are equal to $1 - \Delta$, the next $2 \cdot 2^1$ entries are equal to $1 - 2\Delta$, the next $2 \cdot 2^3$ entries are equal to $1 - 3\Delta$, and so on. Then,

$\mathbf{u}_{\mathsf{base}}^{(t)}$ is a shift of $\mathbf{u}_{\mathsf{base}}^{(1)}$ rightward by $2 \cdot (t-1)$ positions, with entries on the left filled in with $-1$. For each $t$, we set $\mathbf{u}^{(t)}$ to be equal to $\mathbf{u}_{\mathsf{base}}^{(t)}$ with some modifications, which we proceed to describe.

At a high level, on round $t$, the learner is best off by playing either action $2t-1$ or $2t$ (since both have utility equal to 1 for $\mathbf{u}_{\mathsf{base}}^{(t)}$). To make the learner "pay" for doing so, one of $\mathbf{u}^{(t)}[2t-1], \mathbf{u}^{(t)}[2t]$ is randomly perturbed by a small amount (namely, $\Delta/2$) for each round $t$, so if the learner spends only $O(1)$ rounds playing actions $2t-1, 2t$, they will incur $\Omega(1)$ swap regret for the actions $2t-1, 2t$. In particular, swapping either $2t-1$ to $2t$ or $2t$ to $2t-1$ will yield $\Omega(1)$ swap regret. Adding this quantity up over all actions would yield $\Omega(T)$ swap regret (this corresponds to Case 2 in the proof below).

However, the learner could attempt to proceed more cleverly so as to minimize the contribution of the random perturbations to their swap regret: suppose they partition $[T]$ into "moderate-sized" sub-intervals, within each of which they play a fixed action $a$ with utility close to 1 throughout the duration of that sub-interval. If they do so, then they will nevertheless typically incur swap regret $\Omega(\Delta)$ per round due to the ability to swap to some action $2s-1 < a$ during rounds $t \le s$. This lower bound crucially uses the logarithmic scaling of the utilities $\mathbf{u}^{(t)}$.

While the above-described adversary is oblivious, it does not quite rule out small swap regret: indeed, a "trivial" learner which plays some fixed action $a$ for all $t$ rounds will obtain small swap regret against the above-described adversary. To rule such a learner out, whenever an action in $\{2s-1, 2s\}$ (for any $s \in [N/2]$) has been played for $\Omega(1)$ rounds, the adversary sets the utility of $s$ to be $-1$ at all future rounds. In this way, even if the learner decides to play $2s-1$ or $2s$ in later rounds, it incurs large swap regret for not switching to some action $a' > 2s$. This latter modification leads the proof to be somewhat technical, as we need to carefully account for actions which have been "switched" to $-1$ in this manner.

*Proof of Theorem C.1.* Fix $T \in \mathbb{N}$. Fix any learning algorithm $\mathscr{A}$ which, at each step $t \in [T]$, outputs a distribution $\mathbf{x}^{(t)} \in \Delta_N$. We will define an adaptive adversary which generates reward vectors $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(T)} \in [0,1]^N$. At each round $t$, the output of the learner at step $t$ may be described by some function $\mathscr{A}_t(\mathbf{x}^{(1:t-1)}, \mathbf{u}^{(1:t-1)}) \in \Delta_{\Delta_N}$ (namely, $\mathscr{A}_t(\mathbf{x}^{(1:t-1)}, \mathbf{u}^{(1:t-1)})$ is a probability distribution over vectors in $\Delta_N$). We define

$$\mathbf{p}^{(t)}(\mathbf{x}^{(1:t-1)}, \mathbf{u}^{(1:t-1)}) := \mathbb{E}_{\mathbf{x} \sim \mathscr{A}_t(\mathbf{x}^{(1:t-1)}, \mathbf{u}^{(1:t-1)})}[\mathbf{x}].$$

We will often abbreviate $\mathbf{p}^{(t)}(\mathbf{x}^{(1:t-1)}, \mathbf{u}^{(1:t-1)})$ as $\mathbf{p}^{(t)}$. Let $\mathcal{F}^{(t)} := \sigma(\{\mathbf{x}^{(1:t)}, \mathbf{u}^{(1:t)}\})$ be the sigma algebra generated by the learner's and adversary's plays up to round $t$. Thus $\mathbf{p}^{(t)}$ is $\mathcal{F}^{(t-1)}$-measurable.

