Efficient Phi-Regret Minimization in Extensive-Form Games via Online Mirror Descent

Yu Bai
Salesforce Research
yu.bai@salesforce.com

Chi Jin
Princeton University
chij@princeton.edu

Song Mei
UC Berkeley
songmei@berkeley.edu

Ziang Song
Stanford University
ziangs@stanford.edu

Tiancheng Yu
MIT
yutc@mit.edu

Abstract

A conceptually appealing approach for learning Extensive-Form Games (EFGs) is to convert them to Normal-Form Games (NFGs). This approach enables us to directly translate state-of-the-art techniques and analyses in NFGs to learning EFGs, but typically suffers from computational intractability due to the exponential blow-up of the game size introduced by the conversion. In this paper, we address this problem in natural and important setups for the Φ-*Hedge* algorithm—A generic algorithm capable of learning a large class of equilibria for NFGs. We show that Φ-Hedge can be directly used to learn Nash Equilibria (zero-sum settings), Normal-Form Coarse Correlated Equilibria (NFCCE), and Extensive-Form Correlated Equilibria (EFCE) in EFGs. We prove that, in those settings, the Φ-*Hedge* algorithms are equivalent to standard Online Mirror Descent (OMD) algorithms for EFGs with suitable dilated regularizers, and run in polynomial time. This new connection further allows us to design and analyze a new class of OMD algorithms based on modifying its log-partition function. In particular, we design an improved algorithm with balancing techniques that achieves a sharp $\widetilde{\mathcal{O}}(\sqrt{XAT})$ EFCE-regret under bandit-feedback in an EFG with X information sets, A actions, and T episodes. To our best knowledge, this is the first such rate and matches the information-theoretic lower bound.

1 Introduction

Extensive form games (EFGs) is a natural formulation for multi-player games with imperfect information and sequential play, which models real-world games such as Poker [9, 10], Bridge [45], Scotland Yard [41], Diplomacy [7] and has many other important applications such as cybersecurity [32], auction [39], marketing [29]. In multi-player general-sum EFGs, computing an approximate Nash equilibrium (NE) [40] is PPAD-hard [15] and thus likely intractable. A reasonable and computationally tractable solution concept in general-sum EFGs is the *extensive-form correlated equilibria* (EFCE) [46, 27, 13, 23]. It is known that, as long as each player runs an uncoupled dynamics minimizing a suitable EFCE-regret, their average joint policy will converge to an EFCE [28].

Existing algorithms of minimizing the EFCE-regret are mostly built upon the *regret decomposition* techniques [51], which utilize the structure of the game and the set of policy modifications [13, 38, 23, 42]. For example, Morrill et al. [38] decomposes the EFCE-regret to local regrets at each information set (infoset) with each of them handled by a local regret minimizer; Farina et al. [23] utilizes the trigger structure of the policy modification set to decompose the regret to external-like regrets.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

There are at least two alternative approaches to designing regret minimization algorithms for EFGs. The first is to convert a EFG to a normal-form game (NFG) and use NFG-based algorithms such as Φ-Hedge [28]. This approach typically admits simple algorithm designs and sharp regret bounds by directly translating existing results in NFGs [44]. However, the conversion introduces an exponential blow-up in the game size, and makes such algorithms computationally intractable in general. The computational efficiency of these NFG-based algorithms is recently investigated by Farina et al. [25] in the external regret minimization problem, who provided an efficient implementation of an NFG-based algorithm using "kernel tricks". The second is to use Online Mirror Descent (OMD) algorithms via suitably designed regularizers over the parameter space. This approach has been successfully implemented in minimizing the external regret [35] but not yet generalized to the EFCE-regret, as it remains unclear how to design suitable regularizers for the policy modification space.

In this paper, we develop the first line of EFCE-regret minimization algorithms along both lines of approaches above, and identify an equivalence between them. We consider EFCE-regret minimization in EFGs with X infosets, A actions, and maximum L_1-norm of sequence-form policies bounded by $\|\Pi\|_1$ (cf. Section 2.2 for the formal definition). Our contributions can be summarized as follows:

- We present an efficient implementation of the Φ-Hedge algorithm for minimizing the extensive-form trigger regret, by recursively evaluating the gradient of a log-partition function (Section 3.1). The implementation further reveals that this algorithm (via reparametrization) is equivalent to an OMD algorithm with dilated regularizers, which we term as EFCE-OMD (Section C.1).
- We show that EFCE-OMD achieves trigger regret bound $\widetilde{\mathcal{O}}(\sqrt{\|\Pi\|_1 T})$ under full feedback and $\widetilde{\mathcal{O}}(\sqrt{X\,\|\Pi\|_1 AT})$ under bandit feedback (Section 3.3). Notably, the proofs are done using the corresponding NFG analysis straightforwardly, and is independent of the actual implementation.
- We design an improved algorithm Balanced EFCE-OMD, and show that it achieves a sharp $\widetilde{\mathcal{O}}(\sqrt{XAT})$ trigger regret under bandit feedback (Section 4). This improves over EFCE-OMD by a factor of $\|\Pi\|_1$ and is the first to match the information-theoretic lower bound. The algorithm works by modifying the above log-partition function using a variety of *balancing* techniques, and is equivalent to another OMD algorithm (but no longer an NFG algorithm).
- As another example of our framework, we show that the Φ-Hedge algorithm for vanilla (external) regret minimization in EFGs, along with its efficient implementation via "kernelization" developed recently in [25], is actually equivalent to standard OMD with dilated entropy (Section 5).

1.1 Related work

Φ-regret minimization and correlated equilibrium The Φ-regret minimization framework was introduced in Greenwald and Jafari [28] and Stoltz and Lugosi [44]. In particular, Greenwald and Jafari [28] showed that uncoupled no Φ-regret dynamics leads to Φ-correlated equilibria, a generalized notion of correlated equilibria introduced by Aumann [5]. Stoltz and Lugosi [44] then developed a family of Φ-regret minimization algorithms using the fixed-point method (including the Φ-Hedge algorithm considered in this paper), and derived explicit regret bounds. Two important special cases of Φ-regret are the internal regret and swap regret in normal-form games [43, 8]. A recent line of work developed algorithms with $\mathcal{O}(\text{polylog}T)$ swap regret bound in normal-form games [2, 3].

Regret minimization in EFG from full feedback A line of work considers external regret minimization in EFGs from full feedback [51, 12, 11, 21, 50]. In particular, Zhou et al. [50] achieves $\widetilde{\mathcal{O}}(\sqrt{XT})$ external regret. The recent work of Farina et al. [25] develops the first algorithm to achieve $\widetilde{\mathcal{O}}(\|\Pi\|_1\, \text{polylog}T)$ external regret in EFGs by converting it to an NFG and invoking the fast rate of Optimistic Hedge [16], along with an efficient implementation via the "kernel trick". Our Φ-regret framework covers their algorithm as a special case, and we further show that their algorithm (along with its efficient implementation) is equivalent to the standard OMD with dilated entropy.

The notion of Extensive-Form Correlated Equilibria (EFCE) in EFGs was introduced in Von Stengel and Forges [46]. Optimization-based algorithms for computing computing EFCEs in multi-player EFGs from full feedback have been proposed in Huang and von Stengel [31], Farina et al. [20].

Gordon et al. [27] first proposed to use uncoupled EFCE-regret minimization dynamics to compute EFCE; however, they do not explain how to efficiently implement each iteration of the dynamics. Recent works [13, 23, 38, 42] developed uncoupled EFCE regret minimization learning dynamics with efficient implementation; All of these algorithms are based on counterfactual regret decomposition [51] and minimizing each trigger regret (first considered by Dudik and Gordon [17], Gordon et al. [27]) using a different regret minimizer. Celli et al. [13] decomposed the regret to each laminar subtree, but they did not give an explicit regret bound. Farina et al. [23] decomposed the regret to each trigger sequence and used CFR type algorithm to minimize the regret on each trigger sequence and achieved an $\tilde{O}(\sqrt{X^2T})$ EFCE-regret bound. Morrill et al. [38], Song et al. [42] decomposed the regret to each information set and use regret minimization algorithms with time-selection functions [8, 33] to minimize the regret on each information set, giving $\widetilde{\mathcal{O}}(\sqrt{X^2T})$ and $\widetilde{\mathcal{O}}(\sqrt{XT})$ regret bounds respectively. In this paper, we show that the simple Φ-Hedge algorithm, which has an efficient implementation and an intuitive interpretation, can also achieve the state-of-art $\widetilde{\mathcal{O}}(\sqrt{XT})$ regret bound in the full feedback setting.

Regret minimization in EFG from bandit feedback Minimizing the external regret in EFGs from bandit feedback is considered in a more recent line of work [36, 22, 19, 24, 49, 47, 34, 6]. Dudík and Gordon [18] consider sample-based learning of EFCE in succinct extensive-form games; however, their algorithm relies on an approximate Markov-Chain Monte-Carlo sampling subroutine that does not lead to a sample complexity guarantee.

A concurrent work by Song et al. [42] also achieves $\widetilde{\mathcal{O}}(X/\varepsilon^2)$ sample complexity for learning EFCE under bandit feedback (when only highlighting X) using the Balanced K-EFR algorithm. Our work achieves the same linear in X sample complexity, but using a very different algorithm (Balanced EFCE-OMD). We also remark that the algorithm of [42] cannot minimize the EFCE-regret against adversarial opponents from bandit feedback like our algorithm, as their algorithm requires playing multiple episodes against a fixed opponent, which is infeasible when the opponent is adversarial.

2 Preliminaries

2.1 Φ-regret minimization and Φ-Hedge algorithm

Consider a generic linear regret minimization problem on a *policy set* $\Pi \subset \mathbb{R}^d_{\ge 0}$ with respect to a *policy modification set* $\Phi \subset \mathbb{R}^{d\times d}$. Here Π is a convex compact subset of $\mathbb{R}^d$, and Φ is a convex compact subset of $\mathbb{R}^{d\times d}$, where each $\phi \in \Phi$ is a *policy modification function* which is a linear transformation from $\mathbb{R}^d$ to $\mathbb{R}^d$ that maps Π to itself ($\phi(\Pi) \subseteq \Pi$). For any algorithm that plays policies $\{\mu^t\}_{t=1}^T$ within T rounds and receives loss functions $\{\ell^t\}_{t=1}^T \subset \mathbb{R}^d_{\ge 0}$, the Φ-regret is defined as

$$\mathrm{Reg}^{\Phi}(T) := \sup_{\phi\in\Phi} \sum_{t=1}^{T} \langle \mu^t - \phi\mu^t, \ell^t\rangle . \tag{1}$$

The Φ-regret subsumes the vanilla regret (i.e. external regret) as a special case by taking Φ to be the set of all constant modifications $\Phi^{\mathsf{ext}} := \{\phi_{\mu_\star} : \mu_\star \in \Pi\}$ where $\phi_{\mu^\star}\mu = \mu^\star$ for all $\mu \in \Pi$. Another widely studied example is the *swap regret* [8] (and the closely related *internal regret* [26]) for normal-form games, where $\Pi = \Delta_d$ is the probability simplex over d actions, and Φ is the set of all stochastic matrices (i.e. those mapping Δ_d to itself). A primary motivation for minimizing the Φ-regret is for computing various types of *Correlated Equilibria* (CEs) in multi-player games using the online-to-batch conversion (see e.g. [14]), which has been established in many games and has been a cornerstone in the online learning and games literature.

Φ-Hedge algorithm A widely used strategy for minimizing the Φ-regret is to use any (black-box) linear regret minimization algorithm on the Φ set to produce a sequence of $\{\phi^t\}_{t=1}^T \subset \Phi$, combined with the *fixed point technique* (e.g. [43])—Output policy μ^t that satisfies the fixed-point equation $\phi^t\mu^t = \mu^t$ in each round t. In the common scenario where Φ is the convex hull of a finite number of *vertices*, i.e. $\Phi = \mathrm{conv}(\Phi_0)$ where Φ_0 is a finite subset of Φ, a standard regret minimization algorithm over Φ is Hedge (a.k.a. Exponential Weights) [4], leading to the Φ-*Hedge* algorithm (Algorithm 1).

It is a standard result ([44], see also Lemma A.1) that Algorithm 1 achieves Φ-regret bound

$$\mathrm{Reg}^{\Phi}(T) \le \frac{\log|\Phi_0|}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{\phi\in\Phi_0} p^t_\phi(\langle \phi\mu^t, \ell^t\rangle)^2. \tag{2}$$

By choosing $\eta > 0$, this result implies a quite desirable bound $\mathrm{Reg}^{\Phi}(T) \le L\sqrt{2\log|\Phi_0|\cdot T}$ in the full-feedback setting (assuming bounded loss $\langle \phi\mu^t, \ell^t\rangle \le L$), and can also be used to prove regret bounds in the bandit-feedback setting.

Algorithm 1 Φ-Hedge

Require: Finite vertex set $\Phi_0 \subset \mathbb{R}^{d\times d}$ such that $\mathrm{conv}(\Phi_0) = \Phi$; Learning rate η.
Initialize $p^1 \in \Delta_{\Phi_0}$ with $p^1_\phi = 1/|\Phi_0|$ for $\phi \in \Phi_0$.
for iteration $t = 1, \dots, T$ **do**
  Compute $\phi^t = \sum_{\phi\in\Phi_0} p^t_\phi \phi$.
  Set policy μ^t to be the fixed point of equation $\mu^t = \phi^t \mu^t$.
  Receive loss function $\ell^t \in \mathbb{R}^d_{\geq 0}$, suffer loss $\langle \mu^t, \ell^t \rangle$.
  Update $p^{t+1}_\phi \propto_\phi p^t_\phi \cdot \exp\{-\eta \langle \phi\mu^t, \ell^t \rangle\}$.

2.2 Extensive-form games (EFGs) and extensive-form trigger regret

In this paper, we consider m-player imperfect-information extensive-form games (EFGs) with perfect-recall (see Appendix B.1 for detailed definitions). For the purpose of this work, we consider an alternative formulation of EFGs—Tree-Form Adversarial Markov Decision Processes (TFAMDP). This model is equivalent to studying EFGs from the perspective of a single player, while treating all other players as adversaries who can change both transitions and rewards in each round.

Tree-form adversarial MDP We consider an episodic, tabular TFAMDP which consists of the followings $(H, \{\mathcal{X}_h\}_{h\in[H]}, \mathcal{A}, \mathcal{T}, \{p^t_h\}_{h\in\{0\}\cup[H], t\geq 1}, \{R^t_h\}_{h\in[H], t\geq 1})$. Here $H \in \mathbb{N}_+$ is the horizon length; $\mathcal{X}_h$ is the space of information sets (henceforth *infosets*) at step h with size $|\mathcal{X}_h| = X_h$ and $\sum_{h=1}^H X_h = X$; $\mathcal{A}$ is the action space with size $|\mathcal{A}| = A$. Next, $\mathcal{T} = \{\mathcal{C}(x,a)\}_{(x,a)\in\mathcal{X}\times\mathcal{A}}$ defines the tree structure over the infosets and actions, where $\mathcal{C}(x_h, a_h) \subset \mathcal{X}_{h+1}$ denotes the set of immediate children of (x_h, a_h). Furthermore, $\{\mathcal{C}(x_h, a_h)\}_{(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}}$ forms a partition of $\mathcal{X}_{h+1}$. It directly follows from the tree structure of TFAMDP that the player has *perfect recall*, i.e., for any infoset $x_h \in \mathcal{X}_h$, there is a unique history $(x_1, a_1, \dots, x_{h-1}, a_{h-1})$ that leads to x_h. Furthermore, $p^t_0(\cdot) \in \Delta_{\mathcal{X}_1}$ is the initial distribution over $\mathcal{X}_1$ at episode t; $p^t_h(\cdot|x_h, a_h)$ is the transition probability from (x_h, a_h) to its immediate children $\mathcal{C}(x_h, a_h)$ at episode t; $R^t_h(\cdot|x_h, a_h)$ is the distribution of the stochastic reward $r \in [0,1]$ received at (x_h, a_h) at episode t, with expectation $\overline{R}^t_h(x_h, a_h)$.

At the beginning of episode t, an adversary will first choose the initial distribution p^t_0, transition $\{p^t_h\}_{h\in[H]}$, and reward distribution $\{R^t_h\}_{h\in[H]}$. Then in the *bandit feedback* setting, at each step h, the player observes the current infoset x_h, takes an action a_h, receives a bandit feedback of the reward $r^t_h \sim R^t_h(\cdot|x_h, a_h)$, and the environment transitions to the next state $x_{h+1} \sim p^t_h(\cdot|x_h, a_h)$.

Policies We use $\mu = \{\mu_h(\cdot|x_h)\}_{h\in[H], x_h\in\mathcal{X}_h}$ to denote a policy, where each $\mu_h(\cdot|x_h) \in \Delta_{\mathcal{A}}$ is the action distribution at infoset x_h. We say μ is a *deterministic* policy if $\mu_h(\cdot|x_h)$ takes some single action with probability 1 for any (h, x_h). Let Π denote the set of all possible policies. We denote the *sequence form* representation of policy $\mu \in \Pi$ by

$$\mu_{1:h}(x_h, a_h) := \prod_{h'=1}^h \mu_{h'}(a_{h'}|x_{h'}), \tag{3}$$

where $(x_1, a_1, \dots, x_{h-1}, a_{h-1})$ is the unique history of x_h. We also identify μ as a vector in $\mathbb{R}^{XA}_{\geq 0}$, whose (x_h, a_h)-th entry is equal to its sequence form $\mu_{1:h}(x_h, a_h)$. Let $\|\Pi\|_1 := \max_{\mu\in\Pi} \|\mu\|_1$, which admits bound $\|\Pi\|_1 \leq X$ but can in addition be smaller (cf. Appendix B.3).

Expected loss function Given any policy μ^t at round t, the total expected loss received at round t (which equals to H minus the total rewards within round t) is given by

$$\langle \mu^t, \ell^t \rangle := \sum_{h,x_h,a_h} \mu^t_{1:h}(x_h, a_h) \ell^t_h(x_h, a_h),$$

where the loss function for the t-th round is given by $\ell^t = \{\ell^t_h(x_h, a_h)\}_{h,x_h,a_h} \in \mathbb{R}^{XA}_{\geq 0}$:

$$\ell^t_h(x_h, a_h) := p^t_0(x_1) \prod_{h'=1}^{h-1} p^t_{h'}(x_{h'+1}|x_{h'}, a_{h'})[1 - \overline{R}^t_h(x_h, a_h)], \tag{4}$$

where $(x_1, a_1, \dots, x_{h-1}, a_{h-1})$ is the unique history that leads to x_h. In the *full feedback* setting, the learner is further capable of observing the full loss vector $\ell^t \in \mathbb{R}^{XA}_{\geq 0}$ at the end of each round t.

Subtree and subtree policies For any $g \le h$, $x_g \in \mathcal{X}_g, x_h \in \mathcal{X}_h$, and any action $a_g, a_h \in \mathcal{A}$, we say x_h or (x_h, a_h) is in the subtree rooted at x_g, written as $x_h \succeq x_g$ or $(x_h, a_h) \succeq x_g$, if x_g is either equal to x_h or is a part of the unique preceding history $(x_1, a_1, \dots, x_{h-1}, a_{h-1})$ which leads to x_h. Similarly, we say x_h or (x_h, a_h) is in the subtree of (x_g, a_g), written as $x_h \succ (x_g, a_g)$ or $(x_h, a_h) \succeq (x_g, a_g)$, if (x_g, a_g) is either equal to (x_h, a_h) (only in the latter case), or is a part of the unique preceding history $(x_1, a_1, \dots, x_{h-1}, a_{h-1})$ which leads to x_h.

For any $g \in [H]$, and any infoset $x_g \in \mathcal{X}_g$, we use $\mu^{x_g} = \{\mu_h^{x_g}(\cdot|x_h) \in \Delta_{\mathcal{A}} : x_h \succeq x_g\}$ to denote a subtree policy rooted at x_g. We use Π^{x_g} and $\mathcal{V}^{x_g}$ to denote the set of all subtree policies and the set of all *deterministic* subtree policies rooted at x_g. We denote the sequence form representation of $\mu^{x_g} \in \Pi^{x_g}$ by:

$$\mu_{g:h}^{x_g}(x_h, a_h) = \begin{cases} \prod_{h'=g}^{h} \mu_{h'}^{x_g}(a_{h'}|x_{h'}) & \text{if } (x_h, a_h) \succeq x_g, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, we can also identify any subtree policy $\mu^{x_g} \in \Pi^{x_g}$ as a vector in $\mathbb{R}_{\ge 0}^{XA}$, whose (x_h, a_h)-th entry is equal to its sequence form $\mu_{g:h}^{x_g}(x_h, a_h)$ (which is non-zero only on the subtree rooted at x_g).

Extensive-form trigger regret The notion of trigger regret is introduced in [27, 13, 23]. An *(extensive-form) trigger modification* $\phi_{x_g a_g \to m^{x_g}}$ is a policy modification that modifies any policy $\mu \in \Pi$ as follows: When x_g is visited and a_g is about to be taken (by μ), we say $x_g a_g$ is *triggered*, in which case the subtree policy rooted at x_g is then replaced by $m^{x_g} \in \Pi^{x_g}$. One can verify that the trigger modification $\phi_{x_g a_g \to m^{x_g}}$ can be written as a linear transformation that maps from Π to Π:

$$\phi_{x_g a_g \to m^{x_g}} := (I - E_{\succeq x_g a_g}) + m^{x_g} e_{x_g a_g}^{\top} \in \mathbb{R}^{XA \times XA}.$$

Here, $E_{\succeq x_g a_g}$ is a diagonal matrix with diagonal entry 1 at all (x_h, a_h) satisfying $(x_h, a_h) \succeq (x_g, a_g)$, and zero otherwise, and $e_{x_g a_g} \in \mathbb{R}^{XA}$ is an indicator vector whose only non-zero entry is 1 at (x_g, a_g). We say $\phi_{x_g a_g \to v^{x_g}}$ is a deterministic trigger modification if $v^{x_g} \in \mathcal{V}^{x_g}$ is a deterministic subtree policy. We denote the set of all deterministic trigger modifications and its convex hull as Φ_0^{Tr} and Φ^{Tr} respectively, where

$$\Phi_0^{\mathsf{Tr}} := \bigcup_{g, x_g, a_g} \bigcup_{v^{x_g} \in \mathcal{V}^{x_g}} \{\phi_{x_g a_g \to v^{x_g}}\}, \qquad \Phi^{\mathsf{Tr}} = \mathrm{conv}\{\Phi_0^{\mathsf{Tr}}\}. \tag{5}$$

The *(extensive-form) trigger regret* is then defined as the difference in the total loss when comparing against the best extensive-form trigger modification in hindsight. We note that the trigger regret is a special case of Φ-regret (1) with $\Phi = \Phi^{\mathsf{Tr}}$.

Definition 1 (Extensive-Form Trigger Regret). *For any algorithm that plays policies $\mu^t \in \Pi$ at round $t \in [T]$, the extensive-form trigger regret (also the EFCE-regret) is defined as*

$$\mathrm{Reg}^{\mathsf{Tr}}(T) := \max_{\phi \in \Phi^{\mathsf{Tr}}} \sum_{t=1}^{T} \langle \mu^t - \phi \mu^t, \ell^t \rangle . \tag{6}$$

From trigger regret to Extensive-Form Correlated Equilibrium (EFCE) The importance of extensive-form trigger regret is in its connection to computing EFCE: By standard online-to-batch conversion [13, 23], if all players have low trigger regret (with $\mathrm{Reg}_i^{\mathsf{Tr}}(T)$ for the i^{th} player), then the average joint policy $\overline{\pi}$ is an ε-EFCE, where $\varepsilon = \max_{i \in [m]} \mathrm{Reg}_i^{\mathsf{Tr}}(T)/T$ (cf. Appendix B.2). We remark in passing by taking $\Phi = \Phi^{\mathsf{ext}}$, low Φ-regret implies learning (Normal-Form) Coarse Correlated Equilibria in EFGs, as well as Nash Equilibria in the two-player zero-sum setting [6].

3 Efficient Φ-Hedge for Extensive-Form Trigger Regret Minimization

In this section, we study the Φ-Hedge algorithm (Algorithm 1) for minimizing the trigger regret. Naively, Algorithm 1 requires maintaining and updating $p^t \in \Delta_{\Phi_0}$ (cf. Line 6), whose computational cost is linear in $|\Phi_0^{\mathsf{Tr}}|$ which can be exponential in X in the worst case[1]. We begin by deriving an efficient implementation of the iterate $\phi^t \in \Phi$ (of Line 3) directly by exploiting the structure of Φ_0^{Tr}.

[1] $|\Phi_0^{\mathsf{Tr}}|$ is at least the number of deterministic policies of the game, which could be $A^{O(X)}$ in the worst case.

3.1 Efficient implementation of Φ^{Tr}-Hedge algorithm

We first use a standard trick to convert the computation of ϕ^t (Line 3 & 6, Algorithm 1) in Φ-Hedge to evaluating the gradient of a suitable log-partition function. This is stated in the lemma below (for any generic Φ_0), whose proof can be found in Appendix C.2.

Lemma 2 (Conversion to log-partition function)**.** *Define the log-partition function $F^{\Phi_0}:\mathbb{R}^{d\times d}\to\mathbb{R}$*

$$F^{\Phi_0}(M) := \log \textstyle\sum_{\phi\in\Phi_0} \exp\{-\langle \phi, M\rangle\}. \tag{7}$$

Then Line 3 of Φ-Hedge (Algorithm 1) has a closed-form update for all $t\ge 1$:

$$\phi^t = -\nabla F^{\Phi_0}\Big(\eta \textstyle\sum_{s=1}^{t-1} M^s\Big) = -\frac{\sum_{\phi\in\Phi_0}\exp\left\{-\eta\left\langle \phi, \sum_{s=1}^{t-1} M^s\right\rangle\right\}\phi}{\sum_{\phi\in\Phi_0}\exp\left\{-\eta\left\langle \phi, \sum_{s=1}^{t-1} M^s\right\rangle\right\}}, \quad M^t := \ell^t(\mu^t)^\top. \tag{8}$$

Eq. (8) suggests a strategy for evaluating $\phi^t = -\nabla F^{\Phi_0}(\eta\sum_{s=1}^{t-1} M^s)$—So long as the vertex set Φ_0 has some structure that allows efficient evaluation of the sum of exponentials on the numerators and denominators (i.e. faster than naive sum), ϕ^t may be computed directly in sublinear in $|\Phi_0|$ time, and there is no need to maintain the underlying distribution $p^t\in\Delta_{\Phi_0}$.

