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 $\mathcal{O}(\mathrm{polylog}T)$ 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 $\widetilde{\mathcal{O}}(\|\Pi\|_1 \,\mathrm{polylog}T)$ 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* $\Pi \subset \mathbb{R}^d_{\geq 0}$ 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 $\{\ell^t\}_{t=1}^T \subset \mathbb{R}^d_{\geq 0}$, 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

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

By choosing $\eta > 0$, this result implies a quite desirable bound $\mathrm{Reg}^\Phi(T) \leq L\sqrt{2\log|\Phi_0|\cdot T}$ in the full-feedback setting (assuming bounded loss $\langle\phi\mu^t, \ell^t\rangle \leq L$), 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.

1: Initialize $p^1 \in \Delta_{\Phi_0}$ with $p^1_\phi = 1/|\Phi_0|$ for $\phi \in \Phi_0$.
2: **for** iteration $t = 1, \dots, T$ **do**
3: Compute $\phi^t = \sum_{\phi\in\Phi_0} p^t_\phi \phi$.
4: Set policy μ^t to be the fixed point of equation $\mu^t = \phi^t \mu^t$.
5: Receive loss function $\ell^t \in \mathbb{R}^d_{\geq 0}$, suffer loss $\langle \mu^t, \ell^t\rangle$.
6: 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

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

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

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

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

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

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 $(x_1, a_1, \ldots, x_{h-1}, a_{h-1})$ 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 $(x_1, a_1, \ldots, x_{h-1}, a_{h-1})$ 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

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

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*

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

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 $t \geq 1$:

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

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 $\eta > 0$.

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

2: 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 $F^t_{x_g a_g, x_h}$

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

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

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

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

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

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

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

7: 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}:*

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

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

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 $\{\mu^{\star,h}\}_{h\in[H]}$.

1: **for** $t = 1, 2, \ldots, T$ **do**

2: 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 $F^{\star,t}_{x_g a_g, x_h}$

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

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

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

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

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

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

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

7: 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).

8: 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

$$F^{\mathsf{Tr}}_{\mathsf{bal}}(M) := XA \log \textstyle\sum_{g,x_g,a_g} \exp\Big\{\frac{1}{XA}\big[-\big\langle I - E_{\succeq x_g a_g}, M\big\rangle + F^{\star}_{x_g a_g, x_g}(M)\big]\Big\}, \tag{21}$$

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),

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

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

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

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,

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

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:

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

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 $1-\delta$:*

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

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 poly(H) and log factors[2]. By the online-to-batch conversion (Appendix B.2), Theorem 7 also implies an $\widetilde{\mathcal{O}}(H^4XA/\varepsilon^2)$ sample complexity for learning EFCE under bandit feedback (assuming same game sizes for all m players). This improves over the best known $\widetilde{\mathcal{O}}(mH^6XA^2/\varepsilon^2)$ 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 $\Phi = \Phi^{\text{ext}} = \text{conv}\{\Phi_0^{\text{ext}}\}$ 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]

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

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 $\text{polylog}(T)$ 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.