**Construction of the adversary.** Given $T \in \mathbb{N}$, define $L = \lfloor \log(T/2) \rfloor$ and $\Delta := 1/L$. We set $N := 2 \cdot (1 + 2 + \cdots + 2^{L-1}) \le T$. For $0 \le i < N$, define $F_{\mathsf{base}}(i) := \lfloor \log(1 + \frac{i}{2}) \rfloor \cdot \Delta$, where the logarithm is taken base 2. Note that $F_{\mathsf{base}}(0) = 0$, $F_{\mathsf{base}}(2 \cdot (2^j - 1)) = \Delta j$ for integers $j \ge 1$, and $F_{\mathsf{base}}(N-2) = L = (L-1)\Delta = 1 - \Delta$. For $t \in [T]$, define $\mathbf{u}_{\mathsf{base}}^{(t)} \in [-1,1]^N$ as follows: for $a \in [N]$,

$$\mathbf{u}_{\mathsf{base}}^{(t)}[a] := \begin{cases} 1 - F_{\mathsf{base}}(a - (2t-1)) & : a \ge 2t-1 \\ -1 & : a < 2t-1. \end{cases} \tag{74}$$

In words, $\mathbf{u}_{\mathsf{base}}^{(t)}$ is a shift of the function $1 - F_{\mathsf{base}}(\cdot)$ rightward by $2t-1$ units, with entries before $2t-1$ all set to $-1$.

Fix $\zeta = 1/(32L)$. We pair up each action $2t-1$ with action $2t$, for $t \in [N/2]$: for $a \in [N]$, we let $\mathfrak{p}(a)$ denote its pair. Moreover, write $\bar{\mathbf{p}}^{(t)}[a] := \mathbf{p}^{(t)}[a] + \mathbf{p}^{(t)}[\mathfrak{p}(a)]$ (so that $\bar{\mathbf{p}}^{(t)}[a] = \bar{\mathbf{p}}^{(t)}[\mathfrak{p}(a)]$ for all $a \in [N]$). We now define the reward vectors $\mathbf{u}^{(t)}$ chosen by the adversary (as a function of $\mathbf{p}^{(t)}$), as follows: let $r^{(1)}, \dots, r^{(N/2)} \in \{0,1\}$, denote a sequence of independent and uniformly distributed bits. For each $t \le N/2$, $a \in [N]$ and $0 \le k < L$, define the sets $\mathcal{G}^{(t)}, \mathcal{S}^{(t)}(a,k) \subset [t]$ and real numbers $\sigma^{(t)}(a,k), \Sigma^{(t)}(a) \ge 0$ recursively with respect to $t$, as follows:

$$\begin{aligned}
\mathcal{G}^{(t)} :=& \{s \in [t] \ : \ \Sigma^{(s-1)}(2s-1) < \zeta\} \\
\mathcal{S}^{(t)}(a,k) :=& \left\{ s \in \mathcal{G}^{(t)} \ : \ a > 2s, \ \left\lfloor \log\left(\frac{a-(2s-1)}{2}\right) \right\rfloor = k \right\} \\
\sigma^{(t)}(a,k) :=& \sum_{s \in \mathcal{S}^{(t)}(a,k)} \bar{\mathbf{p}}^{(s)}[a], \qquad \Sigma^{(t)}(a) := \max_{0 \le k < L} \sigma^{(t)}(a,k).
\end{aligned}$$

Note that $\Sigma^{(t-1)}(2t-1) = \Sigma^{(t-1)}(2t)$ by our definition of $\bar{\mathbf{p}}^{(t)}$. We write $\mathcal{G} := \mathcal{G}^{(N/2)}$. Intuitively, the meaning of $\mathcal{G}^{(t)}, \mathcal{S}^{(t)}(a,k), \sigma^{(t)}(a,k), \Sigma^{(t)}(a)$ are as follows:

- $\mathcal{G}^{(t)}$ denotes the set of *active* rounds $s$ up to round $t$: a round $s$ is active, if, roughly speaking, actions $2s-1, 2s \in [N]$ have not been played too much by the algorithm $\mathscr{A}$ up to round $s$.
- $\mathcal{S}^{(t)}(a,k)$ denotes the set of active rounds $s$ up to step $t$ for which $\lfloor \frac{a-(2s-1)}{2} \rfloor$ is in $\{2^k, 2^k+1, \dots, 2^{k+1}-1\}$. Since $F_{\mathsf{base}}(2j) \ge \Delta + F_{\mathsf{base}}(2(j-2^k))$ whenever $j \in \{2^k, 2^k+1, \dots, 2^{k+1}-1\}$, it follows that for all $s \in \mathcal{S}^{(t)}(a,k)$, for any $b$ satisfying $a/2 > b \ge 2^k$, $\mathbf{u}^{(s)}_{\mathsf{base}}[a-2b] - \mathbf{u}^{(s)}_{\mathsf{base}}[a] = F_{\mathsf{base}}(a-(2s-1)) - F_{\mathsf{base}}(a-2b-(2s-1)) \ge \Delta$. Since $s' \in \mathcal{S}^{(t)}(a,k)$ implies that $\frac{a-(2s'-1)}{2} \ge 2^k$, it follows by choosing $b = \frac{a-(2s'-1)}{2}$ that, for any $s, s' \in \mathcal{S}^{(t)}(a,k)$,

$$\mathbf{u}^{(s)}_{\mathsf{base}}[2s'-1] - \mathbf{u}^{(s)}_{\mathsf{base}}[a] \ge \Delta. \tag{75}$$

- $\sigma^{(t)}(a,k)$ denotes the aggregate amount of mass that $\mathscr{A}$ puts on action $a$ in rounds in $\mathcal{S}^{(t)}(a,k)$.
- $\Sigma^{(t)}(a)$ denotes the maximum amount of mass that $\mathscr{A}$ puts on action $a$ in any of the sets $\mathcal{S}^{(t)}(a,k)$.

We will now define $\mathbf{u}^{(t)} \in [-1,1]^N$ as follows: if $t \notin \mathcal{G}^{(t)}$, then we set $\mathbf{u}^{(t)} = 0$. (Note that $\mathcal{G}^{(t)}$ only depends on $\mathbf{p}^{(s)}$ for $s < t$, so this operation is well-defined.) Otherwise, we define

$$\mathbf{u}^{(t)}[a] := \begin{cases} \mathbf{u}^{(t)}_{\mathsf{base}}[a] = -1 & : a < 2t-1 \\ \mathbf{u}^{(t)}_{\mathsf{base}}[a] - \frac{\Delta}{2} \cdot \mathbb{1}\{r^{(t)} \equiv a \pmod 2\} & : a \in \{2t-1, 2t\} \\ \mathbf{u}^{(t)}_{\mathsf{base}}[a] & : a > 2t, \ \Sigma^{(t-1)}(a) < \zeta \\ -1 & : a > 2t, \Sigma^{(t-1)}(a) \ge \zeta. \end{cases} \tag{76}$$

Finally, for $N/2 < t \le T$, define $\mathbf{u}^{(t)}[a] = -1$ for all $a \in [N]$.