The following lemma enables such an efficient computation for the log-partition function $F^{\mathsf{Tr}} := F^{\Phi^{\mathsf{Tr}}}$ (and its gradient) associated with the trigger modification set $\Phi = \Phi^{\mathsf{Tr}}$. This lemma (proof deferred to Appendix C.3) is a consequence of the specific structure of Φ_0 (cf. (5)), whose elements are indexed by a sequence $x_g a_g$ and a deterministic subtree policy $v^{x_g}\in\mathcal{V}^{x_g}$.

Lemma 3 (Recursive expression of F^{Tr} and ∇F^{Tr})**.** *For any loss matrix $M\in\mathbb{R}^{XA\times XA}$, the* EFCE log-partition function *can be written as*

$$F^{\mathsf{Tr}}(M) = \log \textstyle\sum_{g,x_g,a_g} \exp\Big\{ -\big\langle I - E_{\succeq x_g a_g}, M\big\rangle + F_{x_g a_g, x_g}(M)\Big\}, \tag{9}$$

where for any $x_h\succeq x_g$,

$$F_{x_g a_g, x_h}(M) := \log \textstyle\sum_{a_h} \exp\Big\{ - M_{x_h a_h, x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} F_{x_g a_g, x_{h+1}}(M)\Big\}. \tag{10}$$

Furthermore, define $\lambda = (\lambda_{x_g a_g})_{x_g a_g\in\mathcal{X}\times\mathcal{A}}\in\Delta_{XA}$ and $m = (m_{x_g a_g})_{x_g a_g\in\mathcal{X}\times\mathcal{A}}$ with $m_{x_g a_g}\in\Pi^{x_g}$ (and also identified as a vector in $\mathbb{R}^{XA}$) as

$$\lambda_{x_g a_g} \propto_{x_g a_g} \exp\Big\{ -\big\langle I - E_{\succeq x_g a_g}, M\big\rangle + F_{x_g a_g, x_g}(M)\Big\}, \tag{11}$$

$$m_{x_g a_g, h}(a_h|x_h) \propto_{a_h} \exp\Big\{ - M_{x_h a_h, x_g a_g} + \textstyle\sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} F_{x_g a_g, x_{h+1}}(M)\Big\}, \tag{12}$$

then we have

$$-\nabla F^{\mathsf{Tr}}(M) = \phi(\lambda, m) := \textstyle\sum_{g,x_g,a_g} \lambda_{x_g a_g}(I - E_{\succeq x_g a_g} + m_{x_g a_g} e_{x_g a_g}^\top). \tag{13}$$

Above, $\lambda = (\lambda_{x_g a_g})_{x_g a_g\in\mathcal{X}\times\mathcal{A}}\in\Delta_{XA}$ is a probability distribution over $\mathcal{X}\times\mathcal{A}$, and $m = (m_{x_g a_g})_{x_g a_g\in\mathcal{X}\times\mathcal{A}}\in\mathcal{M}\equiv\prod_{g,x_g a_g}\Pi^{x_g a_g}$ is a collection of subtree policies $m_{x_g a_g}$, where each $m_{x_g a_g}\in\Pi^{x_g}$ is a subtree policy that specifies an action distribution $m_{x_g a_g, h}(a_h|x_h)$ for every $x_h\succeq x_g$, and can be identified with a vector in $\mathbb{R}^{XA}$ (c.f. Section 2.2).

The recursive structure in Lemma 3 offers a roadmap for evaluating (λ, m) and thus $\nabla F^{\mathsf{Tr}}(M)$ in $O(X^2A^2)$ time (formal statement in Appendix C.4). Applying Lemma 3 with $M = \eta\sum_{s=1}^{t-1} M^s$ gives an efficient implementation of (8), i.e. the Φ-Hedge algorithm with $\Phi = \Phi^{\mathsf{Tr}}$. For clarity, we summarize this in Algorithm 2. We remark that the parameters (λ^t, m^t) therein can also be expressed in terms of (λ^{t-1}, m^{t-1}) and M^{t-1}, which we present in Algorithm 4 (the equivalent "OMD" form) in Appendix C.1. We also note that the fixed point equation $\phi^t\mu = \mu$ in Line 5 can be solved in $O(X^2A^2)$ time [23, Corollary 4.15].

3.2 Equivalence to FTRL and OMD

We now show that Algorithm 2 is equivalent to FTRL and OMD with suitable dilated entropies and divergences (hence the name EFCE-OMD). We define the trigger dilated entropy function and trigger

Algorithm 2 EFCE-OMD (FTRL form; equivalent OMD form in Algorithm 4)

Require: Learning rate $\eta > 0$.

for $t = 1, 2, \dots, T$ **do**

For each $x_g a_g \in \mathcal{X} \times \mathcal{A}$, from the reverse order of x_h, compute $m^t_{x_g a_g, h}(a_h | x_h)$ and $F^t_{x_g a_g, x_h}$

$$m^t_{x_g a_g, h}(a_h | x_h) \propto_{a_h} \exp\Big\{ -\eta \textstyle\sum_{s=1}^{t-1} M^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F^t_{x_g a_g, x_{h+1}} \Big\}, \quad (14)$$

$$F^t_{x_g a_g, x_h} = \log \textstyle\sum_{a_h} \exp\Big\{ -\eta \sum_{s=1}^{t-1} M^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F^t_{x_g a_g, x_{h+1}} \Big\}, \quad (15)$$

Compute $\lambda^t_{x_g a_g}$ as

$$\lambda^t_{x_g a_g} \propto_{x_g a_g} \exp\Big\{ -\eta \Big\langle I - E_{\succeq x_g a_g}, \textstyle\sum_{s=1}^{t-1} M^s \Big\rangle + F^t_{x_g a_g, x_g} \Big\}. \quad (16)$$

Compute $\phi^t = \phi(\lambda^t, m^t)$ where ϕ is in Eq. (13).

Compute the policy μ^t, which is a solution of the fixed point equation $\phi^t \mu^t = \mu^t$.

Receive loss $\ell^t = \{\ell^t_h(x_h, a_h)\}_{(x_h, a_h) \in \mathcal{X} \times \mathcal{A}} \in \mathbb{R}^{XA}_{\geq 0}$.

Compute matrix loss $M^t = \ell^t (\mu^t)^\top \in \mathbb{R}^{XA \times XA}_{\geq 0}$.

dilated KL divergence function over $(\lambda, m) \in \Delta_{XA} \times \mathcal{M}$ as

$$H^{\mathsf{Tr}}(\lambda, m) := H(\lambda) + \textstyle\sum_{g, x_g, a_g} \lambda_{x_g a_g} H_{x_g}(m_{x_g a_g}),$$

$$D^{\mathsf{Tr}}(\lambda, m \| \lambda', m') := D_{\mathrm{KL}}(\lambda \| \lambda') + \textstyle\sum_{g, x_g, a_g} \lambda_{x_g a_g} D_{x_g}(m_{x_g a_g} \| m'_{x_g a_g}),$$

where $H(\cdot)$ and $D_{\mathrm{KL}}(\cdot \| \cdot)$ are the (negative) Shannon entropy and KL divergence; and for any x_g, $H_{x_g}(\cdot)$ is the dilated entropy, and $D_{x_g}(\cdot \| \cdot)$ is the dilated KL divergence [30], both for the subtree rooted at x_g (detailed definitions in Appendix C.5).

Lemma 4 (Equivalent formulations of Φ^{Tr}-hedge)**.** *For any sequence of loss functions $\{M^t\}_{t \geq 1}$, the iterates (λ^t, m^t) in Algorithm 2 (i.e. (14)-(16)) are equivalent to (i.e. satisfy) the following FTRL update on H^{Tr} and OMD update on D^{Tr}:*

$$(\lambda^t, m^t) = \arg\min_{\lambda, m} \Big[\eta \Big\langle \phi(\lambda, m), \textstyle\sum_{s=1}^{t-1} M^s \Big\rangle + H^{\mathsf{Tr}}(\lambda, m) \Big], \quad (17)$$

$$(\lambda^t, m^t) = \arg\min_{\lambda, m} \Big[\eta \big\langle \phi(\lambda, m), M^{t-1} \big\rangle + D^{\mathsf{Tr}}(\lambda, m \| \lambda^{t-1}, m^{t-1}) \Big]. \quad (18)$$

The proof of Lemma 4 follows directly by the concrete forms of (λ^t, m^t) in (14)-(16), and can be found in Appendix C.6.

3.3 Regret bound under full feedback and bandit feedback

We now present the regret bounds of Algorithm 2. We emphasize that these regret bounds are simple consequence of the generic bound for Φ-Hedge in (2), and their proofs do not depend on the actual implementation of Algorithm 2 developed in the preceding two subsections. We first consider the full feedback setting, where the full expected loss vector $\ell^t \in \mathbb{R}^{XA}_{\geq 0}$ is received after each episode.

Theorem 5 (Regret bound of EFCE-OMD under full feedback)**.** *Running Algorithm 2 with $\eta = \mathcal{O}(\sqrt{\|\Pi\|_1 \iota / (H^2 T)})$ achieves the following trigger regret bound*

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \leq \mathcal{O}\big(\sqrt{H^2 \|\Pi\|_1 \iota T}\big),$$

where $\iota := \log(XA)$ is a log factor.

The proof of Theorem 5 is simply by applying (2) and observing that $\log(\Phi^{\mathsf{Tr}}_0) \leq \|\Pi\|_1 \log A + \log(XA)$ (see Appendix D.1). This theorem shows that the Φ^{Tr}-Hedge algorithm gives $\widetilde{\mathcal{O}}(\sqrt{XT})$ trigger regret bound, which matches the information-theoretic lower bound $\Omega(\sqrt{XT})$ [48, Theorem 2] up to a $\widetilde{\mathcal{O}}(\mathrm{poly}(H))$ factor, and is slightly better than the $\widetilde{\mathcal{O}}(\sqrt{XAT})$ upper bound of [42, Corollary F.3] though their definition of EFCE-regret is slightly stricter (thus higher) than ours.

Algorithm 3 Balanced EFCE-OMD (FTRL form; equivalent OMD form in Algorithm 5)

Require: Learning rate η, balanced exploration policy $\{\mu^{\star,h}\}_{h\in[H]}$.

for $t = 1, 2, \dots, T$ **do**

For each $x_g a_g \in \mathcal{X}\times\mathcal{A}$, from the reverse order of x_h, compute $m^t_{x_g a_g, h}(a_h|x_h)$ and $F^{\star,t}_{x_g a_g, x_h}$

$$m^t_{x_g a_g,h}(a_h|x_h) \propto_{a_h} \exp\Big\{\mu^{\star,h}_{g:h}(x_h,a_h)\Big(-\eta\sum_{s=1}^{t-1}\widetilde{M}^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} F^{\star,t}_{x_g a_g, x_{h+1}}\Big)\Big\},$$

$$F^{\star,t}_{x_g a_g, x_h} := \frac{1}{\mu^{\star,h}_{g:h}(x_h,a_h)}\log\sum_{a_h\in\mathcal{A}}\exp\Big\{\mu^{\star,h}_{g:h}(x_h,a_h) \times\Big[-\eta\sum_{s=1}^{t}\widetilde{M}^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h a_h)} F^{\star,t}_{x_g a_g, x_{h+1}}\Big]\Big\}.$$

Compute $\lambda^{t+1}_{x_g a_g}$ as

$$\lambda^t_{x_g a_g} \propto_{x_g a_g} \exp\Big\{\frac{1}{XA}\Big(-\eta\langle I - E_{\succeq x_g a_g}, \sum_{s=1}^{t-1}\widetilde{M}^s\rangle + F^{\star,t}_{x_g a_g, x_g}\Big)\Big\}. \tag{20}$$

Compute $\phi^t = \phi(\lambda^t, m^t)$, where ϕ is as defined in Eq. (13).

Find a μ^t to be a solution of the fixed point equation $\mu^t = \phi^t\mu^t$.

Play policy μ^t, observe trajectory $(x^t_h, a^t_h, r^t_h)_{h\in[H]}$.

Form vector loss estimator $\widetilde{\ell}^{t,x_g a_g} = \{\widetilde{\ell}^{t,x_g a_g}_h(x_h,a_h)\}_{x_h a_h}$ for each $(g, x_g a_g)$ as in Eq. (23).

Compute matrix loss estimator $\widetilde{M}^t = \sum_{g,x_g,a_g}\mu^t_{x_g a_g}\widetilde{\ell}^{t,x_g a_g} e^\top_{x_g a_g}$.

In the bandit feedback setting, the learner only observes her own rewards and infosets. In this case we replace ℓ^t in Algorithm 2 with the following loss estimator (with IX bonus γ) proposed in [34]:

$$\widetilde{\ell}^t_h(x_h,a_h) := \mathbf{1}\{(x^t_h,a^t_h)=(x_h,a_h)\}(1-r^t_h)/(\mu^t_{1:h}(x_h,a_h)+\gamma). \tag{19}$$

We show that EFCE-OMD achieves the following guarantee in the bandit feedback setting (proof in Appendix D.2). The proof follows by plugging the loss estimator $\widetilde{\ell}^t$ into (2) and additionally bounding concentrations (which we remark is a better strategy than using a naive bandit-based loss estimator in the corresponding NFG space).

Theorem 6 (Regret bound of EFCE-OMD under bandit feedback)**.** *Run Algorithm 2 with loss estimator $\{\widetilde{\ell}^t\}_{t=1}^T$ (19), $\eta = \sqrt{\|\Pi\|_1\log A/(HXAT)}$, and $\gamma = \sqrt{\|\Pi\|_1\,\iota/(XAT)}$. Then we have the following trigger regret bound with probability at least $1-\delta$:*

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \le \mathcal{O}\big(\sqrt{HXA\|\Pi\|_1\iota\cdot T}\big),$$

where $\iota = \log(3XA/\delta)$ is a log term.

To our best knowledge, Theorem 6 gives the first trigger regret bound against adversarial opponents and bandit feedback. This $\widetilde{\mathcal{O}}(\sqrt{XA\|\Pi\|_1 T})$ rate is $\sqrt{XA}$ worse than Theorem 5 (ignoring H and log factors), and is at most $\widetilde{\mathcal{O}}(\sqrt{X^2AT})$ using $\|\Pi\|_1 \le X$.

4 Balanced EFCE-OMD for bandit feedback

We now build upon the EFCE-OMD algorithm (Algorithm 2) to develop a new algorithm, *Balanced EFCE-OMD* (Algorithm 3), and show that it achieves near-optimal extensive-form trigger regret guarantee under bandit feedback. Here we discuss the two key modifications in the algorithm design.

Key modification I: "Rebalancing" the log-partition function Building on the balancing technique of [6], we start from Eq. (9) and (10) of the log partition function, and rescale the inner functions $F_{x_g a_g, x_h}$ using *balanced exploration policies* $\{\mu^{\star,h}_{g:h}(x_h,a_h)\}_{g,x_h,a_h}$ (see Definition E.1

for the formal definition), and rescale the outer function F^{Tr} by XA. Concretely, for any matrix $M \in \mathbb{R}^{XA \times XA}$, we define the *balanced EFCE log-partition function* as

$$F^{\mathsf{Tr}}_{\mathsf{bal}}(M) := XA \log \sum\nolimits_{g, x_g, a_g} \exp \left\{ \tfrac{1}{XA} \left[- \left\langle I - E_{\succeq x_g a_g}, M \right\rangle + F^{\star}_{x_g a_g, x_g}(M) \right] \right\}, \tag{21}$$

where for any $x_h \succeq x_g$ (using $\mu^{\star,h}_{g:h} := \mu^{\star,h}_{g:h}(x_h, a_h)$ as shorthand, which depends on x_h but not a_h),

$$F^{\star}_{x_g a_g, x_h}(M) := \frac{1}{\mu^{\star,h}_{g:h}} \log \sum_{a_h} \exp \left\{ \mu^{\star,h}_{g:h} \left[- M_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} F^{\star}_{x_g a_g, x_{h+1}}(M) \right] \right\}. \tag{22}$$

Key modification II: New loss estimator under bandit feedback We use an *adaptive* family of bandit-based loss estimators $\{\widetilde{\ell}^{t, x_g a_g}\}_{x_g a_g} \subset \mathbb{R}^{XA}_{\geq 0}$, one for each $(x_g, a_g) \in \mathcal{X} \times \mathcal{A}$, defined as

$$\widetilde{\ell}^{t, x_g a_g}_h(x_h, a_h) := \frac{\mathbf{1}\left\{(x^t_h, a^t_h) = (x_h, a_h)\right\}(1 - r^t_h)}{\mu^t_{1:h}(x_h, a_h) + \gamma(\mu^{\star,h}_{1:h}(x_h, a_h) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h)\mathbf{1}\{x_h \succeq x_g\})}, \tag{23}$$

where $\mu^t_{x_g a_g} := \mu^t_{1:g}(x_g, a_g)$ for shorthand. The main difference of (23) over (19) is in the adaptive IX bonus term on the denominator that scales with γ but is different for each $x_g a_g$. We then place each $\mu^t_{x_g a_g} \widetilde{\ell}^{t, x_g a_g}$ into the $x_g a_g$-th column of a matrix loss estimator $\widetilde{M}^t$, or in matrix form,

$$\widetilde{M}^t := \sum\nolimits_{g, x_g, a_g} \mu^t_{x_g a_g} \widetilde{\ell}^{t, x_g a_g} e^{\top}_{x_g a_g}.$$

With (21)-(23) at hand, our algorithm Balanced EFCE-OMD is defined as the negative gradient of $F^{\mathsf{Tr}}_{\mathsf{bal}}$ evaluated at the cumulative loss estimators:

$$\phi^t = -\nabla F^{\mathsf{Tr}}_{\mathsf{bal}} \left(\eta \sum\nolimits_{s=1}^{t-1} \widetilde{M}^s \right), \quad \forall t \geq 1, \tag{24}$$

and $\mu^t \in \Pi$ solves the fixed point equation $\phi^t \mu^t = \mu^t$. Similar as EFCE-OMD, (24) also admits efficient implementations in both FTRL and OMD form (cf. Algorithm 3 & 5). The corresponding (λ^t, m^t) is also equivalent to running a FTRL/OMD algorithm with respect to a *balanced* dilated entropy/KL-divergence over $\phi \in \Phi^{\mathsf{Tr}}$ (cf. Lemma E.4 and Appendix E.3 for details).

Main result We now present the theoretical guarantee of Algorithm 3 (proof in Appendix F).

Theorem 7. *Balanced EFCE-OMD (Algorithm 3) with $\eta = \sqrt{XA\iota/H^4T}$ and $\gamma = 2\sqrt{XA\iota/H^2T}$ achieves the following extensive-form trigger regret bound with probability at least $1 - \delta$:*

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \leq \mathcal{O}\left(\sqrt{H^4 XAT\iota}\right),$$

where $\iota = \log(10XA/\delta)$ is a log term.

The $\widetilde{\mathcal{O}}(\sqrt{XAT})$ trigger regret asserted in Theorem 7 improves over Theorem 6 by a factor of $\sqrt{\|\Pi\|_1}$, and matches the information-theoretic lower bound up to $\mathrm{poly}(H)$ and log factors[2]. By the online-to-batch conversion (Appendix B.2), Theorem 7 also implies an $\widetilde{\mathcal{O}}(H^4 XA/\varepsilon^2)$ sample complexity for learning EFCE under bandit feedback (assuming same game sizes for all m players). This improves over the best known $\widetilde{\mathcal{O}}(mH^6 XA^2/\varepsilon^2)$ sample complexity in the recent work of Song et al. [42][3].

Overview of techniques The proof of Theorem 7 is significantly more challenging than that of Theorem 6, even though the algorithm itself is designed by appearingly simple modifications. This happens since Algorithm 3, unlike Algorithm 2, no longer necessarily corresponds to any normal-form algorithm. The technical crux of the proof is to bound the nonlinear part of $F^{\mathsf{Tr}}_{\mathsf{bal}}$ (with respect to the losses), which we do by carefully controlling a series of second-order terms utilizing the balanced policies within $F^{\mathsf{Tr}}_{\mathsf{bal}}$ and the new adaptive IX bonus within $\{\widetilde{\ell}^{t, x_g a_g}\}_{x_g a_g}$ (Lemma F.5-F.8).

---

[2] As the trigger regret is lower bounded by the vanilla (external) regret, [6, Theorem 6] implies an $\Omega(\sqrt{XAT})$ lower bound for the trigger regret as well under bandit feedback.

[3] We remark though that the 1-EFR algorithm of [42] actually finds an "1-EFCE" which is slightly stronger than our EFCE defined via trigger modifications.

5 Equivalence of OMD and Vertex MWU for external regret minimization

As another illustration of our framework, we now choose $\Phi = \Phi^{\mathsf{ext}} = \mathrm{conv}\{\Phi_0^{\mathsf{ext}}\}$ to be the set of *external* policy modifications, which modify any policy to some deterministic policy. In this case, the Φ^{ext}-Hedge algorithm minimizes the external regret in EFGs. In this section, we show that Φ^{ext}-Hedge, same as the vertex MWU algorithm considered in Farina et al. [25], is actually equivalent to the OMD with dilated entropy [30]. Let $\{\ell^t\}_{t\geq 1} \subset \mathbb{R}_{\geq 0}^{XA}$ be an arbitrary sequence of loss vectors.

Vertex MWU We use $\mathcal{V}$ to denote all the deterministic sequence-form policies, which can also be viewed as the vertex set of the policy set Π. A simple reformulation (cf. Appendix G) shows that Φ^{ext}-Hedge (Algorithm 1) gives the vertex MWU algorithm considered by Farina et al. [25]

$$\mu^t = \textstyle\sum_{v\in\mathcal{V}} p_v^t \cdot v \qquad \text{and} \qquad p_v^t \propto_v \exp\left\{-\eta\left\langle v, \sum_{s=1}^{t-1} \ell^s\right\rangle\right\}. \tag{25}$$

OMD with dilated entropy Another popular algorithm for external regret minimization is the OMD algorithm on the sequence-form policy space with the dilated entropy [30, 35]:

$$\mu^t = \arg\min_{\mu\in\Pi} \left[\eta\left\langle \mu, \ell^{t-1}\right\rangle + D_\emptyset(\mu\|\mu^{t-1})\right], \tag{26}$$

$$D_\emptyset(\mu\|\nu) := \sum_{h=1}^{H}\sum_{x_h,a_h} \mu_{1:h}(x_h,a_h)\log\frac{\mu_h(a_h|x_h)}{\nu_h(a_h|x_h)}. \tag{27}$$

Theorem 8 (Equivalence of OMD and Vertex MWU)**.** *For any sequence of loss vectors $\{\ell^t\}_{t\geq 1}$, OMD with dilated entropy is equivalent to Vertex MWU, that is,* (26) *and* (25) *give the same $\{\mu^t\}_{t\geq 1}$.*

The proof of Theorem 8 can be found in Appendix G.1. Our proof also reveals that the efficient implementation of Vertex MWU developed by Farina et al. [25] using the “kernel trick” is actually equivalent to the standard linear-time efficient implementation of OMD with dilated entropy.

6 Conclusion

In this paper, we present an efficient implementation of the Φ-Hedge algorithm for minimizing the extensive form trigger regret. The algorithm is equivalent to OMD with dilated regularizers, and achieves efficient regret bounds under both full feedback and bandit feedback. We also design an improved algorithm Balanced EFCE-OMD, which achieves a sharp trigger regret bound under bandit feedback. We believe our work leads to many open questions, such as efficient implementations of Φ-Hedge with more general Φ sets (e.g. the behavioral modifications considered in [38, 42]), or accelerated $\mathrm{polylog}(T)$ Φ-regret bounds under full feedback by optimistic algorithms.

Acknowledgment

S.M. is supported by NSF grant DMS-2210827. C.J. is supported by Office of Naval Research N00014-22-1-2253. T.Y. is supported by NSF CCF-2112665 (TILOS AI Research Institute).

References

[1] A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pages 1638–1646. PMLR, 2014.

[2] I. Anagnostides, C. Daskalakis, G. Farina, M. Fishelson, N. Golowich, and T. Sandholm. Near-optimal no-regret learning for correlated equilibria in multi-player general-sum games. *arXiv preprint arXiv:2111.06008*, 2021.

[3] I. Anagnostides, G. Farina, C. Kroer, C.-W. Lee, H. Luo, and T. Sandholm. Uncoupled learning dynamics with $o(logt)$ swap regret in multiplayer games. *arXiv preprint arXiv:2204.11417*, 2022.

[4] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of computing*, 8(1):121–164, 2012.

[5] R. J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of mathematical Economics*, 1(1):67–96, 1974.

[6] Y. Bai, C. Jin, S. Mei, and T. Yu. Near-optimal learning of extensive-form games with imperfect information. *arXiv preprint arXiv:2202.01752*, 2022.

[7] A. Bakhtin, D. Wu, A. Lerer, and N. Brown. No-press diplomacy from scratch. *Advances in Neural Information Processing Systems*, 34, 2021.

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

[9] N. Brown and T. Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

[10] N. Brown and T. Sandholm. Superhuman ai for multiplayer poker. *Science*, 365(6456):885–890, 2019.

[11] N. Burch, M. Moravcik, and M. Schmid. Revisiting cfr+ and alternating updates. *Journal of Artificial Intelligence Research*, 64:429–443, 2019.

[12] A. Celli, A. Marchesi, T. Bianchi, and N. Gatti. Learning to correlate in multi-player general-sum sequential games. *Advances in Neural Information Processing Systems*, 32, 2019.

[13] A. Celli, A. Marchesi, G. Farina, and N. Gatti. No-regret learning dynamics for extensive-form correlated equilibrium. *Advances in Neural Information Processing Systems*, 33:7722–7732, 2020.

[14] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[15] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

[16] C. Daskalakis, M. Fishelson, and N. Golowich. Near-optimal no-regret learning in general games. *Advances in Neural Information Processing Systems*, 34, 2021.

[17] M. Dudik and G. Gordon. A sampling-based approach to computing equilibria in succinct extensive-form games. *arXiv preprint arXiv:1205.2649*, 2012.

[18] M. Dudík and G. J. Gordon. A sampling-based approach to computing equilibria in succinct extensive-form games. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 151–160, 2009.

[19] G. Farina and T. Sandholm. Model-free online learning in unknown sequential decision making problems and games. *arXiv preprint arXiv:2103.04539*, 2021.

[20] G. Farina, C. K. Ling, F. Fang, and T. Sandholm. Correlation in extensive-form games: Saddle-point formulation and benchmarks. *Advances in Neural Information Processing Systems*, 32, 2019.

[21] G. Farina, C. Kroer, and T. Sandholm. Faster game solving via predictive blackwell approachability: Connecting regret matching and mirror descent. *arXiv preprint arXiv:2007.14358*, 2020.

[22] G. Farina, C. Kroer, and T. Sandholm. Stochastic regret minimization in extensive-form games. In *International Conference on Machine Learning*, pages 3018–3028. PMLR, 2020.

[23] G. Farina, A. Celli, A. Marchesi, and N. Gatti. Simple uncoupled no-regret learning dynamics for extensive-form correlated equilibrium. *arXiv preprint arXiv:2104.01520*, 2021.

[24] G. Farina, R. Schmucker, and T. Sandholm. Bandit linear optimization for sequential decision making and extensive-form games. *arXiv preprint arXiv:2103.04546*, 2021.

[25] G. Farina, C.-W. Lee, H. Luo, and C. Kroer. Kernelized multiplicative weights for 0/1-polyhedral games: Bridging the gap between learning in extensive-form and normal-form games. *arXiv preprint arXiv:2202.00237*, 2022.

[26] D. P. Foster and R. V. Vohra. Asymptotic calibration. *Biometrika*, 85(2):379–390, 1998.

[27] G. J. Gordon, A. Greenwald, and C. Marks. No-regret learning in convex games. In *Proceedings of the 25th international conference on Machine learning*, pages 360–367, 2008.

[28] A. Greenwald and A. Jafari. A general class of no-regret learning algorithms and game-theoretic equilibria. In *Learning theory and kernel machines*, pages 2–12. Springer, 2003.

[29] P. A. Herbig. Game theory in marketing: Applications, uses and limits. *Journal of Marketing Management*, 7(3):285–298, 1991.

[30] S. Hoda, A. Gilpin, J. Pena, and T. Sandholm. Smoothing techniques for computing nash equilibria of sequential games. *Mathematics of Operations Research*, 35(2):494–512, 2010.

[31] W. Huang and B. von Stengel. Computing an extensive-form correlated equilibrium in polynomial time. In *International Workshop on Internet and Network Economics*, pages 506–513. Springer, 2008.

[32] V. Kakkad, H. Shah, R. Patel, and N. Doshi. A comparative study of applications of game theory in cyber security and cloud computing. *Procedia Computer Science*, 155:680–685, 2019.

[33] S. Khot and A. K. Ponnuswami. Minimizing wide range regret with time selection functions. In *COLT*, pages 81–86, 2008.

[34] T. Kozuno, P. Ménard, R. Munos, and M. Valko. Model-free learning for two-player zero-sum partially observable markov games with perfect recall. *arXiv preprint arXiv:2106.06279*, 2021.

[35] C. Kroer, K. Waugh, F. Kilinç-Karzan, and T. Sandholm. Faster first-order methods for extensive-form game solving. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 817–834, 2015.

[36] M. Lanctot, K. Waugh, M. Zinkevich, and M. H. Bowling. Monte carlo sampling for regret minimization in extensive games. In *NIPS*, pages 1078–1086, 2009.

[37] T. Lattimore and C. Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[38] D. Morrill, R. D'Orazio, M. Lanctot, J. R. Wright, M. Bowling, and A. R. Greenwald. Efficient deviation types and learning for hindsight rationality in extensive-form games. In *International Conference on Machine Learning*, pages 7818–7828. PMLR, 2021.

[39] R. B. Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981.

[40] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.

[41] M. Schmid, M. Moravcik, N. Burch, R. Kadlec, J. Davidson, K. Waugh, N. Bard, F. Timbers, M. Lanctot, Z. Holland, et al. Player of games. *arXiv preprint arXiv:2112.03178*, 2021.

[42] Z. Song, S. Mei, and Y. Bai. Sample-efficient learning of correlated equilibria in extensive-form games. *arXiv preprint arXiv:2205.07223*, 2022.

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

[44] G. Stoltz and G. Lugosi. Learning correlated equilibria in games with compact sets of strategies. *Games and Economic Behavior*, 59(1):187–208, 2007.

[45] Y. Tian, Q. Gong, and T. Jiang. Joint policy search for multi-agent collaboration with imperfect information. *arXiv preprint arXiv:2008.06495*, 2020.

[46] B. Von Stengel and F. Forges. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research*, 33(4):1002–1022, 2008.

[47] B. H. Zhang and T. Sandholm. Finding and certifying (near-) optimal strategies in black-box extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5779–5788, 2021.

[48] Y. Zhou, T. Ren, J. Li, D. Yan, and J. Zhu. Lazy-cfr: fast and near optimal regret minimization for extensive games with imperfect information. *arXiv preprint arXiv:1810.04433*, 2018.

[49] Y. Zhou, J. Li, and J. Zhu. Posterior sampling for multi-agent reinforcement learning: solving extensive games with imperfect information. In *International Conference on Learning Representations*, 2019.

[50] Y. Zhou, T. Ren, J. Li, D. Yan, and J. Zhu. Lazy-cfr: fast and near-optimal regret minimization for extensive games with imperfect information. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rJx4p3NYDB.

[51] M. Zinkevich, M. Johanson, M. Bowling, and C. Piccione. Regret minimization in games with incomplete information. *Advances in neural information processing systems*, 20:1729–1736, 2007.

A Technical tools

The following lemma is standard and gives a Φ-regret bound of the Φ-Hedge algorithm.

Lemma A.1 (Regret bound for Φ-Hedge)**.** *For strategy modification vertex set Φ_0, step size η, and total steps T, running Algorithm 1 gives*

$$\mathrm{Reg}^{\Phi}(T) \le \frac{\log|\Phi_0|}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{\phi\in\Phi_0} p_\phi^t\big(\langle \phi\mu^t, \ell^t\rangle\big)^2.$$

Proof. We have

$$\begin{aligned}\mathrm{Reg}^{\Phi}(T) &= \sup_{\phi\in\Phi}\sum_{t=1}^{T}\langle \mu^t - \phi\mu^t, \ell^t\rangle \overset{(i)}{=} \sup_{\phi\in\Phi}\sum_{t=1}^{T}\langle \phi^t\mu^t - \phi\mu^t, \ell^t\rangle \\ &\overset{(ii)}{=} \sup_{p\in\Delta_{\Phi_0}}\sum_{t=1}^{T}\sum_{\phi\in\Phi_0}\big(p_\phi^t\langle\phi\mu^t,\ell^t\rangle - p_\phi\langle\phi\mu^t,\ell^t\rangle\big).\end{aligned}$$

Above, (i) uses the fixed point equation $\phi^t\mu^t = \mu^t$ (Line 4), and (ii) uses the fact that $\Phi = \mathrm{conv}\{\Phi_0\}$. Note that the above expression is exactly the regret of $\{p^t\}_{t=1}^T$, where the loss vector in the t-th round is $\{\langle\phi\mu^t,\ell^t\rangle\}_{\phi\in\Phi_0}$. Further, the update rule of p^t (Line 6) coincides with Hedge algorithm. So by the standard regret bound for Hedge, see, e.g. (Lattimore and Szepesvári [37], Proposition 28.7), we have

$$\begin{aligned}\mathrm{Reg}^{\Phi}(T) &= \sup_{p\in\Delta_{\Phi_0}}\sum_{t=1}^{T}\sum_{\phi\in\Phi_0}\big(p_\phi^t\langle\phi\mu^t,\ell^t\rangle - p_\phi\langle\phi\mu^t,\ell^t\rangle\big) \\ &\le \frac{\log|\Phi_0|}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{\phi\in\Phi_0} p_\phi\big(\langle\phi\mu^t,\ell^t\rangle\big)^2.\end{aligned}$$

This proves the lemma. □

The following Freedman's inequality can be found in [1, Lemma 9].

Lemma A.2 (Freedman's inequality)**.** *Suppose random variables $\{X_t\}_{t=1}^T$ is a martingale difference sequence, i.e. $X_t \in \mathcal{F}_t$ where $\{\mathcal{F}_t\}_{t\ge1}$ is a filtration, and $\mathbb{E}[X_t|\mathcal{F}_{t-1}] = 0$. Suppose $X_t \le R$ almost surely for some (non-random) $R > 0$. Then for any $\lambda \in (0, 1/R]$, we have with probability at least $1-\delta$ that*

$$\sum_{t=1}^{T} X_t \le \lambda\cdot\sum_{t=1}^{T}\mathbb{E}\big[X_t^2|\mathcal{F}_{t-1}\big] + \frac{\log(1/\delta)}{\lambda}.$$

B Properties of the game

In this section we some properties of EFGs (using the TFAMDP definition in Section 2.2).

B.1 Equivalence to classical definitions of EFGs

We first formally define Extensive-Form Games (EFGs). We then show that solving EFGs with adversarial opponents can be reduced to solving Tree-Formed AMDP.

B.1.1 Classical definition of EFGs

We consider the problem of multi-player general-sum version of EFGs with adversarial opponents. We remark that in order to study this problem, it suffices to study the two-player zero-sum version. To apply the results obtained in the latter setting to the former setting, we simply view the second player as the collection of all other players (who play jointly against the first player), and view the zero-sum rewards as the reward of the first player and its opposite number.

Partially observable Markov games Following the convention of [34, 6], we consider EFGs under the model of finite-horizon, tabular, two-player zero-sum Markov Games with partial observability, which can be described as a tuple $\mathrm{POMG}(H, \mathcal{S}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B}, \mathbb{P}, r)$, where

- H is the horizon length;
- $\mathcal{S} = \bigcup_{h\in[H]} \mathcal{S}_h$ is the (underlying) state space with $|\mathcal{S}_h| = S_h$ and $\sum_{h=1}^{H} S_h = S$;
- $\mathcal{X} = \bigcup_{h\in[H]} \mathcal{X}_h$ is the space of information sets (henceforth *infosets*) for the *max-player* with $|\mathcal{X}_h| = X_h$ and $X := \sum_{h=1}^{H} X_h$. At any state $s_h \in \mathcal{S}_h$, the max-player only observes the infoset $x_h = x(s_h) \in \mathcal{X}_h$, where $x : \mathcal{S} \to \mathcal{X}$ is the emission function for the max-player;
- $\mathcal{Y} = \bigcup_{h\in[H]} \mathcal{Y}_h$ is the space of infosets for the *min-player* with $|\mathcal{Y}_h| = Y_h$ and $Y := \sum_{h=1}^{H} Y_h$. An infoset y_h and the emission function $y : \mathcal{S} \to \mathcal{Y}$ are defined similarly.
- $\mathcal{A}$, $\mathcal{B}$ are the action spaces for the max-player and min-player respectively, with $|\mathcal{A}| = A$ and $|\mathcal{B}| = B$.
- $\mathbb{P} = \{p_1(\cdot) \in \Delta(\mathcal{S}_1)\} \cup \{p_h(\cdot|s_h, a_h, b_h) \in \Delta(\mathcal{S}_{h+1})\}_{(s_h,a_h,b_h)\in\mathcal{S}_h\times\mathcal{A}\times\mathcal{B},\ h\in[H-1]}$ are the transition probabilities, where $p_1(s_1)$ is the probability of the initial state being s_1, and $p_h(s_{h+1}|s_h, a_h, b_h)$ is the probability of transiting to s_{h+1} given state-action (s_h, a_h, b_h) at step h;
- $r = \{r(s_H, a_H, b_H) \in [0, 1]\}_{(s_H,a_H,b_H)\in\mathcal{S}_H\times\mathcal{A}\times\mathcal{B}}$ are the (random) rewards received in the very last step with mean $\bar{r}(s_H, a_H, b_H)$. [4]

Policies, value functions As we consider partially observability, each player's policy can only depend on the infoset rather than the underlying state. A policy for the max-player is denoted by $\mu = \{\mu_h(\cdot|x_h) \in \Delta(\mathcal{A})\}_{h\in[H],x_h\in\mathcal{X}_h}$, where $\mu_h(a_h|x_h)$ is the probability of taking action $a_h \in \mathcal{A}$ at infoset $x_h \in \mathcal{X}_h$. Similarly, a policy for the min-player is denoted by $\nu = \{\nu_h(\cdot|y_h) \in \Delta(\mathcal{B})\}_{h\in[H],y_h\in\mathcal{Y}_h}$. A trajectory for the max player takes the form $(x_1, a_1, x_2, \ldots, x_H, a_H, r)$, where $a_h \sim \mu_h(\cdot|x_h)$, and the reward and infoset transitions depend on the (unseen) opponent's actions and underlying state transition.

The overall game value for any (product) policy (μ, ν) is denoted by $V^{\mu,\nu} := \mathbb{E}_{\mu,\nu}[r(s_H, a_H, b_H)]$. The max-player aims to maximize the value, whereas the min-player aims to minimize the value.

Interaction protocol At the beginning of each episode t, the adversarial opponent chooses a policy ν^t, which is not revealed to the player. Then, the initial state s_1 is sampled from $p_1(\cdot)$, and its corresponding information sets x_1, y_1 are revealed to each player respectively. At each step h, the system is in a underlying state s_h. The player chooses an action a_h according to the observed infoset x_h, while the opponent simultaneously chooses an action b_h according to policy ν^t and the observed infoset y_h. Afterwards, the environment transitions to a new state s_{h+1} according to $p_h(\cdot|s_h, a_h, b_h)$. This episode ends when (a_H, b_H) is played, and reward $r(s_H, a_H, b_H)$ is observed.

Tree structure and perfect recall We assume that our POMG has a *tree structure*: For any h and $s_h \in \mathcal{S}_h$, there exists a unique history $(s_1, a_1, b_1, \ldots, s_{h-1}, a_{h-1}, b_{h-1})$ of past states and actions that leads to s_h. We also assume that both players have *perfect recall*: For any h and any infoset $x_h \in \mathcal{X}_h$ for the max-player, there exists a unique history $(x_1, a_1, \ldots, x_{h-1}, a_{h-1})$ of past infosets and max-player actions that leads to x_h (and similarly for the min-player). We similarly define $\mathcal{C}(x_h, a_h) \subset \mathcal{X}_{h+1}$ to be the set of all immediate children of (x_h, a_h) at step $h + 1$.

With the tree structure and perfect recall, under any product policy (μ, ν), the probability of reaching state-action (s_h, a_h, b_h) at step h takes the form

$$\mathbb{P}^{\mu,\nu}(s_h, a_h, b_h) = p_{1:h}(s_h)\mu_{1:h}(x_h, a_h)\nu_{1:h}(y_h, b_h), \tag{28}$$

where we have defined the sequence-form transition probability as

$$p_{1:h}(s_h) := p_1(s_1) \prod_{h' \le h-1} p_{h'}(s_{h'+1}|s_{h'}, a_{h'}, b_{h'}),$$

[4]We note in this formulation, reward depends on latent state, which can reveal information about latent state beyond information sets. Here, we consider the formulation where reward is only revealed in the very last step, to avoid such information leakage in the earlier steps. The more general formulation where rewards are received at every step can be translated to this formulation (by postponing all rewards to the last step), which will only incur mild (one or two) additional H factors in the rates.

and the *sequence-form policies* as

$$\mu_{1:h}(x_h, a_h) := \prod_{h'=1}^{h} \mu_{h'}(a_{h'}|x_{h'}), \quad \nu_{1:h}(y_h, b_h) := \prod_{h'=1}^{h} \nu_{h'}(b_{h'}|y_{h'}).$$

Above, $\{s_{h'}, a_{h'}, b_{h'}\}_{h' \le h-1}$ are the histories uniquely determined from s_h, and $x_{h'} = x(s_{h'})$, $y_{h'} = y(s_{h'})$.

We let $\Pi_{\max}$ denote the set of all possible policies for the max player ($\Pi_{\min}$ for the min player). In the sequence form representation, $\Pi_{\max}$ is a convex compact subset of $\mathbb{R}^{XA}$ specified by the constraints $\mu_{1:h}(x_h, a_h) \ge 0$ and $\sum_{a_h \in \mathcal{A}} \mu_{1:h}(x_h, a_h) = \mu_{1:h-1}(x_{h-1}, a_{h-1})$ for all (h, x_h, a_h), where (x_{h-1}, a_{h-1}) is the unique pair of prior infoset and action that reaches x_h (understanding $\mu_0(x_0, a_0) = \mu_0(\emptyset) = 1$).

B.1.2 Reduction from classical definition of EFGs to TFAMDP

In this section, we show that solving EFGs with adversarial opponents can be reduced to solving TFAMDP. Formally, we prove the following proposition.

Proposition B.1. *For any EFG (i.e. POMG with tree structure and perfect recall assumptions) $(H, \mathcal{S}, \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B}, \mathbb{P}, r)$ with adversarial opponents' policies $\{\nu^t\}_{t\ge1}$, there exists an adversarial MDP $(\tilde{H}, \tilde{\mathcal{X}}, \tilde{\mathcal{A}}, \tilde{\mathcal{T}})$ with adversarial transition $\{\tilde{p}^t = \{p_h^t\}_{h\in\{0\}\cup[H]}\}_{t\ge1}$ and reward $\{\tilde{R}^t = \{\tilde{R}_h^t\}_{h\in[H]}\}_{t\ge1}$, so that for any policy sequences $\{\mu^t\}_{t\ge1}$, their joint distributions over the learner's trajectory $P(x_1, a_1, x_2, a_2, \dots, x_H, a_H, r)$ are exactly the same for all episodes $t \ge 1$.*

We remark that the joint distribution $P(x_1, a_1, x_2, a_2, \dots, x_H, a_H, r)$ gives a complete description about what the first player can obtain from the dynamic systems in both models. The joint distributions being the same for two models means that information-theoretically, the learner has no way to distinguish the two models, thus proving their equivalence.

Proof of Proposition B.1. In this section, we will use the notation in its original form $\mathcal{X}, \mathcal{A}, p, r$ to denote the quantity in EFGs while use their tilded form $\tilde{\mathcal{X}}, \tilde{\mathcal{A}}, \tilde{p}, \tilde{R}$ to denote the corresponding quantity in tree-form AMDP. It is not hard to see that in order to prove Proposition B.1 for all $t \ge 1$, it suffices to prove for a fixed t it is true.

Construction of AMDP we construct the corresponding AMDP using EFGs in the following way: we let $\tilde{H} = H$, $\tilde{\mathcal{X}} = \mathcal{X}$, $\tilde{\mathcal{A}} = \mathcal{A}$. Since EFG satisfies perfect recall assumption, which defines the immediate children function $\mathcal{C}$. We use the precisely same child function to define the tree structure $\tilde{\mathcal{T}}$ in AMDP. We define the adversarial transition according to the following equations:

$$\tilde{p}_1^t(x_1) := \sum_{s_1 \in x_1} p_1(s_1),$$

$$\tilde{p}_h^t(x_{h+1}|x_h, a_h) := \frac{\sum_{s_{h+1}\in x_{h+1}} \sum_{b_{h+1}\in\mathcal{B}} p_{1:h+1}(s_{h+1}) \nu_{1:h+1}^t(y_{h+1}(s_{h+1}), b_{h+1})}{\sum_{s_h \in x_h} \sum_{b_h \in \mathcal{B}} p_{1:h}(s_h) \nu_{1:h}^t(y_h(s_h), b_h)},$$

where $y_{h+1}(s_{h+1})$ and $y_h(s_h)$ are the infoset of opponent at $(h+1)$-th and h-th steps given state s_{h+1} and s_h respectively. We also define the adversarial reward distribution $\tilde{R}_H^t(\cdot|x_H, a_H)$ such that it gives the following distribution over reward $r \in [0, 1]$ for any fixed (x_H, a_H)

$$r = r(s_H, a_H, b_H) \text{ with probability } \frac{p_{1:H}(s_H)\nu_{1:H}^t(y_H(s_H), b_H)}{\sum_{s'_H \in x_H} \sum_{b'_H \in \mathcal{B}} p_{1:H}(s'_H)\nu_{1:H}^t(y_H(s'_H), b'_H)}.$$

And we set the adversarial reward $\tilde{R}_h^t(\cdot|x_h, a_h)$ to be zero (almost surely) for all $h \le H-1$ and all (x_h, a_h, t).

Proof of equivalence Denote $\tilde{P}^{\mu,t}$ as the probability of AMDP at episode t with policy μ; denote P^{μ,ν^t} as the probability of EFGs under policy μ and ν^t. It is very easy to check by induction over step h, that for any $h \in [H]$, and all policy μ simultaneously:

$$\tilde{P}^{\mu,t}(x_h, a_h) = P^{\mu,\nu^t}(x_h, a_h) = \sum_{s_h\in x_h} \sum_{b_h \in \mathcal{B}} \mu_{1:h}(x_h, a_h) p_{1:h}(s_h) \nu_{1:h}^t(y_h(s_h), b_h).$$

This proves that the joint distribution:

$$\tilde{P}^{\mu,t}(x_1, a_1, \dots, x_H, a_H) = P^{\mu,\nu^t}(x_1, a_1, \dots, x_H, a_H). \tag{29}$$

Finally, the construction of adversarial reward is such that its conditional distribution given (x_H, a_H) is exactly the same as the conditional distribution of the reward in the EFG:

$$\tilde{R}_H^t(r = r(s_H, a_H, b_H)|x_H, a_H) = P^{\mu,\nu^t}(r = r(s_H, a_H, b_H)|x_H, a_H),$$

which immediately gives that:

$$\tilde{P}^{\mu,t}(r_H|x_1, a_1, \dots, x_H, a_H) = P^{\mu,\nu^t}(r|x_1, a_1, \dots, x_H, a_H). \tag{30}$$

Combining (29) and (30), we finish the proof. □

B.2 Online-to-batch conversion

We consider an EFG with m players (e.g. using the definition in Section B.1), where each player faces an equivalent Tree-Form Adversarial MDP. For any product policy $\pi = \{\pi_i\}_{i\in[m]}$, let $\ell^{\pi_{-i}}$ denote the expected loss function for the i^{th} player if that the other players play policy π_{-i}. We define a correlated policy $\overline{\pi}$ as a probability distribution over product policies, i.e. $\pi \sim \overline{\pi}$ gives a product policy π.

An EFCE of the game is defined as follows [13, 23].

Definition B.1 (Extensive-form correlated equilibrium)**.** *A correlated policy $\overline{\pi}$ is an ε-approximate* Extensive-Form Correlated Equilibrium *(EFCE) of the EFG if*

$$\max_{i\in[m]} \max_{\phi\in\Phi_i^{\mathsf{Tr}}} \mathbb{E}_{\pi\sim\overline{\pi}}(\langle \phi\pi_i, \ell^{\pi_{-i}}\rangle - \langle \pi_i, \ell^{\pi_{-i}}\rangle) \le \varepsilon.$$

We say $\overline{\pi}$ is an (exact) EFCE if the above is equality.

When the game is played with product policies for T rounds, suppose the product policy at round t is π^t, the extensive-form trigger regret (6) for the i^{th} player becomes

$$\mathrm{Reg}_i^{\mathsf{Tr}}(T) = \max_{\phi\in\Phi_i^{\mathsf{Tr}}} \sum_{t=1}^{T} \left\langle \phi\pi_i^t - \pi_i^t, \ell^{\pi_{-i}^t} \right\rangle.$$

The following online-to-batch lemma for EFCE is standard, see e.g. [13].

Lemma B.1 (Online-to-batch for EFCE)**.** *Let $\{\pi^t = (\pi_i^t)_{i\in[n]}\}_{t\in[T]}$ be a sequence of product policies for all players over T rounds. Then, for the average (correlated) policy $\overline{\pi} = \mathrm{Unif}(\{\pi^t\}_{t=1}^T)$ is an ε-EFCE, where $\varepsilon = \max_{i\in[m]} \mathrm{Reg}_i^{\mathsf{Tr}}(T)/T$.*

B.3 Properties

For any $h < h'$ and $x_h \in \mathcal{X}_h$, we let $\mathcal{C}_{h'}(x_h, a_h) \equiv \{x \in \mathcal{X}_{h'} : x \succ (x_h, a_h)\}$ and $\mathcal{C}_{h'}(x_h) \equiv \{x \in \mathcal{X}_{h'} : x \succeq x_h\} = \cup_{a_h\in\mathcal{A}} \mathcal{C}_{h'}(x_h, a_h)$ denote the infosets within the h'-th step that are reachable from (i.e. children of) x_h or (x_h, a_h), respectively. For shorthand, let $\mathcal{C}(x_h, a_h) := \mathcal{C}_{h+1}(x_h, a_h)$ and $\mathcal{C}(x_h) := \mathcal{C}_{h+1}(x_h)$ denote the set of immediate children.

We define $X_{\succeq x_h}$ for any $x_h \in \mathcal{X}_h$ as

$$X_{\succeq x_h} := \sum_{h'=h}^{H} |\mathcal{C}_{h'}(x_h)|. \tag{31}$$

It can be interpreted as the number of infosets in the subtree rooted at x_h.

Lemma B.2. *The L^1 norm of a sequence form is upper bounded by $\|\Pi\|_1 \le X$.*

Proof. We can prove $\|\Pi^{x_h}\| \le X_{\succeq x_h}$ for all $h \in [0, H]$ and $x_h \in \mathcal{X}_h$ by backward induction over $h = H, \cdots, 1, 0$. When $h = H$, for each infoset x_h, the sequence form is just a probability distribution, which sums up to $\|\Pi^{x_h}\|_1 = 1 = |X_{\succeq x_h}|$. If the claim holds for $h+1$, consider an infoset x_h in the h-th level. By induction hypothesis we have

$$\|\Pi^{x_h}\|_1 = \max_{a_h} \sum_{x_{h+1}\succeq(x_h,a_h)} \|\Pi^{x_{h+1}}\|_1 \le \max_{a_h} \sum_{x_{h+1}\succeq(x_h,a_h)} |X_{\succeq x_{h+1}}| \le |X_{\succeq x_h}|.$$

So the equation above holds for any x_h. Setting $x_h = \emptyset$ gives $\|\Pi\|_1 \le X$ which completes the proof. □

Lemma B.3. *We have* $|\Phi_0^{\mathsf{Tr}}| \le XA^{\|\Pi\|_1+1}$.