**Proof of regret lower bound.** For each $a \in [N]$, define

$$\tau(a) := \min\{t \in [N/2] \ : \ a < 2t-1 \text{ or } \Sigma^{(t-1)}(a) \ge \zeta\}.$$

Roughly speaking, $\tau(a)$ denotes the first round at which either $2t-1$ exceeds $a$ or else $a$ is played "too much" by $\mathscr{A}$ in the sense that $\Sigma^{(t-1)} \geq \zeta$. We say that an action $a$ is *stale* at round $t$ if $t \geq \tau(a)$. Note that for all $t \in \mathcal{G}$ with $t > \tau(a)$, we have $\mathbf{u}^{(t)}[a] = -1$. We define the following quantities, for $a \in [N]$:

$$P_-(a) := \sum_{t \in \mathcal{G}:\ t < \tau(a), a > 2t} \bar{\mathbf{p}}^{(t)}[a], \qquad P_0(a) := \sum_{t \in \mathcal{G}:\ a \in \{2t-1, 2t\}} \bar{\mathbf{p}}^{(t)}[a], \qquad P_+(a) := \sum_{t \in \mathcal{G}:\ t \geq \tau(a)} \bar{\mathbf{p}}^{(t)}[a].$$

$P_-(a)$ denotes the total mass placed on $a$ and $\mathfrak{p}(a)$ in all rounds when $a$ is active except $2t-1, 2t$, $P_0(a)$ denotes the mass places on $a$ and $\mathfrak{p}(a)$ during the rounds $2t-1, 2t$, and $P_+(a)$ denotes the mass placed on $a$ and $\mathfrak{p}(a)$ in the remaining rounds.

Also write $P(a) = P_-(a) + P_0(a) + P_+(a)$ and $P := \sum_{a \in [N]} P(a)$. Finally, we define

$$\mathcal{A}_0 := \left\{ a \in [N] \ :\ \Sigma^{(t-1)}(a) \geq \zeta \text{ for } t = \left\lfloor \frac{a+1}{2} \right\rfloor \right\}, \quad \mathcal{A}_1 := \{ a \in [N] \ :\ P(a) \geq 4\zeta L \}, \quad \mathcal{A} := \mathcal{A}_0 \cup \mathcal{A}_1.$$

$\mathcal{A}_0$ denotes the set of actions $a$ which have become stale at some point prior to the unique round $t$ for which $a \in \{2t-1, 2t\}$. $\mathcal{A}_1$ denotes the set of actions $a$ which are "played a lot" by $\mathscr{A}$ over all rounds in $\mathcal{G}$. We next state the following claim, whose proof is provided following the proof of the theorem.

**Claim C.2.** *It holds that* $\sum_{a \in \mathcal{A}} P(a) \geq \zeta N/4$.

Consider any $a \in \mathcal{A}$. One of the below cases must hold:

**Case 1:** $P_-(a) \geq P(a)/4$**.** We claim that in fact $a \in \mathcal{A}_0$. To see this, suppose not, which means that $a \in \mathcal{A}_1$, and let $\tau'(a) < \tau(a)$ denote the largest integer $t \in \mathcal{G}$ which is strictly less than $\tau(a)$. Then, letting $t_a = \lfloor (a+1)/2 \rfloor$,

$$\zeta L \leq P(a)/4 \leq P_-(a) = \sum_{k=0}^{L-1} \sigma^{(\tau'(a))}(a,k) = \sum_{k=0}^{L-1} \sigma^{(t_a - 1)}(a,k). \tag{77}$$

The first equality above uses the fact that each $s \in \mathcal{G}^{(t)}$ must belong to (exactly) one of the sets $\mathcal{S}^{(t)}(a,k)$, for $0 \leq k < L$. Thus there is some $0 \leq k_a < L$ so that $\sigma^{(t_a-1)}(a,k_a) = \sigma^{(\tau'(a))}(a,k_a) \geq \zeta$, which implies that $\Sigma^{(t_a-1)}(a) \geq \zeta$, i.e., $a \in \mathcal{A}_0$, thus establishing our claim.