Proof. By Proposition 5.1 of [25], $\mathcal{V} \le A^{\|\Pi\|_1}$. Since there are at most XA different infoset-action pair to be trigger, we have $|\Phi_0^{\mathsf{Tr}}| \le XA^{\|\Pi\|_1+1}$. □

C Proofs for Section 3.1 & 3.2

C.1 Incremental (OMD) form of Algorithm 2

We first present an incremental update of (λ^{t+1}, m^{t+1}) from (λ^t, m^t) as in Algorithm 4. We set the initial values of these variables as

$$\lambda^1_{x_g a_g} \propto_{x_g a_g} \exp\Big\{F^0_{x_g}\Big\}, \qquad m^1_{x_g a_g, h}(a_h|x_h) \propto_{a_h} \exp\Big\{\sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} F^0_{x_{h+1}}\Big\}, \tag{32}$$

where for any $x_h \succeq x_g$, $F^0_{x_h}$ is recursively defined as

$$F^0_{x_h} := \log \sum_{a_h\in\mathcal{A}} \exp\Big\{\sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} F^0_{x_{h+1}}\Big\}.$$

When the summation is over an empty set $\mathcal{C}(x_h, a_h)$, the sum should be understood as zero. Here $F^0_{x_h}$ has an intuitive meaning: it is the logarithm of the number of deterministic sequence-form policies starting from x_h, and can be computed by the above sum-product formulation recursively.

Algorithm 4 is computationally more efficient than Algorithm 2 when the loss estimator is sparse. For example, with bandit feedback, we need to update the loss matrix for at most H infoset-action pairs, and thus incur at most H^3 operations in Algorithm 4. On the contrary, Algorithm 2 requires $O((XA)^2)$ operations to update the policy in each iteration.

We now prove that Algorithm 4 and Algorithm 2 are actually equivalent.

Lemma C.1. *Given the same sequence of M^t, Algorithm 2 and Algorithm 4 outputs the same λ^t and m^t and thus the same ϕ^t.*

Proof. We only need to prove for any $x_g a_g, x_h$ and t, $F^t_{x_g a_g, x_h} = \sum_{s=1}^t \widetilde{F}^s_{x_g a_g, x_h}$. Then λ^t and m^t will be the same in Algorithm 2 and Algorithm 4.

We prove the above claim by induction. For the base case, the claim clearly holds if $h = H+1$ or $t = 1$ by definition. Assume this holds at $t-1$ and $h+1$, then at the h-th step in Algorithm 4,

$$\begin{aligned}
\widetilde{F}^t_{x_g a_g, x_h} &= \log \sum_{a_h\in\mathcal{A}} m^t_{x_g a_g, h}(a_h|x_h) \exp\Big\{-\eta M^t_{x_h a_h, x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} \widetilde{F}^t_{x_g a_g, x_{h+1}}\Big\} \\
&= \log \sum_{a_h\in\mathcal{A}} \exp\Big\{-\eta \sum_{s=1}^t M^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} F^t_{x_g a_g, x_{h+1}}\Big\} \\
&\quad - \log \sum_{a_h\in\mathcal{A}} \exp\Big\{-\eta \sum_{s=1}^{t-1} M^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} F^{t-1}_{x_g a_g, x_{h+1}}\Big\} \\
&= F^t_{x_g a_g, x_h} - F^{t-1}_{x_g a_g, x_h}.
\end{aligned}$$

Thus $F^t_{x_g a_g, x_h} = \sum_{s=1}^t \widetilde{F}^s_{x_g a_g, x_h}$. We completes the proof by noticing at $H+1$ step, $F^t_{x_g a_g, x_{H+1}} = \widetilde{F}^t_{x_g a_g, x_{H+1}} = 0$. □

Algorithm 4 EFCE-OMD (OMD form; equivalent FTRL form in Algorithm 2)

Require: Learning rate η.

Initialize $\lambda^1_{x_g a_g}$, and $m^1_{x_g a_g, h}(a_h|x_h)$, for all $(g, x_g, a_g, h, x_h, a_h)$ with $g \le h$ using Eq. (32).

for $t = 1, 2, \dots, T$ **do**

  Compute $\phi^t = \phi(\lambda^t, m^t)$ where ϕ is in Eq. (13).

  Compute the policy μ^t, which is a solution of the fixed point equation $\mu = \phi^t \mu$.

  Receive loss $\ell^t = \{\ell^t_h(x_h, a_h)\}_{(x_h, a_h) \in \mathcal{X} \times \mathcal{A}} \in \mathbb{R}^{XA}_{\ge 0}$.

  Compute matrix loss $M^t = \ell^t (\mu^t)^\top \in \mathbb{R}^{XA \times XA}_{\ge 0}$.

  For each $x_g a_g \in \mathcal{X} \times \mathcal{A}$, from the reverse order of x_h, compute $m^{t+1}_{x_g a_g, h}(a_h|x_h)$ and $\widetilde{F}^t_{x_g a_g, x_h}$

$$m^{t+1}_{x_g a_g, h}(a_h|x_h) \propto_{a_h} m^t_{x_g a_g, h}(a_h|x_h) \exp\Big\{ -\eta M^t_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} \widetilde{F}^t_{x_g a_g, x_{h+1}} \Big\},$$

$$\widetilde{F}^t_{x_g a_g, x_h} = \log \sum_{a_h \in \mathcal{A}} m^t_{x_g a_g, h}(a_h|x_h) \exp\Big\{ -\eta M^t_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} \widetilde{F}^t_{x_g a_g, x_{h+1}} \Big\},$$

  Compute $\lambda^{t+1}_{x_g a_g}$ as

$$\lambda^{t+1}_{x_g a_g} \propto_{x_g a_g} \lambda^t_{x_g a_g} \exp\Big\{ -\eta \langle I - E_{\succeq x_g a_g}, M^t \rangle + \widetilde{F}^t_{x_g a_g, x_g} \Big\}.$$

C.2 Proof of Lemma 2

By Line 6 of Algorithm 1, we have

$$p^{t+1}_\phi = \frac{p^t_\phi \cdot \exp\{-\eta \langle \phi \mu^t, \ell^t \rangle\}}{\sum_{\phi'} p^t_{\phi'} \cdot \exp\{-\eta \langle \phi' \mu^t, \ell^t \rangle\}} = \frac{p^t_\phi \cdot \exp\{-\eta \langle \phi, M^t \rangle\}}{\sum_{\phi'} p^t_{\phi'} \cdot \exp\{-\eta \langle \phi', M^t \rangle\}}. \tag{33}$$

Repeating this update and using the uniform initialization, we have

$$p^{t+1}_\phi = \frac{\exp\{-\eta \langle \phi, \sum_{s=1}^t M^s \rangle\}}{\sum_{\phi'} \exp\{-\eta \langle \phi', \sum_{s=1}^t M^s \rangle\}}.$$

As a result, we have

$$\phi^t = \sum_\phi p^t_\phi \phi = \frac{\sum_\phi \exp\{-\eta \langle \phi, \sum_{s=1}^{t-1} M^s \rangle\} \phi}{\sum_\phi \exp\{-\eta \langle \phi, \sum_{s=1}^{t-1} M^s \rangle\}} = -\nabla F^{\Phi_0}\left(\eta \sum_{s=1}^{t-1} M^s \right). \tag{34}$$

This proves the lemma. □

C.3 Proof of Lemma 3

For any $x_h \succeq x_g$, we define $F_{x_g a_g, x_h}(M)$ by

$$F_{x_g a_g, x_h}(M) := \log \sum_{m_{x_g a_g} \in \mathcal{V}^{x_h}} \exp(-\langle m_{x_g a_g} e^\top_{x_g a_g}, M \rangle).$$

Note that for any $\phi \in \Phi^{\mathsf{Tr}}_0$, there exists a unique $(g, x_g, a_g, m_{x_g a_g}) \in [H] \times \mathcal{X} \times \mathcal{A} \times \mathcal{V}^{x_g}$ such that $\phi = \phi_{x_g a_g \to m_{x_g a_g}}$. As a consequence, we have

$$\begin{aligned} F^{\mathsf{Tr}}(M) &= \log \sum_{\phi \in \Phi^{\mathsf{Tr}}_0} \exp(-\langle \phi, M \rangle) \\ &= \log \sum_{g, x_g, a_g} \sum_{m_{x_g a_g} \in \mathcal{V}^{x_g}} \exp(-\langle \phi_{x_g a_g \to m_{x_g a_g}}, M \rangle) \\ &= \log \sum_{g, x_g, a_g} \sum_{m_{x_g a_g} \in \mathcal{V}^{x_g}} \exp(-\langle I - E_{\succeq x_g a_g} + m_{x_g a_g} e^\top_{x_g a_g}, M \rangle) \end{aligned}$$

$$= \log \sum_{g, x_g, a_g} \exp\Big\{ -\langle I - E_{\succeq x_g a_g}, M\rangle + F_{x_g a_g, x_g}(M)\Big\}.$$

It remains to evaluate $F_{x_g a_g, x_h}(M)$ recurrently, which is handled by the structure of $\mathcal{V}^{x_h}$ as follows:

$$\begin{aligned}
& F_{x_g a_g, x_h}(M) \\
=& \log \sum_{m_{x_g a_g} \in \mathcal{V}^{x_h}} \exp(-\langle m_{x_g a_g} e_{x_g a_g}^\top, M\rangle) \\
=& \log \sum_{a_h \in \mathcal{A}} \exp(-M_{x_h a_h, x_g a_g}) \prod_{x_{h+1} \in \mathcal{C}(x_h a_h)} \sum_{m_{x_{h+1} a_{h+1}} \in \mathcal{V}^{x_{h+1}}} \exp(-\langle m_{x_{h+1} a_{h+1}} e_{x_g a_g}^\top, M\rangle) \\
=& \log \sum_{a_h \in \mathcal{A}} \exp\Big\{ -M_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} \log \sum_{m_{x_{h+1} a_{h+1}} \in \mathcal{V}^{x_{h+1}}} \exp(-\langle m_{x_{h+1} a_{h+1}} e_{x_g a_g}^\top, M\rangle)\Big\} \\
=& \log \sum_{a_h \in \mathcal{A}} \exp\Big\{ -M_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} F_{x_g a_g, x_{h+1}}(M)\Big\}.
\end{aligned}$$

This proves Eq. (9) and (10).

Calculating the gradient, we have

$$\begin{aligned}
-\nabla F^{\mathsf{Tr}}(M) =& \frac{\sum_{g, x_g, a_g} \exp\Big\{ -\langle I - E_{\succeq x_g a_g}, M\rangle + F_{x_g a_g, x_g}(M)\Big\}\big[I - E_{\succeq x_g a_g} - \nabla F_{x_g a_g, x_h}(M)\big]}{\sum_{g, x_g, a_g} \exp\Big\{ -\langle I - E_{\succeq x_g a_g}, M\langle\rangle + F_{x_g a_g, x_g}(M)\Big\}} \\
=& \sum_{g, x_g, a_g} \lambda_{x_g, a_g}\big[I - E_{\succeq x_g a_g} - \nabla F_{x_g a_g, x_h}(M)\big]. \qquad (35)
\end{aligned}$$

It remains to compute $\nabla F_{x_g a_g, x_h}(M)$. By the recurrent formula, we have

$$\begin{aligned}
& -\nabla F_{x_g a_g, x_h}(M) \\
=& \frac{\sum_{a_h \in \mathcal{A}} \exp\Big\{ -M_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} F_{x_g a_g, x_{h+1}}(M)\Big\}\big[e_{x_h a_h} e_{x_g a_g}^\top - \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} \nabla F_{x_g a_g, x_{h+1}}(M)\big]}{\sum_{a_h \in \mathcal{A}} \exp\Big\{ -M_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} F_{x_g a_g, x_{h+1}}(M)\Big\}} \\
=& \sum_{a_h \in \mathcal{A}} m_{x_g a_g, h}(a_h | x_h)\big[e_{x_h a_h} e_{x_g a_g}^\top + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} (-\nabla F_{x_g a_g, x_{h+1}})(M)\big].
\end{aligned}$$

This gives a recursion formula for $-\nabla F_{x_g a_g, x_h}(M)$. Solving this recursion formula, we get

$$-\nabla F_{x_g a_g, x_h}(M) = m_{x_g a_g} e_{x_g a_g}^\top,$$

Plugging this into Eq. (35) completes the proof. □

C.4 Runtime of Algorithm 2

Here we explain how Lemma 3 and its execution in Algorithm 2 is an $O(X^2A^2)$ time (in floating-point operations) efficient implementation of $-\nabla F^{\mathsf{Tr}}(M)$ for any matrix $M \in \mathbb{R}^{XA \times XA}$.

First, the function value $F^{\mathsf{Tr}}(M)$ can be recursively evaluated using (9) & (10), where we first evaluate (10) for any $x_g a_g \in \mathcal{X} \times \mathcal{A}$ recursively in a bottom-up fashion over $\{x_h : x_h \succeq x_g\}$ (i.e. the subtree rooted at x_g) up until $x_h = x_g$, and then plug in the resulting values of $F_{x_g a_g, x_g}(M)$ into (9) to obtain $F^{\mathsf{Tr}}(M)$. This process costs $O(XA)$ operations for each $x_g a_g$, so in total costs $O(X^2A^2)$ operations. Second, (11)-(13) show that the gradient can be obtained without much extra cost: By (13), $-\nabla F^{\mathsf{Tr}}(M)$ is determined by the parameters (λ, m), which then by (11) & (12) are exactly the ratios of the recursive log-sum-exps which we already evaluated in the previous step, and thus can be directly yielded (with cost of the same-order) while evaluating $F^{\mathsf{Tr}}(M)$. So the total runtime of the recursive computations in Lemma 3 (i.e. Algorithm 2) is $O(X^2A^2)$.

C.5 Dilated entropy and dilated KL

We introduce the subtree dilated entropy and subtree dilated KL divergence, a variant of dilated entropy and dilated KL divergence introduced in [30, 35, 34]. Here the modification we made is that we allow these quantities to be rooted at any infoset x_g. The original version is recovered when the choose the full game tree as the subtree. These quantities were further used to define the trigger dilated entropy and trigger dilated KL divergence as in Section 3.2.

Definition C.1 (Dilated entropy and Dilated KL divergence)**.** *The dilated entropy H_{x_g} rooted at x_g of subtree policy $\mu^{x_g} \in \Pi^{x_g}$ is defined as*

$$H_{x_g}(\mu^{x_g}) := \sum_{h=g}^{H} \sum_{(x_h,a_h)\succeq x_g} \mu^{x_g}_{g:h}(x_h, a_h) \log \mu^{x_g}_h(a_h|x_h). \tag{36}$$

The dilated KL divergence D_{x_g} rooted at x_g between two subtree policies $\mu^{x_g}, \nu^{x_g} \in \Pi^{x_g}$ is defined as

$$D_{x_g}(\mu^{x_g}\|\nu^{x_g}) := \sum_{h=g}^{H} \sum_{(x_h,a_h)\succeq x_g} \mu^{x_g}_{g:h}(x_h, a_h) \log \frac{\mu^{x_g}_h(a_h|x_h)}{\nu^{x_g}_h(a_h|x_h)}. \tag{37}$$

C.6 Proof of Lemma 4

We check solving optimization problem (17) will result in exactly the same form of Algorithm 2. The OMD form (18) is similar.

Using the definition of $H^{\mathsf{Tr}}(\lambda, m)$, the objective function in (17) can be written as

$$H(\lambda) + \sum_{g,x_g a_g} \lambda_{x_g a_g} \left[\eta\langle I - E_{\succeq x_g a_g}, \sum_{s=1}^{t} M^s\rangle + \eta\langle m_{x_g a_g}, \sum_{s=1}^{t} M^s_{\cdot,x_g a_g}\rangle + H_{x_g}(m_{x_g a_g}) \right].$$

First fix λ. and consider $m_{x_g a_g}$, which is just to minimize $\eta\langle m_{x_g a_g}, \sum_{s=1}^{t} M^s_{\cdot,x_g a_g}\rangle + H_{x_g}(m_{x_g a_g})$.

This is similar to form studied in Appendix B of [34] (or see Lemma G.3 for a full proof), which implies that, the optimum is achieved at

$$m^{t+1}_{x_g a_g,h}(a_h|x_h) \propto_{a_h} \exp\Big\{ \sum_{s=1}^{t} \big[-\eta M^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} F^t_{x_g a_g, x_{h+1}} \big] \Big\},$$

where

$$F^t_{x_g a_g, x_h} = \log \sum_{a_h\in\mathcal{A}} \exp\Big\{ \sum_{s=1}^{t} \big[-\eta M^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} F^t_{x_g a_g, x_{h+1}} \big] \Big\}.$$

Plug in the optimal $m_{x_g a_g}$, the object now becomes

$$H(\lambda) + \sum_{g,x_g a_g} \lambda_{x_g a_g} \left[\eta\langle I - E_{\succeq x_g a_g}, \sum_{s=1}^{t} M^s\rangle - F^t_{x_g a_g, x_g} \right].$$

This is a standard KL-regularized linear optimization problem on simplex. The optimum is achieved at

$$\lambda^{t+1}_{x_g a_g} \propto_{x_g a_g} \exp\Big\{ -\eta\langle I - E_{\succeq x_g a_g}, \sum_{s=1}^{t} M^s\rangle + F^t_{x_g a_g, x_g} \Big\}.$$

This gives the update of λ^{t+1} and m^{t+1} as in Algorithm 2. This completes the proof. □

D Proofs for Section 3.3

D.1 Proof of Theorem 5

Using regret bound of Φ-Hedge algorithm (Lemma A.1), we get

$$
\begin{aligned}
\mathrm{Reg}^{\mathsf{Tr}}(T) &\le \frac{\log|\Phi_0^{\mathsf{Tr}}|}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{\phi\in\Phi_0^{\mathsf{Tr}}} p_\phi^t\big(\langle\phi\mu^t,\ell^t\rangle\big)^2 \\
&\overset{(i)}{\le} \frac{\log|\Phi_0^{\mathsf{Tr}}|}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{\phi\in\Phi_0^{\mathsf{Tr}}} p_\phi^t H^2 = \frac{\log|\Phi_0^{\mathsf{Tr}}|}{\eta} + \frac{\eta H^2 T}{2}.
\end{aligned}
$$

Here, (i) uses $\langle\phi\mu^t,\ell^t\rangle\in[0,H]$. Note that $\Phi_0^{\mathsf{Tr}} := \bigcup_{g,x_g a_g}\bigcup_{v^{x_g}\in\mathcal{V}^{x_g}}\left\{\phi_{x_g a_g\to v^{x_g}}\right\}$ has cardinality upper bounded by $|\Phi_0^{\mathsf{Tr}}|\le XA^{\|\Pi\|_1+1}$ by Lemma B.3. Substitute this into the regret bound, we have

$$
\mathrm{Reg}^{\mathsf{Tr}}(T) \le \frac{\log(XA^{\|\Pi\|_1+1})}{\eta} + \frac{\eta H^2 T}{2} \le \frac{2\|\Pi\|_1\iota}{\eta} + \frac{\eta H^2 T}{2},
$$

where $\iota = \log(XA)$ is a log-term. Choosing $\eta = 2\sqrt{\|\Pi\|_1\iota/(H^2T)}$ gives $\mathrm{Reg}^{\mathsf{Tr}}(T) \le 2\sqrt{H^2\|\Pi\|_1\iota T}$, which completes the proof. □

D.2 Proof of Theorem 6

For the loss estimator $\widetilde{\ell}$, we have the following lemma (see Lemma 3 in Kozuno et al. [34]):

Lemma D.1. *Let $\delta\in(0,1)$ and $\gamma\in(0,\infty)$. Fix $h\in[H]$, and let $\alpha(x_h,a_h)\in[0,2\gamma]$ be some constant for each $(x_h,a_h)\in\mathcal{X}_h\times\mathcal{A}$. Then with probability at least $1-\delta$, we have*

$$
\sum_{t=1}^{T}\sum_{x_h\in\mathcal{X}_h,a_h\in\mathcal{A}} \alpha(x_h,a_h)\Big(\widetilde{\ell}_h^t(x_h,a_h) - \ell_h^t(x_h,a_h)\Big) \le \log\frac{1}{\delta}.
$$

Proof of Theorem 6. We have

$$
\begin{aligned}
\mathrm{Reg}^{\mathsf{Tr}}(T) &= \max_{\phi\in\Phi^{\mathsf{Tr}}}\sum_{t=1}^{T}\langle\mu^t-\phi\mu^t,\ell^t\rangle \\
&\le \underbrace{\sum_{t=1}^{T}\langle\mu^t,\ell^t-\widetilde{\ell}^t\rangle}_{\mathrm{BIAS}_1} + \underbrace{\max_{\phi\in\Phi^{\mathsf{Tr}}}\sum_{t=1}^{T}\langle\phi\mu^t,\widetilde{\ell}^t-\ell^t\rangle}_{\mathrm{BIAS}_2} + \underbrace{\max_{\phi\in\Phi^{\mathsf{Tr}}}\sum_{t=1}^{T}\langle\mu^t-\phi\mu^t,\widetilde{\ell}^t\rangle}_{\mathrm{REGRET}}.
\end{aligned}
$$

We use the following three lemmas to bound the terms above respectively. In these lemmas, $\iota = \log(3XA/\delta)$ is a log factor.

Lemma D.2 (Bound on BIAS_1). *With probability at least $1-\delta/3$, we have*

$$
\mathrm{BIAS}_1 \le H\sqrt{2T\iota} + \gamma XAT.
$$

Lemma D.3 (Bound on BIAS_2). *With probability at least $1-\delta/3$, we have*

$$
\mathrm{BIAS}_2 \le \|\Pi\|_1\iota/\gamma.
$$

Lemma D.4 (Bound on REGRET). *With probability at least $1-\delta/3$, we have*

$$
\mathrm{REGRET} \le \log|\Phi_0^{\mathsf{Tr}}|/\eta + \eta HXAT + \eta HXA\iota/\gamma..
$$

Lemma D.2, D.3, and D.4 bound bias terms and regret term respectively. Using these lemmas, we have with probability at least $1-\delta$,

$$
\mathrm{Reg}^{\mathsf{Tr}}(T) \le \frac{\log|\Phi_0^{\mathsf{Tr}}|}{\eta} + \eta HXAT + \eta HXA\iota/\gamma + \|\Pi\|_1\iota/\gamma + H\sqrt{2T\iota} + \gamma XAT.
$$

By Lemma B.3, $|\Phi_0^{\mathsf{Tr}}| \le XA^{\|\Pi\|_1+1}$. We further have

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \le \frac{2\|\Pi\|_1\iota}{\eta} + \eta HXAT + \eta HXA\iota/\gamma + \|\Pi\|_1\iota/\gamma + H\sqrt{2T\iota} + \gamma XAT.$$

Choosing $\gamma = \sqrt{\|\Pi\|_1\iota/(XAT)}$ and $\eta = \sqrt{\|\Pi\|_1\iota/(HXAT)}$ gives

$$\begin{aligned}\mathrm{Reg}^{\mathsf{Tr}}(T) &\le 5\sqrt{HXA\|\Pi\|_1 T\iota} + XA\iota\sqrt{H} + H\sqrt{2T\iota} \\ &\le \mathcal{O}(\sqrt{HXA\|\Pi\|_1\iota \cdot T} + XA\iota\sqrt{H}),\end{aligned}$$

where we uses $\|\Pi\|_1 \ge H$. Notice that there is a "trivial" bound $\mathrm{Reg}^{\mathsf{Tr}}(T) \le HT$. For $T \ge XA\iota/\|\Pi\|_1$, we have $XA\iota\sqrt{H} \le \sqrt{HXA\|\Pi\|_1\iota \cdot T}$, which gives $\mathrm{Reg}^{\mathsf{Tr}} \le \mathcal{O}(\sqrt{HXA\|\Pi\|_1\iota \cdot T})$; For $T \le XA\iota/\|\Pi\|_1$, we have $HT \le \sqrt{HXA\|\Pi\|_1\iota \cdot T}$, which gives $\mathrm{Reg}^{\mathsf{Tr}} \le HT \le \mathcal{O}(\sqrt{HXA\|\Pi\|_1\iota \cdot T})$. Therefore, we always have

$$\mathrm{Reg}^{\mathsf{Tr}} \le \mathcal{O}(\sqrt{HXA\|\Pi\|_1\iota \cdot T}).$$

This completes the proof. □

Here, we give the proofs of the lemmas we used above.

Proof of Lemma D.2. We further decompose BIAS_1 to two terms by

$$\mathrm{BIAS}_1 = \sum_{t=1}^{T}\left\langle \mu^t, \ell^t - \widetilde{\ell}^t \right\rangle = \underbrace{\sum_{t=1}^{T}\left\langle \mu^t, \ell^t - \mathbb{E}\left\{\widetilde{\ell}^t|\mathcal{F}^{t-1}\right\}\right\rangle}_{(A)} + \underbrace{\sum_{t=1}^{T}\left\langle \mu^t, \mathbb{E}\left\{\widetilde{\ell}^t|\mathcal{F}^{t-1}\right\} - \widetilde{\ell}^t\right\rangle}_{(B)}.$$

To bound (A), plug in the definition of loss estimator,

$$\begin{aligned}&\sum_{t=1}^{T}\left\langle \mu^t, \ell^t - \mathbb{E}\left\{\widetilde{\ell}^t|\mathcal{F}^{t-1}\right\}\right\rangle \\ &= \sum_{t=1}^{T}\sum_{h=1}^{H}\sum_{x_h,a_h} \mu_{1:h}^t(x_h,a_h)\left[\ell_h^t(x_h,a_h) - \frac{\mu_{1:h}^t(x_h,a_h)\ell_h^t(x_h,a_h)}{\mu_{1:h}^t(x_h,a_h)+\gamma}\right] \\ &= \sum_{t=1}^{T}\sum_{h=1}^{H}\sum_{x_h,a_h} \mu_{1:h}^t(x_h,a_h)\ell_h^t(x_h,a_h)\left[\frac{\gamma}{\mu_{1:h}^t(x_h,a_h)+\gamma}\right] \\ &\le \gamma\sum_{t=1}^{T}\sum_{h=1}^{H}\sum_{x_h,a_h} \ell_h^t(x_h,a_h) \le \gamma XAT,\end{aligned}$$

where the last inequality is by $\ell_h^t(x_h,a_h) \in [0,1]$.