We remark for later use that by definition of $\tau'(a)$, $\sigma^{(\tau'(a))}(a,k_a)$ must in fact be the largest of the values $\sigma^{(\tau'(a))}(a,k)$, for $k < L$ (as otherwise there would be some $t \leq \tau'(a)$ so that $\Sigma^{(t-1)}(a) \geq \zeta$, contradicting the definition of $\tau(a)$). We may now write the regret for not swapping actions $a$ and $\mathfrak{p}(a)$ to another action $a'$ as follows:

$$\begin{aligned}
&\max_{a' \in [N]} \sum_{s=1}^{N/2} \left( \mathbf{p}^{(s)}[a] \cdot (\mathbf{u}^{(s)}[a'] - \mathbf{u}^{(s)}[a]) + \mathbf{p}^{(s)}[\mathfrak{p}(a)] \cdot (\mathbf{u}^{(s)}[a'] - \mathbf{u}^{(s)}[\mathfrak{p}(a)]) \right) \\
&\geq \max_{a' \in [N]} \sum_{s \in \mathcal{G},\ s \leq \tau'(a)} \bar{\mathbf{p}}^{(s)}[a] \cdot (\mathbf{u}^{(s)}[a'] - \mathbf{u}^{(s)}[a]) \\
&\geq \sum_{s \in \mathcal{S}^{(\tau'(a))}(a,k_a)} \bar{\mathbf{p}}^{(s)}[a] \cdot (\mathbf{u}^{(s)}[2\tau'(a)-1] - \mathbf{u}^{(s)}[a]) \geq \Delta \cdot \frac{P(a)}{8L},
\end{aligned} \tag{78}$$

where the first inequality holds because all $s \notin \mathcal{G}$ have $\mathbf{u}^{(s)} = \mathbf{0}$, for $s \in \mathcal{G}$ with $s > \tau'(a)$ (and thus $s \geq \tau(a)$), we have $\mathbf{u}^{(s)}[a] = -1$, and for $s \in \mathcal{G}$ with $s \leq \tau'(a)$, we have $\mathbf{u}^{(s)}[a] = \mathbf{u}^{(s)}[\mathfrak{p}(a)]$. The second inequality follows by the choice of $a' = 2\tau'(a) - 1$ together with the fact that, since $\tau'(a) \in \mathcal{G}$, for $s \in \mathcal{G}$ with $s \leq \tau'(a)$, we have $\Sigma^{(s-1)}(2\tau'(a) - 1) < \zeta$ and thus $\mathbf{u}^{(s)}[2\tau'(a) - 1] = \mathbf{u}^{(s)}_{\mathsf{base}}[2\tau'(a) - 1] \geq \mathbf{u}^{(s)}_{\mathsf{base}}[a] \geq \mathbf{u}^{(s)}[a]$.

Finally, the third inequality in (78) follows because $\sigma^{(\tau'(a))}(a, k_a) \geq P_-(a)/L \geq P(a)/(4L)$ from (77) and $\mathbf{u}^{(s)}[2\tau'(a) - 1] - \mathbf{u}^{(s)}[a] \geq \Delta$ for all $s \in \mathcal{S}^{(\tau'(a))}(a, k_a)$. To see this latter implication, first note that $\mathbf{u}^{(s)}_{\mathsf{base}}[2\tau'(a) - 1] - \mathbf{u}^{(s)}_{\mathsf{base}}[a] \geq \Delta$ by (75) and the fact that $\tau'(a) \in \mathcal{S}^{(\tau'(a))}(a, k_a)$ (since $\sigma^{(\tau'(a))}(a, k_a)$ must surpass $\zeta$ during iteration $\tau'(a)$, and the only way for this to happen is that $\tau'(a) \in \mathcal{S}^{(\tau'(a))}(a, k_a)$). Then we note that, by definition of $\tau'(a)$, $s \in \mathcal{S}^{(\tau'(a))}(a, k_a)$ satisfies $\mathbf{u}^{(s)}_{\mathsf{base}}[a] = \mathbf{u}^{(s)}[a]$ (since $\tau'(a) < \tau(a)$) and $\mathbf{u}^{(s)}_{\mathsf{base}}[2\tau'(a) - 1] - \frac{\Delta}{2} \leq \mathbf{u}^{(s)}[2\tau'(a) - 1]$. Then

$$\mathbf{u}^{(s)}[2\tau'(a) - 1] - \mathbf{u}^{(s)}[a] \geq -\frac{\Delta}{2} + \mathbf{u}^{(s)}_{\mathsf{base}}[2\tau'(a) - 1] - \mathbf{u}^{(s)}_{\mathsf{base}}[a] \geq \frac{\Delta}{2}.$$