To bound (B), first notice

$$\begin{aligned}\left\langle \mu^t, \widetilde{\ell}^t\right\rangle &= \sum_{h=1}^{H}\sum_{x_h,a_h}\mu_{1:h}^t(x_h,a_h)\frac{\mathbf{1}\left\{(x_h^t,a_h^t)=(x_h,a_h)\right\}\cdot(1-r_h^t)}{\mu_{1:h}^t(x_h,a_h)+\gamma} \\ &\le \sum_{h=1}^{H}\sum_{x_h,a_h}\mathbf{1}\left\{x_h = x_h^t, a_h = a_h^t\right\} = \sum_{h=1}^{H} 1 = H.\end{aligned}$$

Then by Azuma-Hoeffding, with probability at least $1-\delta/3$, we have

$$\sum_{t=1}^{T}\left\langle \mu^t, \mathbb{E}\left\{\widetilde{\ell}^t|\mathcal{F}^{t-1}\right\} - \widetilde{\ell}^t\right\rangle \le H\sqrt{2T\log(3/\delta)} \le H\sqrt{2T\iota}.$$

Combining the bounds for (A) and (B) gives the desired result. □

Proof of Lemma D.3. We have with probability at least $1-\delta/3$,

$$\begin{aligned}
\mathrm{BIAS}_2 &= \max_{\phi\in\Phi^{\mathsf{Tr}}}\sum_{t=1}^{T}\left\langle \phi\mu^t, \widetilde{\ell}^t-\ell^t\right\rangle \\
&= \max_{\phi\in\Phi^{\mathsf{Tr}}}\sum_{t=1}^{T}\sum_{h=1}^{H}\sum_{x_h,a_h}(\phi\mu^t)_{1:h}(x_h,a_h)\left[\widetilde{\ell}_h^t(x_h,a_h)-\ell_h^t(x_h,a_h)\right] \\
&= \max_{\phi\in\Phi^{\mathsf{Tr}}}\sum_{h=1}^{H}\sum_{x_h,a_h}\frac{(\phi\mu^t)_{1:h}(x_h,a_h)}{\gamma}\sum_{t=1}^{T}\gamma\left[\widetilde{\ell}_h^t(x_h,a_h)-\ell_h^t(x_h,a_h)\right] \\
&\overset{(i)}{\le} \frac{\log(3XA/\delta)}{\gamma}\max_{\phi\in\Phi^{\mathsf{Tr}}}\sum_{h=1}^{H}\sum_{x_h,a_h}(\phi\mu^t)_{1:h}(x_h,a_h) \\
&\le \|\Pi\|_1\iota/\gamma,
\end{aligned}$$

where (i) is by applying Lemma D.1 for each (x_h,a_h) pair. To be more specific, we choose $\alpha(x'_h,a'_h)=\gamma\mathbf{1}\{(x'_h,a'_h)=(x_h,a_h)\}$, then Lemma D.1 yields that

$$\sum_{t=1}^{T}\gamma\left[\widetilde{\ell}_h^t(x_h,a_h)-\ell_h^t(x_h,a_h)\right]\le\log\frac{3XA}{\delta}$$

with probability at least $1-\delta/(3XA)$. Then taking union bound gives the inequality in (i). $\square$

Proof of Lemma D.4. Note that by Lemma A.1, we have

$$\mathrm{REGRET}\le\frac{\log|\Phi_0^{\mathsf{Tr}}|}{\eta}+\frac{\eta}{2}\sum_{t=1}^{T}\sum_{\phi\in\Phi_0^{\mathsf{Tr}}}p_\phi^t(\langle\phi\mu^t,\widetilde{\ell}^t\rangle)^2.$$

To bound the second term, we have

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{\phi\in\Phi_0^{\mathsf{Tr}}}p_\phi^t(\langle\phi\mu^t,\widetilde{\ell}^t\rangle)^2 \\
&\le 2\sum_{h'\ge h}\sum_{t=1}^{T}\sum_{x_h,a_h}\sum_{(x_{h'},a_{h'})\in\mathcal{C}_{h'}(x_h,a_h)}\sum_{\phi\in\Phi_0^{\mathsf{Tr}}}p_\phi^t(\phi\mu^t)_{1:h}(x_h,a_h)(\phi\mu^t)_{1:h'}(x_{h'},a_{h'})\widetilde{\ell}_h^t(x_h,a_h)\widetilde{\ell}_{h'}^t(x_{h'},a_{h'}) \\
&\le 2\sum_{h'\ge h}\sum_{t=1}^{T}\sum_{x_h,a_h}\sum_{(x_{h'},a_{h'})\in\mathcal{C}_{h'}(x_h,a_h)}\sum_{\phi\in\Phi_0^{\mathsf{Tr}}}\frac{p_\phi^t(\phi\mu^t)_{1:h}(x_h,a_h)}{\mu_{1:h}^t(x_h,a_h)+\gamma}\widetilde{\ell}_{h'}^t(x_{h'},a_{h'}) \\
&\le 2\sum_{h'\ge h}\sum_{t=1}^{T}\sum_{x_h,a_h}\sum_{(x_{h'},a_{h'})\in\mathcal{C}_{h'}(x_h,a_h)}\sum_{\phi\in\Phi_0^{\mathsf{Tr}}}\frac{p_\phi^t(\phi\mu^t)_{1:h}(x_h,a_h)}{\mu_{1:h}^t(x_h,a_h)}\widetilde{\ell}_{h'}^t(x_{h'},a_{h'}) \\
&\overset{(i)}{=} 2\sum_{h'\ge h}\sum_{t=1}^{T}\sum_{x_h,a_h}\sum_{(x_{h'},a_{h'})\in\mathcal{C}_{h'}(x_h,a_h)}\widetilde{\ell}_{h'}^t(x_{h'},a_{h'}) \\
&\overset{(ii)}{\le} 2\sum_{h'\ge h}\Big(\sum_{t=1}^{T}\sum_{x_h,a_h}\sum_{(x_{h'},a_{h'})\in\mathcal{C}_{h'}(x_h,a_h)}\ell_{h'}^t(x_{h'},a_{h'})+X_hA\iota/\gamma\Big) \\
&\le 2HXAT+2HXA\iota/\gamma,
\end{aligned}$$

where (i) uses that μ^t is the solution of the fixed point equation $\mu^t=\sum_{\phi\in\Phi_0^{\mathsf{Tr}}}p_\phi^t\phi\mu^t$; (ii) is by Lemma D.1, which gives

$$\sum_{t=1}^{T}\sum_{(x_{h'},a_{h'})\in\mathcal{C}_{h'}(x_h,a_h)}\gamma\Big(\widetilde{\ell}_{h'}^t(x_{h'},a_{h'})-\ell_{h'}^t(x_{h'},a_{h'})\Big)\le\log\frac{3XA}{\delta}$$

with probability at least $1-\delta/(3XA)$ (choosing $\alpha(x'_h, a'_h) = \gamma \mathbf{1}\{(x'_h, a'_h) \in \mathcal{C}_{h'}(x_h, a_h)\}$ in the lemma). Then taking union bound yields that (ii) holds with probability at least $1-\delta/3$.

Finally, putting everything together, the lemma is proved. $\square$

E Omitted details in Section 4

E.1 Balanced exploration policy

We define the balanced exploration policies formally as introduced in Definition 2 in Bai et al. [6]. For any $1 \le h < h' \le H$ and $(x_{h'}, a_{h'}) \in \mathcal{X}_{h'} \times \mathcal{A}$, we denote $\mathcal{C}_h(x_{h'}, a_{h'}) := \{x_h \in \mathcal{X}_h : x_h \succ (x_{h'}, a_{h'})\}$ to be the set of children of $(x_{h'}, a_{h'})$ at layer h.

Definition E.1 (Balanced exploration policy)**.** *For any* $1 \le h \le H$*, the* balanced exploration policy for layer h*, denoted as* $\mu^{\star,h} \in \Pi$*, is defined as*

$$\mu^{\star,h}_{h'}(a_{h'}|x_{h'}) := \begin{cases} \dfrac{|\mathcal{C}_h(x_{h'}, a_{h'})|}{|\mathcal{C}_h(x_{h'})|}, & h' \in \{1, \dots, h-1\}, \\ 1/A, & h' \in \{h, \dots, H\}. \end{cases} \tag{38}$$

In words, at time steps $h' \le h-1$, the policy $\mu^{\star,h}$ plays actions proportionally to their number of descendants *within the h-th layer* of the game tree. Then at time steps $h' \ge h$, it plays the uniform policy.

A crucial property of the balanced exploration policy is the balancing property (Lemma C.4 in Bai et al. [6]). We also include it here for convience.

Lemma E.1 (Balancing property of $\mu^{\star,h}$)**.** *For policy* $\mu \in \Pi$ *and any* $h \in [H]$*, we have*

$$\sum_{(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}} \frac{\mu_{1:h}(x_h, a_h)}{\mu^{\star,h}_{1:h}(x_h, a_h)} = X_h A.$$

Moreover, for any x_g *as the root of a subtree, we have*

$$\sum_{(x_h, a_h) \succeq x_g} \frac{\mu_{g:h}(x_h, a_h)}{\mu^{\star,h}_{g:h}(x_h, a_h)} = |\mathcal{C}_h(x_g)| \cdot A.$$

Corollary E.1. *We have*

$$\mu^{\star,h}_{1:h}(x_h, a_h) \ge \frac{1}{X_h A}$$

for any $h \in [H]$ *and* $(x_h, a_h) \in \mathcal{X}_h \times \mathcal{A}$*.*

Proof. Choosing some deterministic policy μ s.t. $\mu_{1:h}(x_h, a_h) = 1$ in Lemma E.1 and noticing each term in the summation is non-negative, we have

$$\frac{\mu_{1:h}(x_h, a_h)}{\mu^{\star,h}_{1:h}(x_h, a_h)} \le X_h A.$$

$\square$

E.2 Algorithms: Balanced EFCE-OMD (in FTRL form and OMD form)

In this section, we present the algorithms omitted in Section 4. We begin with the Balanced EFCE-OMD (in FTRL form) as in Algorithm 3. This algorithm is actually equivalent to the algorithm as in Eq. (24) because of the following lemma, whose proof is similar to Lemma 3.

Lemma E.2. *For any loss matrix* $M \in \mathbb{R}^{XA \times XA}_{\ge 0}$*, recall that the* balanced EFCE log-partition function *as defined in Eq. (21)). Let* $\lambda = (\lambda_{x_g a_g})_{x_g a_g \in \mathcal{X} \times \mathcal{A}} \in \Delta_{XA}$ *and* $m = (m_{x_g a_g})_{x_g a_g \in \mathcal{X} \times \mathcal{A}} \in \mathcal{M}$ *be*

$$\lambda_{x_g a_g} \propto_{x_g a_g} \exp\left\{ \frac{1}{XA} \Big(-\eta \left\langle I - E_{\succeq x_g a_g}, M \right\rangle + F^{\star}_{x_g a_g, x_g} \Big) \right\} \tag{39}$$

Algorithm 5 Balanced EFCE-OMD (OMD form; equivalent FTRL form in Algorithm 3)

Require: Learning rate η, balanced exploration policy $\{\mu^{\star,h}\}_{h\in[H]}$.

Initialize $\lambda^1_{x_g a_g} \propto_{x_g a_g} \exp\{(X_{\succeq x_g}/X)\log A\}$, and $m^1_{x_g a_g,h}(a_h|x_h) = 1/A$, for all $(g, x_g, a_g, h, x_h, a_h)$ with $g \le h$.

for $t = 1, 2, \dots, T$ **do**

Compute $\phi^t = \phi(\lambda^t, m^t)$, where ϕ is as defined in Eq. (13).

Find a μ^t to be a solution of the fixed point equation $\mu^t = \phi^t \mu^t$.

Play policy μ^t, observe trajectory $(x_h^t, a_h^t, r_h^t)_{h\in[H]}$.

Form vector loss estimator $\widetilde{\ell}^{t,x_g a_g} = \{\widetilde{\ell}_h^{t,x_g a_g}(x_h, a_h)\}_{x_h a_h}$ for each $(g, x_g a_g)$ as in Eq. (23).

Compute matrix loss estimator $\widetilde{M}^t = \sum_{g,x_g,a_g} \mu^t_{x_g a_g} \widetilde{\ell}^{t,x_g a_g} e^\top_{x_g a_g}$.

For each $x_g a_g \in \mathcal{X}\times\mathcal{A}$, from the reverse order of x_h, compute $m^t_{x_g a_g,h}(a_h|x_h)$ and $F^{\star,t}_{x_g a_g,x_h}$

$$
\begin{aligned}
m^{t+1}_{x_g a_g,h}(a_h|x_h) \propto_{a_h} m^t_{x_g a_g,h}(a_h|x_h) \exp\Big\{\mu^{\star,h}_{g:h}(x_h, a_h) \\
\Big(-\eta \widetilde{M}^t_{x_h a_h, x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} \widetilde{F}^{\star,t}_{x_g a_g, x_{h+1}}\Big)\Big\}, \\
\widetilde{F}^{\star,t}_{x_g a_g,x_h} := \frac{1}{\mu^{\star,h}_{g:h}(x_h,a_h)} \log \sum_{a_h\in\mathcal{A}} m^t_{x_g a_g,h}(a_h|x_h) \exp\Big\{\mu^{\star,h}_{g:h}(x_h, a_h) \\
\times \big[-\eta \widetilde{M}^t_{x_h a_h, x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h a_h)} \widetilde{F}^{\star,t}_{x_g a_g, x_{h+1}}\big]\Big\}.
\end{aligned}
$$

Compute $\lambda^{t+1}_{x_g a_g}$ as

$$
\lambda^{t+1}_{x_g a_g} \propto_{x_g a_g} \lambda^t_{x_g a_g} \exp\Big\{\frac{1}{XA}\Big(-\eta\Big\langle I - E_{\succeq x_g a_g}, \widetilde{M}^t\Big\rangle + \widetilde{F}^{\star,t}_{x_g a_g, x_g}\Big)\Big\}. \tag{41}
$$

$$
m_{x_g a_g,h}(a_h|x_h) \propto_{a_h} \exp\Big\{\mu^{\star,h}_{g:h}(x_h,a_h)\Big(-\eta M_{x_h a_h, x_g a_g} + \sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} F^\star_{x_g a_g, x_{h+1}}\Big)\Big\}. \tag{40}
$$

then we have $-\nabla F^{\mathsf{Tr}}_{\mathsf{bal}}(M) = \phi(\lambda, m)$, where ϕ is as defined in Eq. (13).

We also present an efficient update of (λ^{t+1}, m^{t+1}) from (λ^t, m^t), which gives the OMD form of the Balanced EFCE-OMD algorithm as in Algorithm 5. Notice the initialization of Balanced EFCE-OMD is different from EFCE-OMD (Algorithm 4) due to the presence of the balanced exploration policy. Algorithm 3 and Algorithm 5 are indeed equivalent due to the following lemma, whose proof is similar to that of Lemma C.1.

Lemma E.3. *Given the same sequence of M^t, Algorithm 3 and Algorithm 5 outputs the same λ^t and m^t and thus the same ϕ^t.*

E.3 Equivalence to FTRL and OMD

Similar as Section 3.2, we show that the Balanced EFCE-OMD algorithm (Algorithm 3) is equivalent to FTRL with the balanced trigger dilated entropy, and OMD with the balanced dilated KL divergence, both over the (λ, m) parametrization.

We first introduce Balanced dilated entropy and balanced dilated KL divergence, and their trigger versions as below.

Definition E.2 (Balanced dilated entropy and balanced dilated KL divergence)**.** *The balanced dilated entropy $H^{\mathsf{bal}}_{x_g}$ rooted at x_g of subtree policy $\mu^{x_g} \in \Pi^{x_g}$ is defined as*

$$
H^{\mathsf{bal}}_{x_g}(\mu^{x_g}) := \sum_{h=g}^{H} \sum_{(x_h,a_h)\succeq x_g} \frac{\mu^{x_g}_{g:h}(x_h,a_h)}{\mu^{\star,h}_{g:h}(x_h,a_h)} \log \mu^{x_g}_h(a_h|x_h). \tag{42}
$$

The balanced dilated KL divergence $D^{\mathsf{bal}}_{x_g}$ rooted at x_g between two subtree policies $\mu^{x_g}, \nu^{x_g} \in \Pi^{x_g}$ is defined as

$$D^{\mathsf{bal}}_{x_g}(\mu^{x_g} \| \nu^{x_g}) := \sum_{h=g}^{H} \sum_{(x_h, a_h) \succeq x_g} \frac{\mu^{x_g}_{g:h}(x_h, a_h)}{\mu^{\star,h}_{g:h}(x_h, a_h)} \log \frac{\mu^{x_g}_h(a_h|x_h)}{\nu^{x_g}_h(a_h|x_h)}. \tag{43}$$

Definition E.3 (Balanced trigger dilated entropy and balanced trigger dilated KL divergence)**.** *The balanced trigger dilated entropy function on $(\lambda, m) \in \Delta_{XA} \times \mathcal{M}$ is defined as*

$$H^{\mathsf{Tr}}_{\mathsf{bal}}(\lambda, m) = XA \cdot H(\lambda) + \sum_{g, x_g, a_g} \lambda_{x_g a_g} H^{\mathsf{bal}}_{x_g}(m_{x_g a_g}). \tag{44}$$

The balanced trigger dilated KL divergence function on $(\lambda, m), (\lambda', m') \in \Delta_{XA} \times \mathcal{M}$ is defined as

$$D^{\mathsf{Tr}}_{\mathsf{bal}}(\lambda, m \| \lambda', m') = XA \cdot D_{\mathrm{KL}}(\lambda \| \lambda') + \sum_{g, x_g, a_g} \lambda_{x_g a_g} D^{\mathsf{bal}}_{x_g}(m_{x_g a_g} \| m'_{x_g a_g}). \tag{45}$$

The following lemma shows that the Balanced EFCE-OMD (Algorithm 3 and 5) are essentially FTRL with the balanced trigger dilated entropy, and OMD with the balanced tirgger dilated KL divergence. The proof of this lemma is similar to that of Lemma 4.

Lemma E.4 (Equivalent of Balanced EFCE-OMD to OMD/FTRL on (λ, m))**.** *For any sequence of loss functions $\{\widetilde{M}^t\}_{t \geq 1}$, the algorithm as in Eq. (24) is equivalent to (i.e. satisfy) the following FTRL update on $H^{\mathsf{Tr}}_{\mathsf{bal}}$ and OMD update on $D^{\mathsf{Tr}}_{\mathsf{bal}}$:*

$$(\lambda^{t+1}, m^{t+1}) = \ \arg\min_{\lambda, m} \Big[\eta \Big\langle \phi(\lambda, m), \textstyle\sum_{s=1}^{t} \widetilde{M}^s \Big\rangle + H^{\mathsf{Tr}}_{\mathsf{bal}}(\lambda, m)\Big], \tag{46}$$

$$(\lambda^{t+1}, m^{t+1}) = \ \arg\min_{\lambda, m} \Big[\eta \Big\langle \phi(\lambda, m), \widetilde{M}^t \Big\rangle + D^{\mathsf{Tr}}_{\mathsf{bal}}(\lambda, m \| \lambda^t, m^t)\Big], \tag{47}$$

with $\phi^{t+1} = \phi(\lambda^{t+1}, m^{t+1})$.

F Proof of Theorem 7

Here we restate the theorem for convenience.

Theorem F.1 (Sample complexity under bandit feedback)**.** *Run Balanced EFCE-OMD (Algorithm 3) with $\eta = \sqrt{XA\iota/H^4T}$ and $\gamma = 2\sqrt{XA\iota/H^2T}$. Then with probability at least $1-\delta$, we have the following extensive-form trigger regret bound,*

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \leq 200\sqrt{XAH^4T\iota},$$

where $\iota = \log(10HXA/\delta)$ is a log factor.

Proof. By the fixed point property of our algorithm, we have the regret decomposition

$$\begin{aligned}
&\mathrm{Reg}^{\mathsf{Tr}}(T) \\
&= \sup_{\phi^\star \in \Phi^{\mathsf{Tr}}} \sum_{t=1}^{T} \langle \mu^t - \phi^\star \mu^t, \ell^t \rangle = \sup_{\phi^\star \in \Phi^{\mathsf{Tr}}} \sum_{t=1}^{T} \langle \phi^t \mu^t - \phi^\star \mu^t, \ell^t \rangle = \sup_{\phi^\star \in \Phi^{\mathsf{Tr}}} \sum_{t=1}^{T} \langle \phi^t - \phi^\star, \ell^t (\mu^t)^\top \rangle \\
&\leq \underbrace{\sup_{\phi^\star \in \Phi^{\mathsf{Tr}}} \sum_{t=1}^{T} \langle \phi^t - \phi^\star, \widetilde{M}^t \rangle}_{\widetilde{\mathrm{REGRET}}^{\mathsf{Tr}}(T)} + \underbrace{\sum_{t=1}^{T} \langle \phi^t, \ell^t (\mu^t)^\top - \widetilde{M}^t \rangle}_{\mathrm{BIAS}_1} + \underbrace{\sup_{\phi^\star \in \Phi^{\mathsf{Tr}}} \sum_{t=1}^{T} \langle \phi^\star, \widetilde{M}^t - \ell^t (\mu^t)^\top \rangle}_{\mathrm{BIAS}_2}.
\end{aligned}$$

We bound the term $\widetilde{\mathrm{REGRET}}^{\mathsf{Tr}}(T)$, BIAS_1, and BIAS_2 in the following lemmas, whose proofs are presented in Section F.2 and F.3.

Lemma F.1 (Bound on $\widetilde{\mathrm{REGRET}}^{\mathsf{Tr}}(T)$)**.** *Assume that $\gamma \geq 2\eta H$. We have with probability at least $1-\delta/3$ that*

$$\widetilde{\mathrm{REGRET}}^{\mathsf{Tr}}(T) \leq \frac{XA\log(XA^2)}{\eta} + 22\eta H^4 T + \frac{38\eta H^3 XA\iota}{\gamma},$$

where $\iota = \log(10HXA/\delta)$ is a log factor.

Lemma F.2 (Bound on BIAS_1)**.** *We have with probability at least $1-\delta/3$ that*

$$\mathrm{BIAS}_1 \leq 2\gamma H^2 T + 2H\sqrt{T\iota},$$

where $\iota = \log(3/\delta)$ is a log factor.

Lemma F.3 (Bound on BIAS_2)**.** *We have with probability at least $1-\delta/3$ that*

$$\mathrm{BIAS}_2 \leq \frac{XA\iota}{\gamma},$$

where $\iota = \log(3XA/\delta)$ is a log factor.

By these three lemmas, whenever $\gamma \geq 2\eta H$, we have with probability at least $1-\delta$ that (for $\iota := \log(10HXA/\delta)$)

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \leq \frac{XA\log(XA^2)}{\eta} + 22\eta H^4 T + \frac{38\eta H^3 XA\iota}{\gamma} + 2\gamma H^2 T + 2H\sqrt{T\iota} + \frac{XA\iota}{\gamma}.$$

Taking $\eta = \sqrt{XA\iota/H^4T}$ and $\gamma = 2\sqrt{XA\iota/H^2T}$, we get

$$\mathrm{Reg}^{\mathsf{Tr}}(T) \leq 100\Big[\sqrt{XAH^4T\iota} + H^2XA\iota\Big].$$

Notice that we always have the "trivial" bound $\mathrm{Reg}^{\mathsf{Tr}}(T) \leq HT$. For $T \geq XA\iota$, we have $H^2XA\iota \leq \sqrt{XAH^4T\iota}$, which gives $\mathrm{Reg}^{\mathsf{Tr}} \leq 200\sqrt{H^4XAT\iota}$; For $T \leq XA\iota$, we have $HT \leq \sqrt{XAH^4T\iota}$, which gives $\mathrm{Reg}^{\mathsf{Tr}} \leq HT \leq \sqrt{H^4XAT\iota}$. Therefore, we always have

$$\mathrm{Reg}^{\mathsf{Tr}} \leq 200\sqrt{H^4XAT\iota}.$$

This gives the desired bound. □

The rest of this section is organized as follows. We introduce some notation in Section F.1. In Section F.2, we bound the regret term $\widetilde{\mathrm{REGRET}}^{\mathsf{Tr}}(T)$. In Section F.3, we bound the two bias terms BIAS_1 and BIAS_2.

F.1 Some preparations

Note that $m^t_{x_ga_g} \in \Pi^{x_g}$ is a subtree policy rooted at x_g, we define for any $(\widetilde{x}_h, \widetilde{a}_h) \succ x_g$

$$m^t_{x_ga_g,g:h}(\widetilde{x}_h, \widetilde{a}_h) := \prod_{h'=g}^{h} m^t_{x_ga_g,h'}(\widetilde{a}_{h'}|\widetilde{x}_{h'}),$$

where $(\widetilde{x}_g, \widetilde{a}_g, \widetilde{x}_{g+1}, \widetilde{a}_{g+1}, \cdots, \widetilde{x}_{h-1}, \widetilde{a}_{h-1})$ is the unique history leading to $(\widetilde{x}_h, \widetilde{a}_h)$, and $\widetilde{x}_g = x_g$.

Note that we have $\phi^t = \sum_{g,x_g,a_g} \lambda^t_{x_ga_g}(I - E_{\succeq x_ga_g} + m^t_{x_ga_g}e^\top_{x_ga_g})$ and $\phi^t\mu^t = \mu^t$. These two equations give

$$\sum_{g,x_g,a_g} \lambda^t_{x_ga_g}(I - E_{\succeq x_ga_g})\mu^t = \sum_{g,x_g,a_g} \lambda^t_{x_ga_g}\mu^t_{x_ga_g}m^t_{x_ga_g} \in \mathbb{R}^{XA}. \quad (48)$$

As a consequence, for any x_ga_g, we have

$$\lambda^t_{x_ga_g}\mu^t_{x_ga_g}m^t_{x_ga_g} \leq \sum_{g,x_g,a_g} \lambda^t_{x_ga_g}\mu^t = \mu^t. \quad (49)$$

Here $\lambda^t_{x_g a_g} \in \Delta_{XA}, \mu^t_{x_g a_g} = \mu^t_{1:g}(x_g, a_g)$ are two scalars, $m^t_{x_g a_g} \in \Pi^{x_g}$ and $\mu^t \in \Pi$ are two vectors of length XA, and the $\leq$ above is understood in an entrywise sense.

We also define (recall that $\{p^t_h\}_{h \in \{0\} \cup [H], t \geq 1}$ are the adversarial transition probabilities)

$$p^t(x_h) := p^t_0(x_1) \prod_{h'=1}^{h-1} p^t_{h'}(x_{h'+1}|x_{h'}, a_{h'}). \tag{50}$$

Note that $p^t(x_h) \in [0, 1]$. Furthermore, for any policy $\mu \in \Pi$ and any (h, t), we have

$$\sum_{x_h, a_h} \mu_{1:h}(x_h, a_h) p^t(x_h) = 1, \tag{51}$$

as the left-hand side is the probability of visiting *some* (x_h, a_h) in episode t using policy μ.

F.2 Proof of Lemma F.1

Recall that $\widetilde{\mathrm{REGRET}}^{\mathsf{Tr}}(T)$ is defined as

$$\widetilde{\mathrm{REGRET}}^{\mathsf{Tr}}(T) := \sup_{\phi^\star \in \Phi^{\mathsf{Tr}}} \sum_{t=1}^{T} \langle \phi^t - \phi^\star, \widetilde{M}^t \rangle.$$

First, we claim that

$$\sup_{\phi^\star \in \Phi^{\mathsf{Tr}}} \langle -\phi^\star, M \rangle = \sup_{\phi^\star \in \Phi^{\mathsf{Tr}}_0} \langle -\phi^\star, M \rangle \leq \frac{1}{\eta} F^{\mathsf{Tr}}_{\mathsf{bal}}(M).$$

for any $M \in \mathbb{R}^{XA \times XA}$. Indeed, the equality follows from $\Phi^{\mathsf{Tr}} = \mathrm{conv}\{\Phi^{\mathsf{Tr}}_0\}$. The inequality is due to the following argument: for any fixed M, the maximizer $\phi_{x_g a_g \to m_{x_g a_g}} \in \Phi^{\mathsf{Tr}}_0$ specifies a trigger sequence $x_g a_g$ and a deterministic subtree policy $m_{x_g a_g} \in \Pi^{x_g}$ starting from x_g. Replacing all the sums by this realization in the formula of $F^{\mathsf{Tr}}_{\mathsf{bal}}$ (c.f. Eq. (21)) and $F^\star_{x_g a_g, x_h}$ (c.f. Eq. (22)) exactly gives $F^{\mathsf{Tr}}(M) \geq \eta \langle -\phi_{x_g a_g \to m_{x_g a_g}}, M \rangle = \eta \sup_{\phi^\star \in \Phi^{\mathsf{Tr}}_0} \langle -\phi^\star, M \rangle$. This proves the claim.

This claim gives

$$\begin{aligned} \widetilde{\mathrm{REGRET}}^{\mathsf{Tr}}(T) &= \sup_{\phi^\star \in \Phi^{\mathsf{Tr}}} \sum_{t=1}^{T} \langle \phi^t - \phi^\star, \widetilde{M}^t \rangle = \sup_{\phi^\star \in \Phi^{\mathsf{Tr}}} \langle -\phi^\star, \sum_{t=1}^{T} \widetilde{M}^t \rangle + \sum_{t=1}^{T} \langle \phi^t, \widetilde{M}^t \rangle \\ &\leq \frac{1}{\eta} F^{\mathsf{Tr}}_{\mathsf{bal}}\Big(\sum_{t=1}^{T} \widetilde{M}^t \Big) + \sum_{t=1}^{T} \langle \phi^t, \widetilde{M}^t \rangle = \frac{1}{\eta} F^{\mathsf{Tr}}_{\mathsf{bal}}(0) + \sum_{t=1}^{T} D^t, \end{aligned} \tag{52}$$

where D^t is given by

$$D^t = \frac{1}{\eta} F^{\mathsf{Tr}}_{\mathsf{bal}}\Big(\eta \sum_{s=1}^{t} \widetilde{M}^s \Big) - \frac{1}{\eta} F^{\mathsf{Tr}}_{\mathsf{bal}}\Big(\eta \sum_{s=1}^{t-1} \widetilde{M}^s \Big) + \langle \phi^t, \widetilde{M}^t \rangle. \tag{53}$$

The following lemma gives bound on the initial entropy $F^{\mathsf{Tr}}_{\mathsf{bal}}(0)$ with proof in Section F.2.1.

Lemma F.4 (Bound on initial entropy)**.** *We have*

$$F^{\mathsf{Tr}}_{\mathsf{bal}}(0) = XA \log \sum_{g, x_g, a_g} \exp\{[X_{\succeq x_g} A \log A]/XA\} \leq XA \log(XA^2). \tag{54}$$

The following lemma gives a reformulation of the stability term D^t with proof in Section F.2.2.

Lemma F.5 (Reformulation of stability term via incremental update)**.** *We have*

$$D^t = \overline{F}^t / \eta + \langle \phi^t, \widetilde{M}^t \rangle, \tag{55}$$

where we have

$$\overline{F}^t = XA \log \sum_{g, x_g a_g} \lambda^t_{x_g a_g} \exp\Big\{ \frac{1}{XA}\big[-\eta\langle I - E_{\succeq x_g a_g}, \widetilde{M}^t\rangle + F^{\star,t}_{x_g a_g, x_g}\big] \Big\}, \tag{56}$$

$$F^{\star,t}_{x_g a_g, x_h} = \frac{1}{\mu^{\star,h}_{g:h}(x_h, a_h)} \log \sum_{a_h\in\mathcal{A}} m^t_{x_g a_g, h}(a_h|x_h) \exp\Big\{ \mu^{\star,h}_{g:h}(x_h, a_h) \times \big[-\eta \underbrace{\mu^t_{x_g a_g} \widetilde{\ell}^{t, x_g a_g}_h (x_h, a_h)}_{=\widetilde{M}^t_{x_h a_h, x_g a_g}} + \sum_{x_{h+1}\in\mathcal{C}(x_h, a_h)} F^{\star,t}_{x_g a_g, x_{h+1}} \big] \Big\}, \quad \forall (x_h, a_h)\succeq x_g, \tag{57}$$

and (note that $F^\star_{x_g a_g, x_g}(\mathbf{0})$ is as defined in Eq. (22) by plugging in $M = \mathbf{0}$)

$$\lambda^t_{x_g a_g} \propto_{x_g a_g} \exp\Big\{ \frac{1}{XA} F^\star_{x_g a_g, x_g}(\mathbf{0}) + \frac{1}{XA}\sum_{s=1}^{t-1}\Big(-\eta\langle I - E_{\succeq x_g a_g}, \widetilde{M}^s\rangle + F^{\star,s}_{x_g a_g, x_g}\Big)\Big\}, \tag{58}$$

$$m^t_{x_g a_g, h}(a_h|x_h) \propto_{a_h} \exp\Big\{ \mu^{\star,h}_{g:h}(x_h, a_h) \sum_{s=1}^{t-1}\Big(-\eta \underbrace{\mu^s_{x_g a_g}\widetilde{\ell}^{s, x_g a_g}_h(x_h, a_h)}_{=\widetilde{M}^s_{x_h a_h, x_g a_g}} + \sum_{x_{h+1}\in\mathcal{C}(x_h, a_h)} F^{\star,s}_{x_g a_g, x_{h+1}}\Big)\Big\}. \tag{59}$$

To upper bound D^t, note that by Lemma F.5 we have

$$\begin{aligned} D^t &= \overline{F}^t/\eta + \langle \phi^t, \widetilde{M}^t\rangle \\ &= \langle \phi^t, \widetilde{M}^t\rangle + \frac{XA}{\eta}\log \sum_{g, x_g a_g} \lambda^t_{x_g a_g}\exp\Big\{ \frac{1}{XA}\big[-\eta\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle + \Delta^t_{x_g a_g}\big]\Big\}, \end{aligned}$$

where $\Delta^t_{x_g a_g}$ is given by

$$\begin{aligned} \Delta^t_{x_g a_g} &:= F^{\star,t}_{x_g a_g, x_g} + \eta\langle m^t_{x_g a_g}\mu^t_{x_g a_g}, \widetilde{\ell}^{t, x_g a_g}\rangle \\ &= F^{\star,t}_{x_g a_g, x_g} + \eta\langle m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle. \end{aligned} \tag{60}$$

Note that $-\eta\langle m^t_{x_g a_g}\mu^t_{x_g a_g}, \widetilde{\ell}^{t, x_g a_g}\rangle$ is the linear term in the Taylor expansion of $F^{\star,t}_{x_g a_g, x_h}$ over variable $\widetilde{\ell}^t$ at 0, so $\Delta^t_{x_g a_g}$ can be understood as the nonlinear part within $F^{\star,t}_{x_g a_g, x_g}$. By convexity of $F^{\star,t}_{x_g a_g, x_h}$ as a function of $\widetilde{\ell}^t$, we have $\Delta^t_{x_g a_g}\ge 0$. Furthermore, we have the following almost sure upper bound of $\sup_{g, x_g a_g}\Delta^t_{x_g a_g}$ with proof in Section F.2.4.

Lemma F.6 (Bound on $\sup_{g, x_g a_g}\Delta^t_{x_g a_g}$)**.** *We have for all $t\in[T]$ that, almost surely,*

$$\frac{1}{XA}\sup_{g, x_g a_g}\Delta^t_{x_g a_g} \le \frac{2\eta^2}{\gamma^2}H^2.$$

Given this lemma, we further assume that $\gamma \ge 2H\eta$ so that $\frac{1}{XA}\sup_{g, x_g a_g}\Delta^t_{x_g a_g}\le 1$. Now we use elementary inequalities $\log(1+x)\le x$ and

$$e^{-x+c} \le 1 - (x-c) + \frac{e}{2}(x-c)^2 \le 1 - (x-c) + e(x^2 + c^2), \qquad \forall x\ge 0, c\le 1,$$

and (taking $c = \Delta^t_{x_g a_g}$ for each $(g, x_g a_g)$ below) we get

$$\begin{aligned} &\log\sum_{g, x_g a_g}\lambda^t_{x_g a_g}\exp\Big\{\frac{1}{XA}\big[-\eta\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle + \Delta^t_{x_g a_g}\big]\Big\} \\ &\le \sum_{g, x_g a_g}\lambda^t_{x_g a_g}\exp\Big\{\frac{1}{XA}\big[-\eta\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle + \Delta^t_{x_g a_g}\big]\Big\} - 1 \\ &\le \Big\{ \sum_{g, x_g a_g}\lambda^t_{x_g a_g}\Big(1 + \frac{1}{XA}\big[-\eta\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle + \Delta^t_{x_g a_g}\big] \end{aligned}$$

$$
+ \frac{e}{X^2A^2}\big[\eta^2\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle^2 + (\Delta^t_{x_g a_g})^2\big]\Big\} - 1
$$

$$
= -\frac{\eta}{XA}\langle \phi^t, \widetilde{M}^t\rangle + \frac{1}{XA}\sum_{g, x_g a_g} \lambda^t_{x_g a_g}\Delta^t_{x_g a_g}
$$

$$
+ \frac{e}{X^2A^2}\sum_{g, x_g a_g}\lambda^t_{x_g a_g}\Big(\eta^2\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle^2 + (\Delta^t_{x_g a_g})^2\Big).
$$

This gives that

$$
\begin{aligned}
D^t &\le \frac{1}{\eta}\sum_{g, x_g a_g}\lambda^t_{x_g a_g}\Delta^t_{x_g a_g} + \frac{e}{\eta XA}\sum_{g, x_g a_g}\lambda^t_{x_g a_g}\Big(\eta^2\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle^2 + (\Delta^t_{x_g a_g})^2\Big) \\
&\overset{(i)}{\le} \frac{1}{\eta}\sum_{g, x_g a_g}\lambda^t_{x_g a_g}\Delta^t_{x_g a_g}\left(1 + \frac{e}{XA}\sup_{g, x_g a_g}\Delta^t_{x_g a_g}\right) + \frac{e\eta}{XA}\sum_{g, x_g a_g}\lambda^t_{x_g a_g}\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle^2 \\
&\overset{(ii)}{\le} \underbrace{\frac{4}{\eta}\sum_{g, x_g a_g}\lambda^t_{x_g a_g}\Delta^t_{x_g a_g}}_{\mathrm{I}_t} + \underbrace{\frac{e\eta}{XA}\sum_{g, x_g a_g}\lambda^t_{x_g a_g}\langle I - E_{\succeq x_g a_g} + m^t_{x_g a_g} e^\top_{x_g a_g}, \widetilde{M}^t\rangle^2}_{\mathrm{II}_t}.
\end{aligned} \tag{61}
$$

Above, (i) used $\Delta^t_{x_g a_g} \ge 0$, and (ii) used Lemma F.6 and $\gamma \ge 2\eta H$.

Next, we use the following lemmas to bound $\sum_{t=1}^T \mathrm{I}_t$ and $\sum_{t=1}^T \mathrm{II}_t$, with proofs in Section F.2.5 and F.2.6.

Lemma F.7 (Bound on $\sum_{t=1}^T \mathrm{I}_t$)**.** *With probability at least $1 - \delta/10$, we have*

$$
\sum_{t=1}^T \mathrm{I}_t \le 16\eta H^3 T + \frac{32\eta H^3 XA\iota}{\gamma},
$$

where $\iota := \log(10H/\delta)$ is a log factor.

Lemma F.8 (Bound on $\sum_{t=1}^T \mathrm{II}_t$)**.** *With probability at least $1 - \delta/10$, we have*

$$
\sum_{t=1}^T \mathrm{II}_t \le 6\eta HT + \frac{6\eta HXA\iota}{\gamma},
$$

where $\iota := \log(10XA/\delta)$ is a log factor.

Combining Eq. (52), Lemma F.4 and F.5, Eq. (61), and Lemma F.7 and F.8, we have

$$
\widetilde{\mathrm{REGRET}}^{\mathsf{Tr}}(T) \le \frac{XA\log(XA^2)}{\eta} + 22\eta H^4 T + \frac{38\eta H^3 XA\iota}{\gamma}
$$

with probability at least $1 - \delta/3$, where $\iota := \log(10XAH/\delta)$ is a log factor. This completes the proof of Lemma F.1. □

F.2.1 Proof of Lemma F.4

Proof of Lemma F.4. By the definition of balanced EFCE log-partition function (see (21) and (22)), we have

$$
F^{\mathsf{Tr}}_{\mathsf{bal}}(\mathbf{0}) = XA\log\sum_{g, x_g, a_g}\exp\Big\{\frac{1}{XA}\big[F^\star_{x_g a_g, x_g}(\mathbf{0})\big]\Big\},
$$

where for any $x_h \succeq x_g$,

$$
F^\star_{x_g a_g, x_h}(\mathbf{0}) = \frac{1}{\mu^{\star,h}_{g:h}(x_h, a_h)}\log\sum_{a_h}\exp\Big\{\mu^{\star,h}_{g:h}(x_h, a_h)\big[\sum_{x_{h+1}\in\mathcal{C}(x_h, a_h)} F^\star_{x_g a_g, x_{h+1}}(\mathbf{0})\big]\Big\}. \tag{62}
$$

So we only need to prove that $F^\star_{x_g a_g, x_g}(\mathbf{0}) = X_{\succeq x_g} A \log A$. In fact, we can use backward induction to prove the following: for any $x_g \in \mathcal{X}_g$ and $x_h \in \mathcal{C}_h(x_g)$, we have

$$F^\star_{x_g a_g, x_h}(\mathbf{0}) = \sum_{h'=h}^{H} \frac{|\mathcal{C}_{h'}(x_h)|}{\mu^{\star:h'}_{g:h-1}(x_{h-1}, a_{h-1})} A \log A, \tag{63}$$

(with convention $\mu^{\star,h'}_{g:g-1} = 1$) where x_{h-1}, a_{h-1} is uniquely determined as $x_g \prec (x_{h-1}, a_{h-1}) \prec x_h$. It is easy to see that, choosing $x_h = x_g$ in (63) gives $F^\star_{x_g a_g, x_g}(\mathbf{0}) = X_{\succeq x_g} A \log A$.

Next we prove (63). We use backward induction on h. When $h = H$, from (62), for any $x_H \in \mathcal{C}_H(x_g)$, we have

$$F^\star_{x_g a_g, x_H}(\mathbf{0}) = \frac{1}{\mu^{\star,H}_{g:H}(x_H, a_H)} \log A = \frac{1}{\mu^{\star,H}_{g:H-1}(x_{H-1}, a_{H-1})} A \log A.$$

This proves the base case.

Now suppose (63) is true for $h+1$ where $h \in [g, H-1]$. By the recursive formula and the inductive hypothesis, for any $x_h \in \mathcal{C}_h(x_g)$, we have

$$\begin{aligned}
&F^\star_{x_g a_g, x_h}(\mathbf{0}) \\
&= \frac{1}{\mu^{\star,h}_{g:h}(x_h, a_h)} \log \sum_{a_h} \exp\Big\{\mu^{\star,h}_{g:h}(x_h, a_h)\big[\sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} F^\star_{x_g a_g, x_{h+1}}(\mathbf{0})\big]\Big\} \\
&= \frac{1}{\mu^{\star,h}_{g:h}(x_h, a_h)} \log \sum_{a_h} \exp\Big\{\mu^{\star,h}_{g:h}(x_h, a_h)\big[\sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} \sum_{h'=h+1}^{H} \frac{|\mathcal{C}_{h'}(x_{h+1})|}{\mu^{\star,h'}_{g:h}(x_h, a_h)} A \log A\big]\Big\}.
\end{aligned}$$

Then by the definition of the balanced policies (E.1), we have

$$\begin{aligned}
&\sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} \frac{|\mathcal{C}_{h'}(x_{h+1})|}{\mu^{\star,h'}_{g:h}(x_h, a_h)} \\
&= \frac{\sum_{x_{h+1}\in\mathcal{C}(x_h,a_h)} |\mathcal{C}_{h'}(x_{h+1})|}{\mu^{\star,h'}_{g:h-1}(x_{h-1}, a_{h-1})} \cdot \frac{|\mathcal{C}_{h'}(x_h)|}{|\mathcal{C}_{h'}(x_h, a_h)|} = \frac{|\mathcal{C}_{h'}(x_h)|}{\mu^{\star,h'}_{g:h-1}(x_{h-1}, a_{h-1})},
\end{aligned}$$

which is also independent of a_h. So we have

$$\begin{aligned}
&F^\star_{x_g a_g, x_h}(\mathbf{0}) \\
&= \frac{1}{\mu^{\star,h}_{g:h}(x_h, a_h)} \log \Big\{A \cdot \exp\Big\{\mu^{\star,h}_{g:h}(x_h, a_h)\big[\sum_{h'=h+1}^{H} \frac{|\mathcal{C}_{h'}(x_h)|}{\mu^{\star,h'}_{g:h-1}(x_{h-1}, a_{h-1})} A \log A\big]\Big\}\Big\} \\
&= \frac{\log A}{\mu^{\star,h}_{g:h}(x_h, a_h)} + \sum_{h'=h+1}^{H} \frac{|\mathcal{C}_{h'}(x_h)|}{\mu^{\star,h'}_{g:h-1}(x_{h-1}, a_{h-1})} A \log A \\
&= \sum_{h'=h}^{H} \frac{|\mathcal{C}_{h'}(x_h)|}{\mu^{\star:h'}_{g:h-1}(x_{h-1}, a_{h-1})} A \log A.
\end{aligned}$$

This proves (63), and thus we proved the first equation in Eq. (54). The inequality in Eq. (54) is direct since $\exp\{[X_{\succeq x_g} A \log A]/XA\} \le A$. This proves the lemma. □

F.2.2 Proof of Lemma F.5

Proof of Lemma F.5. We only need to verify that

$$\overline{F}^t := F^{\mathsf{Tr}}_{\mathsf{bal}}\Big(\eta \sum_{s=1}^{t} \widetilde{M}^s\Big) - F^{\mathsf{Tr}}_{\mathsf{bal}}\Big(\eta \sum_{s=1}^{t-1} \widetilde{M}^s\Big) \tag{64}$$

can be computed via recursive formulas (56)-(59).

For any $x_g a_g$, $x_h \succeq x_g$, define

$$G^{\star,t}_{x_g a_g, x_h} := F^{\star}_{x_g a_g, x_h}\left(\eta \sum_{s=1}^{t} \widetilde{M}^s\right) - F^{\star}_{x_g a_g, x_h}\left(\eta \sum_{s=1}^{t-1} \widetilde{M}^s\right).$$

By the definition of $F^{\star}_{x_g a_g, x_h}$ as in Eq. (22), we have

$$\begin{aligned}
&G^{\star,t}_{x_g a_g, x_h} = \frac{1}{\mu^{\star}_{g:h}(x_h, a_h)} \\
&\quad \times \log \frac{\sum_{a_h \in \mathcal{A}} \exp\left\{\mu^{\star,h}_{g:h}(x_h, a_h) \times \left\{F^{\star}_{x_g a_g, x_h}(\mathbf{0}) + \sum_{s=1}^{t}\left[-\eta \widetilde{M}^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} G^{\star,s}_{x_g a_g, x_{h+1}}\right]\right\}\right\}}{\sum_{a_h \in \mathcal{A}} \exp\left\{\mu^{\star,h}_{g:h}(x_h, a_h) \times \left\{F^{\star}_{x_g a_g, x_h}(\mathbf{0}) + \sum_{s=1}^{t-1}\left[-\eta \widetilde{M}^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} G^{\star,s}_{x_g a_g, x_{h+1}}\right]\right\}\right\}} \\
&= \frac{1}{\mu^{\star}_{g:h}(x_h, a_h)} \log \sum_{a_h \in \mathcal{A}} n^t_{x_g a_g, h}(a_h | x_h) \exp\left\{\mu^{\star,h}_{g:h}(x_h, a_h) \times \left[-\eta \widetilde{M}^t_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} G^{\star,t}_{x_g a_g, x_{h+1}}\right]\right\},
\end{aligned}$$

where

$$\begin{aligned}
n^t_{x_g a_g, h}(a_h | x_h) \propto_{a_h} \exp\Big\{&\mu^{\star,h}_{g:h}(x_h, a_h) F^{\star}_{x_g a_g, x_h}(\mathbf{0}) \\
&+ \mu^{\star,h}_{g:h}(x_h, a_h) \sum_{s=1}^{t-1}\Big(-\eta \widetilde{M}^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} G^{\star,s}_{x_g a_g, x_{h+1}}\Big)\Big\}.
\end{aligned}$$

Because $\mu^{\star,h}_{g:h}(x_h, a_h)$ and $F^{\star}_{x_g a_g, x_h}(\mathbf{0})$ are independent of a_h for any $(x_h, a_h) \succeq (x_g, a_g)$ (see proof of Lemma F.4), we have

$$n^t_{x_g a_g, h}(a_h | x_h) \propto_{a_h} \exp\Big\{\mu^{\star,h}_{g:h}(x_h, a_h) \sum_{s=1}^{t-1}\Big(-\eta \widetilde{M}^s_{x_h a_h, x_g a_g} + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} G^{\star,s}_{x_g a_g, x_{h+1}}\Big)\Big\}.$$

So $G^{\star,t}_{x_g a_g, x_h}$ and $F^{\star,t}_{x_g a_g, x_h}$ (c.f. Eq. (57)) have the same recursive formula, which means that $G^{\star,t}_{x_g a_g, x_h} = F^{\star,t}_{x_g a_g, x_h}$ for any $x_g a_g$ and $x_h \succeq x_g$.

Finally, by the definition of $\overline{F}^t$ as in Eq. (64) and the definition of $F^{\mathsf{Tr}}_{\mathsf{bal}}$ as in Eq. (21), we have

$$\begin{aligned}
&\overline{F}^t \\
&= XA \log \frac{\sum_{g, x_g, a_g} \exp\left\{\frac{1}{XA}\left[-\langle I - E_{\succeq x_g a_g}, \eta \sum_{s=1}^{t} \widetilde{M}^s\rangle + F^{\star}_{x_g a_g, x_g}(\eta \sum_{s=1}^{t} \widetilde{M}^s)\right]\right\}}{\sum_{g, x_g, a_g} \exp\left\{\frac{1}{XA}\left[-\langle I - E_{\succeq x_g a_g}, \eta \sum_{s=1}^{t-1} \widetilde{M}^s\rangle + F^{\star}_{x_g a_g, x_g}(\eta \sum_{s=1}^{t-1} \widetilde{M}^s)\right]\right\}} \\
&= XA \log \frac{\sum_{g, x_g, a_g} \exp\left\{\frac{1}{XA} F^{\star}_{x_g a_g, x_g}(\mathbf{0})\right\} \exp\left\{\frac{1}{XA}\left[\sum_{s=1}^{t}\left(-\langle I - E_{\succeq x_g a_g}, \eta \widetilde{M}^s\rangle + F^{\star,s}_{x_g a_g, x_g}\right)\right]\right\}}{\sum_{g, x_g, a_g} \exp\left\{\frac{1}{XA} F^{\star}_{x_g a_g, x_g}(\mathbf{0})\right\} \exp\left\{\frac{1}{XA}\left[\sum_{s=1}^{t-1}\left(-\langle I - E_{\succeq x_g a_g}, \eta \widetilde{M}^s\rangle + F^{\star,s}_{x_g a_g, x_g}\right)\right]\right\}} \\
&= XA \log \sum_{g, x_g a_g} \lambda^t_{x_g a_g} \exp\left\{\frac{1}{XA}\left[-\eta \langle I - E_{\succeq x_g a_g}, \widetilde{M}^t\rangle + F^{\star,t}_{x_g a_g, x_g}\right]\right\},
\end{aligned}$$

where

$$\lambda^t_{x_g a_g} \propto_{x_g a_g} \exp\left\{\frac{1}{XA} F^{\star}_{x_g a_g, x_g}(\mathbf{0}) + \frac{1}{XA} \sum_{s=1}^{t-1}\left(-\eta \langle I - E_{\succeq x_g a_g}, \widetilde{M}^s\rangle + F^{\star,s}_{x_g a_g, x_g}\right)\right\}.$$

This proves the lemma. □

F.2.3 Bound on $\Delta^t_{x_g a_g}$ via Hessian

The following lemma can be proved similar to Lemma D.11 in Bai et al. [6] by calculating the Hessian of $\Delta^t_{x_g a_g}$ with respect to $\widetilde{\ell}^{t, x_g a_g}$. This result is the starting point of both Lemma F.6 and Lemma F.7.