**Case 2:** $P_0(a) \geq P(a)/12$**.** By replacing $a$ with its pair $\mathfrak{p}(a)$ if necessary, we may assume that $\sum_{t \in \mathcal{G}:\ a \in \{2t-1, 2t\}} \mathbf{p}^{(t)}[a] \geq P(a)/24$. Let us further suppose that $a$ is odd (the case that $a$ is even is handled symmetrically). Write $t_a = \frac{a+1}{2}$, and note that, since for all $t \neq t_a$, we have $\mathbf{u}^{(t)}[a] = \mathbf{u}^{(t)}[a+1]$,

$$\sum_{t \in \mathcal{G}} \mathbf{p}^{(t)}[a] \cdot (\mathbf{u}^{(t)}[a+1] - \mathbf{u}^{(t)}[a]) = \frac{\Delta}{2} \cdot \mathbf{p}^{(t_a)}[a] \cdot (2r^{(t_a)} - 1).$$

Thus, the expected swap regret for action $a$ may be lower bounded by

$$\mathbb{E}\left[\max\left\{0, \frac{\Delta}{2} \cdot \mathbf{p}^{(t_a)}[a] \cdot (2r^{(t_a)} - 1)\right\} \mid \mathcal{F}^{(t_a - 1)}\right] = \frac{\Delta}{2} \cdot \mathbf{p}^{(t_a)}[a] \geq \frac{\Delta \cdot P(a)}{48}. \tag{79}$$

**Case 3:** $P_+(a) \geq 2P(a)/3$**.** Note that for all $t \in \mathcal{G}$ with $t > \tau(a)$, we have $\mathbf{u}^{(t)}[a] = -1$. Let $a^\star \in [N]$ denote the largest action so that $2a^\star - 1 \in \mathcal{G}$, and suppose first that $a \notin \{a^\star, \mathfrak{p}(a^\star)\}$ (hence $a < a^\star$). Then the swap regret for actions $a, \mathfrak{p}(a)$ can be shown to be large by using the swap function which swaps to the action $a^\star > a$:

$$\begin{aligned}
&\max_{a' \in [N]} \sum_{s=1}^{N/2} \left(\mathbf{p}^{(s)}[a] \cdot (\mathbf{u}^{(s)}[a'] - \mathbf{u}^{(s)}[a]) + \mathbf{p}^{s}[\mathfrak{p}(a)] \cdot (\mathbf{u}^{(s)}[a'] - \mathbf{u}^{(s)}[\mathfrak{p}(a)])\right) \\
&= \max_{a' \in [N]} \sum_{s \in \mathcal{G}} \left(\mathbf{p}^{(s)}[a] \cdot (\mathbf{u}^{(s)}[a'] - \mathbf{u}^{(s)}[a]) + \mathbf{p}^{s}[\mathfrak{p}(a)] \cdot (\mathbf{u}^{(s)}[a'] - \mathbf{u}^{(s)}[\mathfrak{p}(a)])\right) \\
&\geq -\sum_{s \in \mathcal{G}: s \leq \tau(a)} \bar{\mathbf{p}}^{(s)}[a] + \sum_{s \in \mathcal{G}: s > \tau(a)} \bar{\mathbf{p}}^{(s)}[a] \\
&= P_+(a) - (P_0(a) + P_-(a)) \geq P(a)/3,
\end{aligned} \tag{80}$$

where the first equality uses that $\mathbf{u}^{(s)}[a] = -1$ for all $a$ and $s \notin \mathcal{G}$, and the inequality uses the fact that $\mathbf{u}^{(s)}[a^\star] - \mathbf{u}^{(s)}[a] \geq -1$ for all $s \in \mathcal{G}$ with $s \leq \tau(a)$ as well as the fact that for $s > \tau(a)$, we have $\mathbf{u}^{(s)}[a] = -1$ and $\mathbf{u}^{(s)}[a^\star] \geq 0$ since $s \in \mathcal{G}$ and all $s \in \mathcal{G}$ satisfy $\Sigma^{(s-1)}(a^\star) < \zeta$ (by our choice of $a^\star$ as large as possible so that $2a^\star - 1 \in \mathcal{G}$).

In the event that $a \in \{a^\star, \mathfrak{p}(a^\star)\}$, we may use the same argument as above with $a^\star$ instead being the *second-largest* action so that $2a^\star - 1 \in \mathcal{G}$, which gives a lower bound of $P(a)/3 - 2$ in (80). (The $-1$ is insignificant in our final lower bound on swap regret, since $\{a^\star, \mathfrak{p}(a^\star)\}$ only contains two actions.)

Combining the above cases, we obtain that

$$2 \cdot \mathbb{E}\left[\sum_{a\in\mathcal{A}} \max_{a'\in[N]} \sum_{s=1}^{N/2} \mathbf{p}^{(s)}[a] \cdot (\mathbf{u}^{(s)}[a'] - \mathbf{u}^{(s)}[a])\right] \geq -2 + \sum_{a\in\mathcal{A}} \frac{P(a)\Delta}{48L} \geq \frac{N\cdot\zeta\Delta}{48L} - 1,$$

where the factor of 2 on the left-hand side arises because in our arguments above we have double-counted each action $a$ with its $\mathfrak{p}(a)$ in Equations (78) to (80), and the final inequality uses Claim C.2. Thus the expected swap regret is bounded below by $\Omega(N/\log^3 N) = \Omega(T/\log^3 T)$.

□

*Proof of Claim C.2.* We consider two cases, depending on $|\mathcal{G}|$:

In the first case, we assume $|\mathcal{G}| < N/4$. Consider any $a \in \mathcal{A}_0$, and let $t_a = \lfloor \frac{a+1}{2} \rfloor$, so that $a \in \{2t_a - 1, 2t_a\}$. Since $a \in \mathcal{A}_0$, we must have $\Sigma^{(t_a-1)}(a) \geq \zeta$, i.e., $t_a \in [N/2]\backslash\mathcal{G}$. Since $[N/2]\backslash\mathcal{G}$ has size at least $N/4$ by assumption, we see that

$$\sum_{a\in\mathcal{A}_0} P(a) \geq \sum_{t\in[N/2]\backslash\mathcal{G}} \Sigma^{(t-1)}(2t-1) \geq \zeta N/4.$$

In the second case, we have $|\mathcal{G}| \geq N/4$. Then $\sum_{a\in[N]} P(a) = \sum_{a\in[N]} \sum_{t\in\mathcal{G}} \bar{\mathbf{p}}^{(t)}[a] \geq |\mathcal{G}| \geq N/4$. Since $4\zeta LN \leq N/8$ by our choice of $\zeta = 1/(32L)$, it follows that $\sum_{a\in\mathcal{A}_1} P(a) \geq N/4 - N/8 \geq \zeta N/4$. □

## References


[AAD+23] Angelos Assos, Idan Attias, Yuval Dagan, Constantinos Daskalakis, and Maxwell K Fishelson. Online learning and solving infinite games with an erm oracle. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 274–324. PMLR, 2023.

[ABD+21] Noga Alon, Omri Ben-Eliezer, Yuval Dagan, Shay Moran, Moni Naor, and Eylon Yogev. Adversarial laws of large numbers and optimal regret in online classification. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 2021.

[Aum74] Robert Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67–96, 1974.

[Bab20] Yakov Babichenko. Informational bounds on equilibria (a survey). *SIGecom Exch.*, 17(2):25–45, January 2020.

[BBD+22] Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sasha Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of Diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

[BBP14] Yakov Babichenko, Siddharth Barman, and Ron Peretz. Simple approximate equilibria in large games. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC '14, pages 753–770, Palo Alto, California, USA. Association for Computing Machinery, 2014. ISBN: 9781450325653.