Lemma F.9 (Bound on $\Delta^t_{x_g a_g}$). *We have, almost surely,*

$$\Delta^t_{x_g a_g} \le 2\eta^2 \sum_{g\le h\le h'\le H} \sum_{h''=g}^{h} \mu^{\star,h''}_{g:h''}(x^t_{h''}, a^t_{h''}) m^t_{x_g a_g, h''+1:h'}(x^t_{h'}, a^t_{h'}) m^t_{x_g a_g, g:h}(x^t_h, a^t_h)$$
$$\times \widetilde{\ell}^{t,x_g a_g}_h(x^t_h, a^t_h) \widetilde{\ell}^{t,x_g a_g}_{h'}(x^t_{h'}, a^t_{h'}) \cdot (\mu^t_{x_g a_g})^2 \mathbf{1}\left\{(x^t_{h'}, a^t_{h'}) \succeq (x^t_h, a^t_h) \succeq x_g\right\}$$
$$\le 2\eta^2 \sum_{g\le h\le h'\le H} \sum_{h''=g}^{h} \sum_{x_{h'}, a_{h'}} \mu^{\star,h''}_{g:h''}(x_{h''}, a_{h''}) m^t_{x_g a_g, h''+1:h'}(x_{h'}, a_{h'}) m^t_{x_g a_g, g:h}(x_h, a_h)(\mu^t_{x_g a_g})^2$$
$$\times \frac{\mathbf{1}\left\{x^t_h, a^t_h = x_h, a_h\right\}}{(\mu^t_{1:h}(x_h, a_h) + \gamma(\mu^{\star,h}_{1:h}(x_h, a_h) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h) \mathbf{1}\{x_h \succeq x_g\}))}$$
$$\times \frac{\mathbf{1}\left\{x^t_{h'}, a^t_{h'} = x_{h'}, a_{h'}\right\}}{(\mu^t_{1:h'}(x_{h'}, a_{h'}) + \gamma(\mu^{\star,h'}_{1:h'}(x_{h'}, a_{h'}) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h'}(x_{h'}, a_{h'}) \mathbf{1}\{x_{h'} \succeq x_g\}))}$$
$$\times \mathbf{1}\left\{(x_{h'}, a_{h'}) \succeq (x_h, a_h) \succeq x_g\right\}.$$

Proof. First, notice that $F^{\star,t}_{x_g a_g, x_h}$ has a similar form as Ξ^t_1 in Appendix D.6 of [6] with three minor differences:

- $m^t_{x_g a_g, h}$ is used instead of their μ^t_h for layer h,
- $\mu^t_{x_g a_g} \widetilde{\ell}^{t,x_g a_g}_h$ is used instead of their $\widetilde{\ell}^t_h$ for layer h,
- We terminate the induction argument earlier at (x_g, a_g) instead of the root of the full tree.

Therefore $\Delta^t_{x_g a_g}$ defined as below (cf. (60)) is exactly the non-linear part (i.e. remainder term of the first-order Taylor expansion with respect to $\mu^t_{x_g a_g} \widetilde{\ell}^{t,x_g a_g}$ around 0) within $F^{\star,t}_{x_g a_g, x_g}$:

$$\Delta^t_{x_g a_g} := F^{\star,t}_{x_g a_g, x_h} + \eta \langle m^t_{x_g a_g} \mu^t_{x_g a_g}, \widetilde{\ell}^{t,x_g a_g} \rangle.$$

Further taking the above differences into account and following exactly the same analysis as in Appendix D.6.1 of [6], we get the first inequality as claimed. The second inequality follows by definition of the loss estimator $\widetilde{\ell}^{t,x_g a_g}_h$ in (23). □

F.2.4 Proof of Lemma F.6

Proof of Lemma F.6. By Lemma F.9, for any $x_g a_g$, we have

$$\Delta^t_{x_g a_g}$$
$$\le 2\eta^2 \sum_{g\le h\le h'\le H} \sum_{h''=g}^{h} \sum_{x_{h'}, a_{h'}} \mu^{\star,h''}_{g:h''}(x_{h''}, a_{h''}) m^t_{x_g a_g, h''+1:h'}(x_{h'}, a_{h'}) m^t_{x_g a_g, g:h}(x_h, a_h)(\mu^t_{x_g a_g})^2$$
$$\times \frac{\mathbf{1}\left\{x^t_h, a^t_h = x_h, a_h\right\}}{(\mu^t_{1:h}(x_h, a_h) + \gamma(\mu^{\star,h}_{1:h}(x_h, a_h) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h) \mathbf{1}\{x_h \succeq x_g\}))}$$
$$\times \frac{\mathbf{1}\left\{x^t_{h'}, a^t_{h'} = x_{h'}, a_{h'}\right\}}{(\mu^t_{1:h'}(x_{h'}, a_{h'}) + \gamma(\mu^{\star,h'}_{1:h'}(x_{h'}, a_{h'}) + \mu^t_{x_g a_g} m^t_{x_g a_g, g:h'}(x_{h'}, a_{h'}) \mathbf{1}\{x_{h'} \succeq x_g\}))}$$
$$\times \mathbf{1}\left\{(x_{h'}, a_{h'}) \succeq (x_h, a_h) \succeq x_g\right\}$$
$$\le 2\eta^2 \sum_{g\le h\le h'\le H} \sum_{h''=g}^{h} \sum_{x_{h'}, a_{h'}} \mu^{\star,h''}_{g:h''}(x_{h''}, a_{h''}) m^t_{x_g a_g, h''+1:h'}(x_{h'}, a_{h'}) m^t_{x_g a_g, g:h}(x_h, a_h)(\mu^t_{x_g a_g})^2$$
$$\times \frac{\mathbf{1}\left\{x^t_h, a^t_h = x_h, a_h\right\}}{\gamma \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h)} \times \frac{\mathbf{1}\left\{x^t_{h'}, a^t_{h'} = x_{h'}, a_{h'}\right\}}{\gamma \mu^{\star,h'}_{1:h'}(x_{h'}, a_{h'})} \times \mathbf{1}\left\{(x_{h'}, a_{h'}) \succeq (x_h, a_h) \succeq x_g\right\}$$

$$\le \frac{2\eta^2}{\gamma^2} \sum_{g\le h\le h'\le H} \sum_{h''=g}^{h} \sum_{x_{h'},a_{h'}} \frac{\mu^t_{x_ga_g}\mu^{\star,h''}_{g:h''}(x_{h''},a_{h''})m^t_{x_ga_g,h''+1:h'}(x_{h'},a_{h'})}{\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'})}\mathbf{1}\{(x_{h'},a_{h'})\succeq x_g\}$$

$$\overset{(i)}{\le} \frac{2\eta^2}{\gamma^2} \sum_{g\le h\le h'\le H} \sum_{h''=g}^{h} \sum_{x_{h'},a_{h'}}$$

$$\frac{\mu^t_{1:g-1}(x_{g-1},a_{g-1})\mu^{\star,h''}_{g:h''}(x_{h''},a_{h''})m^t_{x_ga_g,h''+1:h'}(x_{h'},a_{h'})\mathbf{1}\{(x_{h'},a_{h'})\succeq x_g\}}{\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'})}$$

$$\overset{(ii)}{\le} \frac{2\eta^2}{\gamma^2} \sum_{g\le h\le h'\le H} \sum_{h''=g}^{h} X_{h'}A$$

$$\le \frac{2\eta^2}{\gamma^2} H^2 XA.$$

Here, (i) uses that

$$\mu^t_{x_ga_g} = \mu^t_{1:g}(x_g,a_g) \le \mu^t_{1:g-1}(x_{g-1},a_{g-1}),$$

and (ii) uses the property of the balanced policy as in Lemma E.1, and observing that

$$\mu^t_{1:g-1}(x_{g-1},a_{g-1})\mu^{\star,h''}_{g:h''}(x_{h''},a_{h''})m^t_{x_ga_g,h''+1:h'}(x_{h'},a_{h'})\mathbf{1}\{(x_{h'},a_{h'})\succeq x_g\}$$

is bounded by some sequence form policy over steps $1:h'$.

Taking supremum over x_ga_g, we get

$$\frac{1}{XA}\sup_{g,x_ga_g}\Delta^t_{x_ga_g} \le \frac{2\eta^2}{\gamma^2}H^2.$$

This proves the lemma. □

F.2.5 Proof of Lemma F.7

Proof of Lemma F.7. We first upper bound I_t:

$$\begin{aligned}
\mathrm{I}_t \le& 8\eta \sum_{g\le h\le h'\le H} \sum_{h''=g}^{h} \sum_{x_{h'},a_{h'}} \sum_{x_ga_g} \lambda^t_{x_ga_g}\mu^{\star,h''}_{g:h''}(x_{h''},a_{h''})m^t_{x_ga_g,h''+1:h'}(x_{h'},a_{h'})m^t_{x_ga_g,g:h}(x_h,a_h)(\mu^t_{x_ga_g})^2 \\
&\times \frac{\mathbf{1}\{x^t_h,a^t_h=x_h,a_h\}}{(\mu^t_{1:h}(x_h,a_h)+\gamma(\mu^{\star,h}_{1:h}(x_h,a_h)+\mu^t_{x_ga_g}m^t_{x_ga_g,g:h}(x_h,a_h)\mathbf{1}\{x_h\succeq x_g\}))} \\
&\times \frac{\mathbf{1}\{x^t_{h'},a^t_{h'}=x_{h'},a_{h'}\}}{(\mu^t_{1:h'}(x_{h'},a_{h'})+\gamma(\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'})+\mu^t_{x_ga_g}m^t_{x_ga_g,g:h'}(x_{h'},a_{h'})\mathbf{1}\{x_{h'}\succeq x_g\}))} \\
&\times \mathbf{1}\{(x_{h'},a_{h'})\succeq(x_h,a_h)\succeq x_g\} \\
\overset{(i)}{\le}& 8\eta \sum_{g\le h\le h'\le H} \sum_{h''=g}^{h} \sum_{x_{h'},a_{h'}} \sum_{x_ga_g} \mu^{\star,h''}_{g:h''}(x_{h''},a_{h''})m^t_{x_ga_g,h''+1:h'}(x_{h'},a_{h'})\mu^t_{x_ga_g}\mu^t_{1:h}(x_h,a_h) \\
&\times \frac{\mathbf{1}\{x^t_h,a^t_h=x_h,a_h\}}{(\mu^t_{1:h}(x_h,a_h)+\gamma(\mu^{\star,h}_{1:h}(x_h,a_h)+\mu^t_{x_ga_g}m^t_{x_ga_g,g:h}(x_h,a_h)\mathbf{1}\{x_h\succeq x_g\}))} \\
&\times \frac{\mathbf{1}\{x^t_{h'},a^t_{h'}=x_{h'},a_{h'}\}}{(\mu^t_{1:h'}(x_{h'},a_{h'})+\gamma(\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'})+\mu^t_{x_ga_g}m^t_{x_ga_g,g:h'}(x_{h'},a_{h'})\mathbf{1}\{x_{h'}\succeq x_g\}))} \\
&\times \mathbf{1}\{(x_{h'},a_{h'})\succeq(x_h,a_h)\succeq x_g\} \\
\le& 8\eta \sum_{g\le h\le h'\le H} \sum_{h''=g}^{h} \sum_{x_{h'},a_{h'}} \sum_{x_ga_g} \mu^{\star,h''}_{g:h''}(x_{h''},a_{h''})m^t_{x_ga_g,h''+1:h'}(x_{h'},a_{h'})\mu^t_{x_ga_g} \\
&\times \frac{\mathbf{1}\{x^t_{h'},a^t_{h'}=x_{h'},a_{h'}\}\times\mathbf{1}\{(x_{h'},a_{h'})\succeq(x_h,a_h)\succeq x_g\}}{(\mu^t_{1:h'}(x_{h'},a_{h'})+\gamma(\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'})+\mu^t_{x_ga_g}m^t_{x_ga_g,g:h'}(x_{h'},a_{h'})\mathbf{1}\{x_{h'}\succeq x_g\}))}
\end{aligned}$$

$$\le 8\eta H \sum_{g\le h'\le H}\sum_{h''=g}^{h'}\sum_{x_{h'},a_{h'}}\sum_{x_g a_g}\mu^{\star,h''}_{g:h''}(x_{h''},a_{h''})m^t_{x_g a_g,h''+1:h'}(x_{h'},a_{h'})\mu^t_{x_g a_g}$$
$$\times\frac{\mathbf{1}\{x^t_{h'},a^t_{h'}=x_{h'},a_{h'}\}\times\mathbf{1}\{x_{h'}\succeq x_g\}}{\mu^t_{1:h'}(x_{h'},a_{h'})+\gamma(\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'})+\mu^t_{x_g a_g}m^t_{x_g a_g,g:h'}(x_{h'},a_{h'}))}$$
$$\overset{(ii)}{=}8\eta H\sum_{g\le h'\le H}\sum_{h''=g}^{h'}\widetilde{\Delta}^t_{g,h',h''}.$$

Here, (i) used the fact that $\lambda^t_{x_g a_g}m^t_{x_g a_g,g:h}(x_h,a_h)\mu^t_{x_g a_g}\le\mu^t_{1:h}(x_h,a_h)$ as shown in Eq. (49). Moreover, in (ii), we define for any $g\le h''\le h'$:

$$\widetilde{\Delta}^t_{g,h',h''}:=\sum_{x_{h'},a_{h'}}\sum_{x_g a_g}\mu^{\star,h''}_{g:h''}(x_{h''},a_{h''})m^t_{x_g a_g,h''+1:h'}(x_{h'},a_{h'})\mu^t_{x_g a_g}$$
$$\times\frac{\mathbf{1}\{x^t_{h'},a^t_{h'}=x_{h'},a_{h'}\}\times\mathbf{1}\{x_{h'}\succeq x_g\}}{\mu^t_{1:h'}(x_{h'},a_{h'})+\gamma(\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'})+\mu^t_{x_g a_g}m^t_{x_g a_g,g:h'}(x_{h'},a_{h'}))}.$$

Observe that the random variable $\widetilde{\Delta}^t_{g,h',h''}$ satisfies the following properties:

- $\widetilde{\Delta}^t_{g,h',h''}\le X_{h'}A/\gamma$ almost surely: We have

  $$\widetilde{\Delta}^t_{g,h',h''}\le\frac{1}{\gamma}\sum_{x_{h'},a_{h'}}\frac{\sum_{x_g a_g}\mu^{\star,h''}_{g:h''}(x_{h''},a_{h''})m^t_{x_g a_g,h''+1:h'}(x_{h'},a_{h'})\mu^t_{x_g a_g}\mathbf{1}\{x_{h'}\succeq x_g\}}{\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'})}.$$

  Notice that (for this fixed g,h'')

  $$\sum_{x_g a_g}\mu^{\star,h''}_{g:h''}(x_{h''},a_{h''})m^t_{x_g a_g,h''+1:h'}(x_{h'},a_{h'})\mu^t_{x_g a_g}\mathbf{1}\{x_{h'}\succeq x_g\}\tag{65}$$

  is the sequence-form of a certain policy at $(x_{h'},a_{h'})$, where the policy is defined as follows: First, take policy $\mu^t_{1:g}$ and arrive at some $x_g\in\mathcal{X}_g$. Let a_g be the action sampled from $\mu^t_g(\cdot|x_g)$. Then, starting from x_g, discard a_g and instead take policy $\mu^{\star,h''}_{g:h''}m^t_{x_g a_g,h''+1:H}$ until the end of the game. One may check that the sequence-form of this policy is indeed given by (65). Therefore, we have $\widetilde{\Delta}^t_{g,h,h''}\le X_{h'}A/\gamma$ by the balancing property of $\mu^{\star,h'}_{1:h'}$ (Lemma E.1).

- $\mathbb{E}[\widetilde{\Delta}^t_{g,h',h''}|\mathcal{F}_{t-1}]\le 1$: We have

  $$\begin{aligned}&\mathbb{E}[\widetilde{\Delta}^t_{g,h',h''}|\mathcal{F}_{t-1}]\\ &\le\sum_{x_{h'},a_{h'}}\sum_{x_g a_g}\mu^{\star,h''}_{g:h''}(x_{h''},a_{h''})m^t_{x_g a_g,h''+1:h'}(x_{h'},a_{h'})\mu^t_{x_g a_g}p^t(x_{h'})\mathbf{1}\{x_{h'}\succeq x_g\}=1.\end{aligned}$$

  Above, the last equality used again the fact that (65) is the sequence-form of a policy.

- $\mathbb{E}[(\widetilde{\Delta}^t_{g,h',h''})^2|\mathcal{F}_{t-1}]\le X_{h'}A/\gamma$: Note that $\widetilde{\Delta}^t_{g,h',h''}$ is non-negative, so by the almost sure bound that $\widetilde{\Delta}^t_{g,h',h''}\le X_{h'}A/\gamma$, we have

  $$\mathbb{E}[(\widetilde{\Delta}^t_{g,h',h''})^2|\mathcal{F}_{t-1}]\le\mathbb{E}[\widetilde{\Delta}^t_{g,h',h''}|\mathcal{F}_{t-1}]\cdot X_{h'}A/\gamma\le X_{h'}A/\gamma.$$

By Freedman's inequality (Lemma A.2) and taking the union bound, with probability at least $1-\delta/(10H^3)$ and some fixed $\lambda\le\gamma/(X_{h'}A)$, we get

$$\sum_{t=1}^{T}\widetilde{\Delta}^t_{g,h',h''}\le T+\frac{\lambda X_{h'}AT}{\gamma}+\frac{4\log(10H/\delta)}{\lambda}.$$

Taking $\lambda = \gamma/(X_{h'}A)$, we have

$$\sum_{t=1}^{T} \widetilde{\Delta}_{g,h',h''}^{t} \le 2T + \frac{4X_{h'}A\log(10H/\delta)}{\gamma}.$$

Finally summing up $\widetilde{\Delta}_{g,h',h''}^{t}$ over g, h, h'' and taking the union bound, we have with probability at least $1-\delta/10$, we have

$$\sum_{t=1}^{T} \mathrm{I}_t \le 16\eta H^4 T + \frac{32\eta H^3 XA\iota}{\gamma},$$

where $\iota := \log(10H/\delta)$. This proves the lemma. $\square$

F.2.6 Proof of Lemma F.8

Proof. First, recall that the matrix loss estimator is defined as $\widetilde{M}^t = \sum_{g,x_ga_g} \mu_{g,x_ga_g}^t \widetilde{\ell}^{t,x_ga_g} e_{x_ga_g}^\top$, and the vector loss estimator is computed by

$$\widetilde{\ell}_h^{t,x_ga_g}(x_h,a_h) = \frac{\mathbf{1}\{(x_h^t,a_h^t)=(x_h,a_h)\}(1-r_h^t)}{\mu_{1:h}^t(x_h,a_h) + \gamma(\mu_{1:h}^{\star,h}(x_h,a_h) + \mu_{x_ga_g}^t m_{x_ga_g,g:h}^t(x_h,a_h)\mathbf{1}\{x_h \succeq x_g\})}.$$

We define a vector $\widetilde{\ell}^t = \{\widetilde{\ell}_h^t(x_h,a_h)\}_{(x_h,a_h)\in\mathcal{X}\times\mathcal{A}} \in \mathbb{R}_{\ge 0}^{XA}$ as

$$\widetilde{\ell}_h^t(x_h,a_h) := \frac{\mathbf{1}\{(x_h^t,a_h^t)=(x_h,a_h)\}(1-r_h^t)}{\mu_{1:h}^t(x_h,a_h) + \gamma\mu_{1:h}^{\star,h}(x_h,a_h)}. \tag{66}$$

Then we have for any (t, x_ga_g), (x_h,a_h) that

$$\widetilde{\ell}_h^{t,x_ga_g}(x_h,a_h) \le \widetilde{\ell}_h^t(x_h,a_h).$$

Then $\langle I - E_{\succeq x_ga_g} + m_{x_ga_g}^t e_{x_ga_g}^\top, \widetilde{M}^t\rangle$ can be upper bounded as follows:

$$\begin{aligned}
&\langle I - E_{\succeq x_ga_g} + m_{x_ga_g}^t e_{x_ga_g}^\top, \widetilde{M}^t\rangle \\
&= \langle I - E_{\succeq x_ga_g} + m_{x_ga_g}^t e_{x_ga_g}^\top, \sum_{h,x_ha_h} \mu_{x_ha_h}^t \widetilde{\ell}^{t,x_ga_g} e_{x_ha_h}^\top\rangle \\
&\le \langle I - E_{\succeq x_ga_g} + m_{x_ga_g}^t e_{x_ga_g}^\top, \widetilde{\ell}^t \sum_{h,x_h,a_h} \mu_{x_ha_h}^t e_{x_ha_h}^\top\rangle \\
&= \langle \phi_{x_ga_g\to m_{x_ga_g}^t}, \widetilde{\ell}^t(\mu^t)^\top\rangle = \langle \phi_{x_ga_g\to m_{x_ga_g}^t}\mu^t, \widetilde{\ell}^t\rangle \\
&= \sum_{h=1}^{H}\sum_{h,x_ha_h} (\phi_{x_ga_g\to m_{x_ga_g}^t}\mu^t)_{1:h}(x_h,a_h)\widetilde{\ell}_h^t(x_h,a_h),
\end{aligned}$$

where we have used $\phi_{x_ga_g\to m_{x_ga_g}^t} := I - E_{\succeq x_ga_g} + m_{x_ga_g}^t e_{x_ga_g}^\top$ to denote the EFCE modification triggered at (x_g,a_g) and then playing the policy $m_{x_ga_g}^t$. Also, $\langle I - E_{\succeq x_ga_g} + m_{x_ga_g}^t e_{x_ga_g}^\top, \widetilde{M}^t\rangle \ge 0$ as both matrices have non-negative entries. As a result, we get that

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{g,x_ga_g} \lambda_{x_ga_g}^t \langle I - E_{\succeq x_ga_g} + m_{x_ga_g}^t e_{x_ga_g}^\top, \widetilde{M}^t\rangle^2 \\
&\le \sum_{t=1}^{T}\sum_{g,x_ga_g} \lambda_{x_ga_g}^t \left(\langle \phi_{x_ga_g\to m_{x_ga_g}^t}\mu^t, \widetilde{\ell}^t\rangle\right)^2 \\
&\le 2\sum_{t=1}^{T}\sum_{g,x_ga_g} \lambda_{x_ga_g}^t \sum_{1\le h\le h'\le H}\sum_{x_h,a_h}\sum_{(x_{h'},a_{h'})\in\mathcal{C}_{h'}(x_h,a_h)}
\end{aligned}$$

$$
\begin{aligned}
&\frac{(\phi_{x_ga_g\to m^t_{x_ga_g}}\mu^t)_{1:h}(x_h,a_h)\mathbf{1}\left\{(x^t_{h'},a^t_{h'})=(x_{h'},a_{h'})\right\}\cdot(1-r^t_h)\cdot(1-r^t_{h'})}{(\mu^t_{1:h}(x_h,a_h)+\gamma\mu^{\star,h}_{1:h}(x_h,a_h))(\mu^t_{1:h'}(x_{h'},a_{h'})+\gamma\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'}))}\\
&\le 2\sum_{1\le h\le h'\le H}\sum_{t=1}^T\sum_{x_h,a_h}\sum_{(x_{h'},a_{h'})\in\mathcal{C}_{h'}(x_h,a_h)}\sum_{g,x_ga_g}\frac{\lambda^t_{x_ga_g}(\phi_{x_ga_g\to m^t_{x_ga_g}}\mu^t)_{1:h}(x_h,a_h)}{\mu^t_{1:h}(x_h,a_h)}\widetilde{\ell}^t_{h'}(x_{h'},a_{h'})\\
&\overset{(i)}{=} 2\sum_{1\le h\le h'\le H}\sum_{t=1}^T\sum_{x_h,a_h}\sum_{(x_{h'},a_{h'})\in\mathcal{C}_{h'}(x_h,a_h)}\widetilde{\ell}^t_{h'}(x_{h'},a_{h'})\\
&\le 2H\sum_{t=1}^T\sum_{h',x_{h'},a_{h'}}\widetilde{\ell}^t_{h'}(x_{h'},a_{h'})\\
&\overset{(ii)}{\le} 2H\sum_{t=1}^T\sum_{h',x_{h'},a_{h'}}\underbrace{\ell^t_{h'}(x_{h'},a_{h'})}_{\le 1}+2H\sum_{h',x_{h'},a_{h'}}\frac{\log(10XA/\delta)}{\gamma\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'})}\\
&\overset{(iii)}{\le} 2HXAT+2H\sum_{h',x_{h'},a_{h'}}\iota\cdot X_{h'}A/\gamma\\
&\le 2HXAT+2HX^2A^2\iota/\gamma.
\end{aligned}
$$

Above, (i) uses that μ^t is the solution of the fixed point equation $\mu=\sum_{g,x_ga_g}\lambda^t_{x_g,a_g}(I-E_{\succeq x_ga_g}+m^t_{x_ga_g}e^\top_{x_ga_g})\mu$; (ii) is by [6, Corollary D.6] for each $(h',x_{h'},a_{h'})$ with probability $1-\delta/(10XA)$ and a union bound; (iii) uses $\mu^{\star,h'}_{1:h'}(x_{h'},a_{h'})\ge 1/(X_{h'}A)$ by Corollary E.1. Therefore, we have with probability at least $1-\delta/10$ that

$$
\begin{aligned}
\sum_{t=1}^T \mathrm{II}_t &= \frac{e\eta}{XA}\cdot\sum_{t=1}^T\sum_{g,x_ga_g}\lambda^t_{x_ga_g}\langle I-E_{\succeq x_ga_g}+m^t_{x_ga_g}e^\top_{x_ga_g},\widetilde{M}^t\rangle^2\\
&\le\frac{e\eta}{XA}\big(2HXAT+2HX^2A^2\iota/\gamma\big)\le 6\eta HT+6\eta HXA\iota/\gamma.
\end{aligned}
$$

This proves the lemma. □

F.3 Bound on two bias terms

Proof of Lemma F.2. First, recall that the matrix loss estimator gives $\widetilde{M}^t=\sum_{g,x_g,a_g}\mu^t_{x_ga_g}\widetilde{\ell}^{t,x_ga_g}e^\top_{x_ga_g}$ and the vector loss estimator is computed by

$$
\widetilde{\ell}^{t,x_ga_g}_h(x_h,a_h)=\frac{\mathbf{1}\left\{(x^t_h,a^t_h)=(x_h,a_h)\right\}(1-r^t_h)}{\mu^t_{1:h}(x_h,a_h)+\gamma(\mu^{\star,h}_{1:h}(x_h,a_h)+\mu^t_{x_ga_g}m^t_{x_ga_g,g:h}(x_h,a_h)\mathbf{1}\left\{x_h\succeq x_g\right\})}.
$$

Then we decompose BIAS_1 as

$$
\begin{aligned}
\mathrm{BIAS}_1 &= \sum_{t=1}^T\langle\phi^t,\ell^t(\mu^t)^\top-\widetilde{M}^t\rangle\\
&=\underbrace{\sum_{t=1}^T\langle\phi^t,\ell^t(\mu^t)^\top-\mathbb{E}\Big[\widetilde{M}^t|\mathcal{F}_{t-1}\Big]\rangle}_{(A)}+\underbrace{\sum_{t=1}^T\langle\phi^t,\mathbb{E}\Big[\widetilde{M}^t|\mathcal{F}_{t-1}\Big]-\widetilde{M}^t\rangle}_{(B)}.
\end{aligned}
$$

We first the second term (B) by Azuma-Hoeffding inequality. Recall the definition of $\widetilde{\ell}^t$ in (66). We immediately have $\widetilde{\ell}^{t,x_ga_g}\le\widetilde{\ell}^t$ pointwisely, so we can upper bound $\langle\phi^t,\widetilde{M}^t\rangle$ by

$$
\langle\phi^t,\widetilde{M}^t\rangle\le\langle\phi^t,\sum_{g,x_g,a_g}\mu^t_{x_ga_g}\widetilde{\ell}^te^\top_{x_ga_g}\rangle=\langle\phi^t\mu^t,\widetilde{\ell}^t\rangle=\langle\mu^t,\widetilde{\ell}^t\rangle,
$$

where the last equality comes from fixed point equation $\mu^t = \phi^t \mu^t$. Then we have

$$
\begin{aligned}
&\langle \phi^t, \widetilde{M}^t \rangle \le \langle \mu^t, \widetilde{\ell}^t \rangle \\
=& \sum_{h=1}^{H} \sum_{x_h, a_h} \mu_{1:h}^t(x_h, a_h) \frac{\mathbf{1}\left\{(x_h^t, a_h^t) = (x_h, a_h)\right\} \cdot (1 - r_h^t)}{\mu_{1:h}^t(x_h, a_h) + \gamma \mu_{1:h}^{\star,h}(x_h, a_h)} \\
\le& \sum_{h=1}^{H} \sum_{x_h, a_h} \mathbf{1}\left\{x_h = x_h^t, a_h = a_h^t\right\} = \sum_{h=1}^{H} 1 = H.
\end{aligned}
$$

As a consequence, by Azuma-Hoeffding inequality, with probability at least $1 - \delta/10$, we have

$$
\sum_{t=1}^{T} \langle \phi^t, \mathbb{E}\left[\widetilde{M}^t | \mathcal{F}_{t-1}\right] - \widetilde{M}^t \rangle \le H\sqrt{2T\log(10/\delta)} \le H\sqrt{2T\iota}.
$$

Then we turn to bound the first term (A). Denote $\ell^{t, x_g a_g} = \mathbb{E}\left[\widetilde{\ell}^{t, x_g a_g} | \mathcal{F}_{t-1}\right]$ and plug in the definition of $\widetilde{\ell}^{t, x_g a_g}$, we get

$$
\begin{aligned}
&\langle \phi^t, \ell^t (\mu^t)^\top - \mathbb{E}\left[\widetilde{M}^t | \mathcal{F}_{t-1}\right] \rangle \\
=& \langle \phi^t, \ell^t (\mu^t)^\top \rangle - \sum_{g, x_g, a_g} \langle \phi^t, \mu_{x_g a_g}^t \ell^{t, x_g a_g} e_{x_g a_g}^\top \rangle \\
=& \sum_{g, x_g, a_g} \langle \phi^t e_{x_g a_g} \mu_{x_g a_g}^t, \ell^t - \ell^{t, x_g a_g} \rangle.
\end{aligned}
$$

Note that by the definition of the loss estimator as in Eq. (4), we have

$$
\ell_h^t(x_h, a_h) = p^t(x_h)[1 - \overline{R}_h^t(x_h, a_h)] \le p^t(x_h),
$$

where we recall the definition of $p^t(x_h)$ in (50).

Moreover, the $\ell^{t, x_g a_g}(x_h, a_h)$ is related to $\ell_h^t(x_h, a_h)$ by a rescaling

$$
\ell^{t, x_g a_g}(x_h, a_h) = \frac{\mu_{1:h}^t(x_h, a_h) \ell_h^t(x_h, a_h)}{\mu_{1:h}^t(x_h, a_h) + \gamma(\mu_{1:h}^{\star,h}(x_h, a_h) + \mu_{x_g a_g}^t m_{x_g a_g, g:h}^t(x_h, a_h) \mathbf{1}\left\{x_h \succeq x_g\right\})}.
$$

So we get

$$
\begin{aligned}
&\langle \phi^t, \ell^t (\mu^t)^\top - \mathbb{E}\left[\widetilde{M}^t | \mathcal{F}_{t-1}\right] \rangle \\
=& \sum_{g, x_g, a_g} \sum_{h, x_h, a_h} \mu_{x_g a_g}^t (\phi^t e_{x_g a_g})_{1:h}(x_h, a_h) \\
&\times \frac{\gamma(\mu_{1:h}^{\star,h}(x_h, a_h) + \mu_{x_g a_g}^t m_{x_g a_g, g:h}^t(x_h, a_h) \mathbf{1}\left\{x_h \succeq x_g\right\}) \ell_h^t(x_h, a_h)}{\mu_{1:h}^t(x_h, a_h) + \gamma(\mu_{1:h}^{\star,h}(x_h, a_h) + \mu_{x_g a_g}^t m_{x_g a_g, g:h}^t(x_h, a_h) \mathbf{1}\left\{x_h \succeq x_g\right\})} \\
\le& \gamma \sum_{g, x_g, a_g} \sum_{h, x_h, a_h} \frac{\mu_{x_g a_g}^t (\phi^t e_{x_g a_g})_{1:h}(x_h, a_h) \mu_{1:h}^{\star,h}(x_h, a_h) p^t(x_h)}{\mu_{1:h}^t(x_h, a_h) + \gamma(\mu_{1:h}^{\star,h}(x_h, a_h) + \mu_{x_g a_g}^t m_{x_g a_g, g:h}^t(x_h, a_h) \mathbf{1}\left\{x_h \succeq x_g\right\})} \\
&+ \gamma \sum_{g, x_g, a_g} \sum_{h, x_h, a_h} \frac{\mu_{x_g a_g}^t (\phi^t e_{x_g a_g})_{1:h}(x_h, a_h) \mu_{x_g a_g}^t m_{x_g a_g, g:h}^t(x_h, a_h) \mathbf{1}\left\{x_h \succeq x_g\right\} p^t(x_h)}{\mu_{1:h}^t(x_h, a_h) + \gamma(\mu_{1:h}^{\star,h}(x_h, a_h) + \mu_{x_g a_g}^t m_{x_g a_g, g:h}^t(x_h, a_h) \mathbf{1}\left\{x_h \succeq x_g\right\})}.
\end{aligned}
$$

The first term admits an upper bound

$$
\begin{aligned}
&\gamma \sum_{h, x_h, a_h} \sum_{g, x_g, a_g} \frac{\mu_{x_g a_g}^t (\phi^t e_{x_g a_g})_{1:h}(x_h, a_h)}{\mu_{1:h}^t(x_h, a_h)} \mu_{1:h}^{\star,h}(x_h, a_h) p^t(x_h) \\
\overset{(i)}{=}& \gamma \sum_{h, x_h, a_h} \mu_{1:h}^{\star,h}(x_h, a_h) p^t(x_h) \overset{(ii)}{=} \gamma H.
\end{aligned}
$$

Here, (i) uses $\mu^t = \phi^t \mu^t$ and (ii) uses Eq. (51).

The second term can be upper bounded by

$$
\begin{aligned}
&\gamma \sum_{h,x_h,a_h} \sum_{g,x_g,a_g,x_h \succeq x_g} \frac{\mu^t_{x_g a_g} (\phi^t e_{x_g a_g})_{1:h}(x_h, a_h) \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h) p^t(x_h)}{\mu^t_{1:h}(x_h, a_h)} \\
\le\ & \gamma \sum_{h,x_h,a_h} \left(\sum_{g,x_g,a_g,x_h \succeq x_g} \frac{\mu^t_{x_g a_g} (\phi^t e_{x_g a_g})_{1:h}(x_h, a_h)}{\mu^t_{1:h}(x_h, a_h)} \right) \cdot \left(\sum_{g,x_g,a_g,x_h \succeq x_g} \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h) p^t(x_h) \right) \\
\overset{(i)}{\le}\ & \gamma \sum_{h,x_h,a_h} \sum_{g,x_g,a_g,x_h \succeq x_g} \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h) p^t(x_h) \\
=\ & \gamma \sum_{h,x_h,a_h} \sum_{g=1}^{h} \sum_{x_g a_g} \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h) \mathbf{1}\left\{x_h \succeq x_g\right\} \cdot p^t(x_h) \\
=\ & \gamma \sum_{1 \le g \le h \le H} \sum_{x_h,a_h} \sum_{x_g a_g} \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h) \mathbf{1}\left\{x_h \succeq x_g\right\} \cdot p^t(x_h) \\
\overset{(ii)}{\le}\ & \gamma H^2.
\end{aligned}
$$

Here, the inequality in (i) also uses $\mu^t = \phi^t \mu^t$; (ii) used the fact for any fixed g, $\sum_{x_g a_g} \mu^t_{x_g a_g} m^t_{x_g a_g, g:h}(x_h, a_h) \mathbf{1}\left\{x_h \succeq x_g\right\}$ is the sequence-form of a policy, similar as (65).

Taking summation over $t = 1, 2, \cdots, T$, we have

$$
(A) \le 2\gamma H^2 T.
$$

Combined with the bound on (B), we have with probability at least $1 - \delta/10$ that

$$
\mathrm{BIAS}_1 \le 2\gamma H^2 T + 2H\sqrt{T\iota}.
$$

This completes the proof of this lemma. □

Proof of Lemma F.3. Recall the definition of $\widetilde{\ell}^t$ in (66). We have $\widetilde{\ell}^{t,x_g a_g} \le \widetilde{\ell}^t$ pointwisely, so we have

$$
\begin{aligned}
\langle \phi^\star, \widetilde{M}^t - \ell^t (\mu^t)^\top \rangle &= \langle \phi^\star, \sum_{g,x_g a_g} \widetilde{\ell}^{t,x_g a_g} \mu^t_{x_g a_g} e^\top_{x_g a_g} - \ell^t (\mu^t)^\top \rangle \\
&\le \langle \phi^\star, \widetilde{\ell}^t (\mu^t)^\top - \ell^t (\mu^t)^\top \rangle = \langle \phi^\star \mu^t, \widetilde{\ell}^t - \ell^t \rangle.
\end{aligned}
$$

Then we can get that with probability at least $1 - \delta/3$

$$
\begin{aligned}
\mathrm{BIAS}_2 &\le \max_{\phi^\star \in \Phi^{\mathsf{Tr}}} \sum_{t=1}^{T} \left\langle \phi^\star \mu^t, \widetilde{\ell}^t - \ell^t \right\rangle \\
&= \max_{\phi^\star \in \Phi^{\mathsf{Tr}}} \sum_{t=1}^{T} \sum_{h=1}^{H} \sum_{x_h,a_h} (\phi^\star \mu^t)_{1:h}(x_h, a_h) \left[\widetilde{\ell}^t_h(x_h, a_h) - \ell^t_h(x_h, a_h) \right] \\
&= \max_{\phi^\star \in \Phi^{\mathsf{Tr}}} \sum_{t=1}^{T} \sum_{h=1}^{H} \sum_{x_h,a_h} \frac{(\phi^\star \mu^t)_{1:h}(x_h, a_h)}{\gamma \mu^{\star,h}_{1:h}(x_h, a_h)} \gamma \mu^{\star,h}_{1:h}(x_h, a_h) \left[\widetilde{\ell}^t_h(x_h, a_h) - \ell^t_h(x_h, a_h) \right] \\
&= \max_{\phi^\star \in \Phi^{\mathsf{Tr}}} \sum_{h=1}^{H} \sum_{x_h,a_h} \frac{(\phi^\star \mu^t)_{1:h}(x_h, a_h)}{\gamma \mu^{\star,h}_{1:h}(x_h, a_h)} \sum_{t=1}^{T} \gamma \mu^{\star,h}_{1:h}(x_h, a_h) \left[\widetilde{\ell}^t_h(x_h, a_h) - \ell^t_h(x_h, a_h) \right] \\
&\overset{(i)}{\le} \frac{\log(3XA/\delta)}{\gamma} \max_{\phi^\star \in \Phi^{\mathsf{Tr}}} \sum_{h=1}^{H} \sum_{x_h,a_h} \frac{(\phi^\star \mu^t)_{1:h}(x_h, a_h)}{\mu^{\star,h}_{1:h}(x_h, a_h)} \\
&\overset{(ii)}{=} \frac{\iota}{\gamma} \sum_{h=1}^{H} X_h A = XA\iota/\gamma,
\end{aligned}
$$

where (i) is a high probability bound by applying Corollary D.6 in Bai et al. [6] for each (x_h, a_h) and taking union bound, and (ii) is by the balancing property of $\mu^{\star,h}$. This proves the lemma. □

Algorithm 6 OMD (FTRL form)

Require: Learning rate $\eta > 0$.

for $t = 1, 2, \dots, T$ **do**

Compute $\mu_h^t(a_h|x_h)$ and $F_{x_h}^t$ in the bottom-up order over $x_h \in \mathcal{X}$:

$$\mu_h^t(a_h|x_h) \propto_{a_h} \exp\Big\{ -\eta \sum_{s=1}^{t-1} \ell_h^s(x_h, a_h) + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F_{x_{h+1}}^t \Big\}, \tag{67}$$

$$F_{x_h}^t = \log \sum_{a_h} \exp\Big\{ -\eta \sum_{s=1}^{t-1} \ell_h^s(x_h, a_h) + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F_{x_{h+1}}^t \Big\}. \tag{68}$$

Receive loss $\ell^t = \{\ell_h^t(x_h, a_h)\}_{(x_h, a_h) \in \mathcal{X} \times \mathcal{A}} \in \mathbb{R}_{\geq 0}^{XA}$.

G Equivalence between Vertex MWU and OMD

G.1 Proof of Theorem 8

In this section we prove Theorem 8. Our proof is based on Algorithm 6, which is just (the efficient implementation of) the standard OMD algorithm with dilated entropy regularizer in FTRL form [35]. Indeed, Lemma G.3 show that its output policy $\{\mu^t\}_{t \geq 1}$ is the same as (26). Then, Lemma G.1 & G.2 show that its output policy $\{\mu^t\}_{t \geq 1}$ is the same as (25). These together imply the equivalence of (26) and (25), thereby proving Theorem 8.

The rest of this subsection is devoted to stating and proving Lemma G.1-G.3.

Remark on optimistic algorithms As pointed out in [25], Theorem 8 does not depend on the concrete values of $\{\ell^t\}_{t \geq 1}$. As a result, the equivalence also holds for the optimistic version of the algorithms (where the algorithms are fed with loss functions $\{2\ell^t - \ell^{t-1}\}_{t \geq 1}$, with $\ell^0 := 0$) which achieves an faster $\mathcal{O}(\mathrm{poly}(\log T))$ regret. In words: The Kernelized OMWU algorithm of Farina et al. [25] is equivalent to an Optimistic OMD algorithm with the dilated KL distance.

Lemma G.1 (Conversion to log-partition function). *Define the log-partition function $F^{\mathcal{V}} : \mathbb{R}^{XA} \to \mathbb{R}$*

$$F^{\mathcal{V}}(\ell) := \log \sum_{v \in \mathcal{V}} \exp\{-\langle v, \ell \rangle\}. \tag{69}$$

Then update (25) has a closed-form update for all $t \geq 1$:

$$\mu^t = -\nabla F^{\mathcal{V}}\Big(\eta \sum_{s=1}^{t-1} \ell^s\Big) = -\frac{\sum_{v \in \mathcal{V}} \exp\big\{-\eta \big\langle v, \sum_{s=1}^{t-1} \ell^s \big\rangle\big\} v}{\sum_{v \in \mathcal{V}} \exp\big\{-\eta \big\langle v, \sum_{s=1}^{t-1} \ell^s \big\rangle\big\}}. \tag{70}$$

Proof. By (25),

$$\mu^t = \sum_v p_v^t v = \frac{\sum_\phi \exp\{-\eta \left\langle v, \sum_{s=1}^{t-1} \ell^s \right\rangle\} v}{\sum_v \exp\{-\eta \left\langle v, \sum_{s=1}^{t-1} \ell^s \right\rangle\}} = -\nabla F^{\mathcal{V}}\left(\eta \sum_{s=1}^{t-1} \ell^s\right).$$

□

Lemma G.2 (Recursive expression of $F^{\mathcal{V}}$ and $\nabla F^{\mathcal{V}}$). *For any loss matrix $\ell \in \mathbb{R}^{XA}$, the* log-partition function *can be written as $F^{\mathcal{V}}(\ell) = F_\emptyset(\ell)$ where $F_{x_h}(\ell) := \log \sum_{v \in \mathcal{V}^{x_h}} \exp\{-\langle v, \ell \rangle\}$ can be computed recurrently by $F_{x_{H+1}}(\cdot) = 0$ and*

$$F_{x_h}(\ell) := \log \sum_{a_h} \exp\Big\{ -\ell_h(x_h, a_h) + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F_{x_{h+1}}(\ell) \Big\}. \tag{71}$$

Furthermore, define a (sequence form) policy μ by

$$\mu(a_h|x_h) \propto_{a_h} \exp\Big\{ -\ell_h(x_h, a_h) + \sum_{x_{h+1} \in \mathcal{C}(x_h, a_h)} F_{x_{h+1}}(\ell) \Big\}, \tag{72}$$

then we have

$$-\nabla F^{\mathcal{V}}(\ell) = \mu. \tag{73}$$

Proof. We first show (71). Using the structure of $\mathcal{V}^{x_h}$,

$$\begin{aligned}F_{x_h}(\ell) &= \log \sum_{v\in\mathcal{V}^{x_h}} \exp\{-\langle v,\ell\rangle\}\\ &= \log \sum_{a_h\in\mathcal{A}_{x_h}} \exp\Big\{-\ell_h(x_h,a_h) + \sum_{x_{h+1}\in\mathcal{C}(x_ha_h)}\sum_{v\in\mathcal{V}^{x_{h+1}}} \exp\{-\langle v,\ell\rangle\}\Big\}.\\ &= \log \sum_{a_h\in\mathcal{A}_{x_h}} \exp\Big\{-\ell_h(x_h,a_h) + \sum_{x_{h+1}\in\mathcal{C}(x_ha_h)} F_{x_{h+1}}(\ell)\Big\}.\end{aligned}$$

Next we show (73). Taking the gradient,

$$\begin{aligned}&-\nabla F_{x_h}(\ell)\\ =& \frac{\sum_{a_h\in\mathcal{A}_{x_h}} \exp\Big\{-\ell_h(x_h,a_h)+\sum_{x_{h+1}\in\mathcal{C}(x_ha_h)}F_{x_{h+1}}(\ell)\Big\}[e_{x_ha_h}-\sum_{x_{h+1}\in\mathcal{C}(x_ha_h)}\nabla F_{x_{h+1}}(\ell)]}{\sum_{a_h\in\mathcal{A}_{x_h}} \exp\Big\{-\ell_h(x_h,a_h)+\sum_{x_{h+1}\in\mathcal{C}(x_ha_h)}F_{x_{h+1}}(\ell)\Big\}}\\ =& \sum_{a_h\in\mathcal{A}_{x_h}} \mu_h(a_h|x_h)[e_{x_ha_h} + \sum_{x_{h+1}\in\mathcal{C}(x_ha_h)} (-\nabla F_{x_{h+1}})(\ell)].\end{aligned}$$

For any x_h, repeat the above process along the treeplex, the contribution of the production of $\mu(\cdot|\cdot)$ will be the sequence form. As a result, $-\nabla F^{\mathcal{V}}(\ell)=\mu$, which completes the proof. □

Lemma G.3. *The policy μ^t in Algorithm 6 is the optimizer of the optimization problem*

$$\arg\min_{\mu\in\Pi^{x_h}} \left[\eta\left\langle \mu, \sum_{s=1}^{t-1}\ell^s\right\rangle + H_{x_h}(\mu)\right]$$

for all x_h simultaneously. Furthermore,

$$\min_{\mu\in\Pi^{x_h}} \left[\eta\left\langle \mu, \sum_{s=1}^{t-1}\ell^s\right\rangle + H_{x_h}(\mu)\right] = -F_{x_h}\left(\eta\sum_{s=1}^{t-1}\ell^s\right).$$

The result is known in the literature (e.g. [35]) and its proof is similar to Appendix B of [34], which focuses on the special case when the loss function is the bandit-based loss estimator (19):

$$\ell_h^t(x_h,a_h) := \frac{\mathbf{1}\left\{(x_h^t,a_h^t)=(x_h,a_h)\right\}(1-r_h^t)}{\mu_{1:h}^t(x_h,a_h)+\gamma}.$$

For completeness, we include a proof for any generic loss vector $\ell\in\mathbb{R}^{XA}$; Lemma G.3 follows by taking $\ell=\eta\sum_{s=1}^{t-1}\ell^s$.

Proof. We prove by induction for $h=H,\cdots,1$. For $h=H$, since there is no further decisions to be make, this is just a linear optimization problem with entropy regularizer on simplex. As a result, $\mu_H(a_H|x_H)\propto_{a_H}\exp\{-\ell_H(x_H,a_H)\}$ as desired and the minimum is $-\log\sum_{a_H}\exp\Big\{-\ell_H(x_H,a_H)\Big\}=-F_{x_H}(\ell)$.

If the claim holds for levels after $h+1$, consider the h-th level. Plug in the optimizer after the $h+1$-th level, the optimization problem in the sub-tree rooted at x_h becomes

$$\arg\min_{\mu\in\Pi^{x_h}}\left[\sum_{a_h}\mu_h(a_h|x_h)\left(\ell_h(x_h,a_h)-\sum_{x_{h+1}\in\mathcal{C}(x_ha_h)}F_{x_{h+1}}(\ell)\right)+H(\mu_h(\cdot|x_h))\right],$$

which is again a linear optimization problem with entropy regularizer on simplex. As a result, $\mu_h(a_h|x_h)\propto_{a_h}\exp\{-\ell_h(x_h,a_h)+\sum_{x_{h+1}\in\mathcal{C}(x_ha_h)}F_{x_{h+1}}(\ell))\}$ as desired and the minimum is $-\log\sum_{a_h}\exp\Big\{-\ell_h(x_h,a_h)+\sum_{x_{h+1}\in\mathcal{C}(x_ha_h)}F_{x_{h+1}}(\ell))\Big\}=-F_{x_h}(\ell)$. □

G.2 Equivalence between OMD implementation and "Kernelized MWU" implementation of Farina et al. [25]

Farina et al. [25] design another efficient implementation of the Vertex MWU algorithm (25) via the "kernel trick", which they term as the Kernelized MWU algorithm (Algorithm 1 in [25][5]). As Theorem 8 shows that Vertex MWU is equivalent to standard OMD, the Kernelized MWU algorithm is also equivalent to standard OMD.

In this section, we further show that the implementation in Kernelized MWU is also "equivalent" to the standard linear-time implementation of OMD (Algorithm 6), by showing that the key intermediate quantities in both implementations are also equivalent.

Since the notation used in [25] is slightly different from ours, we first describe their key intermediate quantities using our notation. Their exponential weight $b^t \in \mathbb{R}^{XA}$ is defined by

$$b^t(x_h, a_h) = \exp\{-\eta \sum_{s=1}^{t} \ell_h^s(x_h, a_h)\}.$$

Then, their kernel function $K : \mathbb{R}^{XA} \times \mathbb{R}^{XA} \to \mathbb{R}$ is defined by

$$K_{x_g}(b, b') = \sum_{v \in \mathcal{V}^{x_g}} \sum_{(x_h, a_h) \in v} b(x_h, a_h) b'(x_h, a_h),$$

where $(x_h, a_h) \in v$ is a shorthand notation meaning that (x_h, a_h) is such that $v_{1:h}(x_h, a_h) = 1$.

We will also use $\mathbf{1} \in \mathbb{R}^{XA}$ to denote the all-ones vector in $\mathbb{R}^{XA}$. [25, Proposition 5.3] shows that the output policy μ^t of kernelized OMWU can be written in conditional-form as

$$\mu^t(a_h|x_h) = \frac{b^{t-1}(x_h, a_h) \prod_{x_{h+1} \in \mathcal{C}(x_h a_h)} K_{x_{h+1}}(b^t, \mathbf{1})}{K_{x_h}(b^{t-1}, \mathbf{1})}.$$

The key step within Farina et al. [25]'s Kernelized MWU implementation is the recursive evaluation of the quantity $K_{x_h}(b^{t-1}, \mathbf{1})$ in the bottom-up order over $x_h \in \mathcal{X}$, whereas our Algorithm 6's key step is the recursive evaluation of $F_{x_h}^t$ in the bottom-up order over $x_h \in \mathcal{X}_h$ in (68).

The following proposition shows that these two quantities are exactly equivalent, thereby showing the equivalence of the two implementations.

Proposition G.1. *We have for all $x_h \in \mathcal{X}$ and all $t \geq 1$ that*

$$K_{x_h}(b^{t-1}, \mathbf{1}) = \exp\{F_{x_h}^t\}.$$

Proof. We prove this by induction for $h = H = 1, \cdots, 1$. For $h = H + 1$, $K_{x_h}(b^{t-1}, 1) = 1$ and $F_{x_h}^t = 0$ by definition. If the claim holds for $h + 1$, then by Theorem 5.2 of [25],

$$\begin{aligned} K_{x_h}(b^{t-1}, \mathbf{1}) &= \sum_{a_h} \exp\{-\eta \sum_{s=1}^{t-1} \ell^s(x_h, a_h)\} \prod_{x_{h+1} \in \mathcal{C}(x_h a_h)} K_{x_{h+1}}(b^{t-1}, \mathbf{1}) \\ &= \sum_{a_h} \exp\{-\eta \sum_{s=1}^{t-1} \ell^s(x_h, a_h) + \sum_{x_{h+1} \in \mathcal{C}(x_h a_h)} F_{x_{h+1}^t}\} \\ &= \exp\{F_{x_h^t}\}. \end{aligned}$$

□

[5] Their Algorithm 1 is an optimistic algorithm with a "prediction vector". Here we are referring to their non-optimistic version where the prediction vectors are set to zero.