[BC+12] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[BDR21] Adam Block, Yuval Dagan, and Alexander Rakhlin. Majorizing measures, sequential complexities, and online learning. In *Conference on Learning Theory*, pages 587–590. PMLR, 2021.

[BM07] Avrim Blum and Yishay Mansour. From external to internal regret. *J. Mach. Learn. Res.*, 8:1307–1324, 2007.

[BPS09] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT*, volume 3, page 1, 2009.

[BS19] Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365(6456):885–890, 2019.

[CDT09] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 2009.

[CL06] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[CMBG19] Andrea Celli, Alberto Marchesi, Tommaso Bianchi, and Nicola Gatti. Learning to correlate in multi-player general-sum sequential games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[DG22] Constantinos Daskalakis and Noah Golowich. Fast rates for nonparametric online learning: from realizability to learning in games. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 846–859, 2022.

[DGP09] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1), 2009.

[FKS21] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Better regularization for sequential decision spaces: fast convergence rates for nash, correlated, and team equilibria. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, EC '21, page 432, Budapest, Hungary. Association for Computing Machinery, 2021. ISBN: 9781450385541.

[FL98] Drew Fudenberg and David Levine. *The Theory of Learning in Games*. MIT Press, 1998.

[FLLK22] Gabriele Farina, Chung-Wei Lee, Haipeng Luo, and Christian Kroer. Kernelized multiplicative weights for 0/1-polyhedral games: bridging the gap between learning in extensive-form and normal-form games. 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 6337–6357. PMLR, 17–23 Jul 2022.

[FP23] Gabriele Farina and Charilaos Pipis. Polynomial-time linear-swap regret minimization in imperfect-information sequential games. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[GHK+23] Ira Globus-Harris, Declan Harrison, Michael Kearns, Aaron Roth, and Jessica Sorrell. Multicalibration as boosting for regression. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23, Honolulu, Hawaii, USA. JMLR.org, 2023.

[GPM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[Han57] James Hannan. Approximation to Bayes risk in repeated play. In volume III, Contributions to the Theory of Games, pages 97–139. Princeton University Press, 1957.

[Ito20] Shinji Ito. A tight lower bound and efficient reduction for swap regret. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18550–18559. Curran Associates, Inc., 2020.

[JLWY22] Chi Jin, Qinghua Liu, Yuanhao Wang, and Tiancheng Yu. V-learning – a simple, efficient, decentralized algorithm for multiagent RL. In *ICLR 2022 Workshop on Gamification and Multiagent Solutions*, 2022.

[KLST23] Bobby Kleinberg, Renato Paes Leme, Jon Schneider, and Yifeng Teng. U-calibration: forecasting for an unknown agent. In Gergely Neu and Lorenzo Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 5143–5145. PMLR, December 2023.

[KWKS20] Christian Kroer, Kevin Waugh, Fatma Kılınç-Karzan, and Tuomas Sandholm. Faster algorithms for extensive-form game solving via improved smoothing functions. *Mathematical Programming*, 2020.

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

[MMSS22] Yishay Mansour, Mehryar Mohri, Jon Schneider, and Balasubramanian Sivan. Strategizing against learners in bayesian games. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 5221–5252. PMLR, February 2022.

[PR23] Binghui Peng and Aviad Rubinstein. Fast swap regret minimization and applications to approximate correlated equilibria, 2023.

[RST14] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning via sequential complexities. *Journal of Machine Learning Research*, 2014.

[SL05] Gilles Stoltz and Gábor Lugosi. Internal regret in on-line portfolio selection. *Machine Learning*, 59(1-2):125–159, 2005.

[SSS16] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

[VF08] Bernhard Von Stengel and Françoise Forges. Extensive-form correlated equilibrium: definition and computational complexity. *Mathematics of Operations Research*, 33(4):1002–1022, 2008.

[ZJBP07] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

[ZTS+20] Stephan Zheng, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C Parkes, and Richard Socher. The ai economist: improving equality and productivity with ai-driven tax policies. *arXiv preprint arXiv:2004.13332*, 2020.