UNIVERSITY OF SOUTHAMPTON
FACULTY OF ENGINEERING AND PHYSICAL SCIENCES
School of Electronics and Computer Science

Online Learning in the Presence of Strategic Adversary

by

Le Cong Dinh
ORCID: 0000-0002-3306-0603

A thesis for the degree of Doctor of Philosophy

Supervisors: Dr Long Tran-Thanh, Dr Alain Zemkoho and Dr Tri-Dung Nguyen
Examiners: Dr Tu Vuong Phan and Dr Panayotis Mertikopoulos

May 2023

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES
School of Electronics and Computer Science

Thesis

Online Learning in the Presence of Strategic Adversary

by Le Cong Dinh
ORCID: 0000-0002-3306-0603

This thesis offers a comprehensive exploration of the online learning problem in which an agent needs to strategise against a strategic adversary (also known as a no-regret adversary). Through examination of three interrelated settings, we have devised novel algorithms that achieve improved performance guarantees and last round convergence to the Nash Equilibrium both in theoretical and empirical contexts. Our findings open the door to further investigation of complex problems in online learning and game theory, where strategic adversaries play a crucial role in a multitude of applications.

In the first of the three main chapters comprising our study, we examine the problem of playing against a strategic adversary under a two-player zero-sum game setting. In this scenario, we introduce a new no-dynamic regret algorithm, namely the Last Round Convergence of Asymmetric Games (LRCA), that achieves last round convergence to the minimax equilibrium. Building on this work, the second main chapter investigates the more general problem of online linear optimization and proposes several new algorithms, including Online Single Oracle (OSO), Accurate Follow the Regularized Leader (AFTRL), and Prod-Best Response algorithm (Prod-BR). These algorithms achieve state-of-the-art performance guarantees against a strategic adversary, such as no-forward regret and no-dynamic regret. Additionally, we show that a special case of AFTRL, the Accurate Multiplicative Weights Update (AMWU), can achieve last round convergence to the Nash equilibrium in self-play settings. In the third and final main chapter, we extend our results to the challenging setting of Online Markov Decision Processes (OMDPs), which have many significant applications in practice. Here, we propose two new algorithms, MDP-Online Oracle Expert (MDP-OOE) and Last Round Convergence-OMDP (LRC-OMDP), that achieve no-policy regret and last round convergence to the Nash equilibrium, respectively, against a strategic adversary.

Contents

List of Figures

Declaration of Authorship

I, Le Cong Dinh, declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:

   1. Dinh, Le Cong, Nguyen, Tri-Dung, B. Zemhoho, Alain, and Tran-Thanh, Long. "Last round convergence and no-dynamic regret in asymmetric repeated games." In Algorithmic Learning Theory, pp. 553-577. PMLR, 2021.

   2. Dinh, Le Cong. "Online Learning against Strategic Adversary." In Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, pp. 1841-1842. 2022.

   3. Dinh, Le Cong, McAleer, Stephen Marcus, Tian Zheng, Perez-Nieves, Nicolas, Slumbers, Oliver, Henry Mguni, David, Wang, Jun, Bou Ammar, Haitham and Yang, Yaodong. "Online Double Oracle." In Transactions on Machine Learning Research, TMLR, 2022.

   4. Dinh, Le Cong, Henry Mguni, David, Long, Tran-Thanh, Wang, Jun, and Yaodong, Yang. "Online Markov decision processes with non-oblivious strategic adversary." Autonomous Agents and Multi-Agent Systems 37, 15 (2023).

Signed: Date:

Acknowledgements

I would like to express my heartfelt gratitude to my amazing supervisors, Long Tran-Thanh, Alain Zemkoho and Tri-Dung Nguyen, for their unwavering support and guidance throughout my PhD journey. Your knowledge, expertise and encouragement have been instrumental in shaping me into the researcher I am today.

I am immensely grateful to the University of Southampton and the ECS IDS DTP Studentship for giving me the opportunity to pursue my PhD studies. The generous support provided by the University and the Studentship has been instrumental in enabling me to undertake this exciting and challenging academic journey.

I am thankful to my family for their love and support. My dad, Dr The Dinh Dinh, is the main reason for me to pursue an academic path. My mom, Thi Kim Oanh Le, has given me the strength to keep pushing forward, even during the most challenging times. My brother, Nho Lam Dinh, has given me the confidence to go outside the world while he takes care of our family back home. My girlfriend, Dieu Linh Dao, has been by my side from the beginning, and together we have overcome challenges and will continue to conquer the world.

Lastly, but certainly not least, I would like to thank Stefano Coniglio for his invaluable guidance and wonderful perspectives. I am also thankful to the members of the Vietnamese society in Southampton, who have been there to cheer me up and motivate me to achieve better results. Your encouragement and support have been a source of inspiration for me, and I feel truly fortunate to have had you in my corner throughout this journey.

Thank you all for your support and for being a part of my journey.

Nomenclature

Chapter 2: Last Round Convergence to NE Against Strategic Adversary

$\boldsymbol{A}$	payoff matrix in the zero-sum game
T	number of rounds in the repeated game
n	number of pure strategies for the row player
m	number of pure strategies for the column player
$\boldsymbol{x}_t$	mixed strategy of the row player at round t
$\boldsymbol{y}_t$	mixed strategy of the column player at round t
Δ_n	the $n-$dimensional simplex
$(\boldsymbol{x}^*, \boldsymbol{y}^*)$	the minimax equilibrium of the two-player zero-sum game
$\top$	vector transpose
v	minimax value of a two-player zero-sum game
$RE(.\|\|.)$	the relative entropy or K-L divergence
DR_T	dynamic regret
μ_t	learning rate at round t
$\boldsymbol{e}_i$	unit-vector with 1 at the $i-$th component
$\|.\|$	norm of a vector
$\|.\|_*$	dual norm of $\|.\|$

Chapter 3: Achieving Better Regret Against Strategic Adversary

$\boldsymbol{f}_t$	strategy of the learner at round t
$\mathcal{F}$	strategy domain of the learner
$\boldsymbol{x}_t$	loss vector chosen by the environment at round t
$exp()$	natural exponential function
T	total number of rounds in play
n	size of learner's strategy
$\boldsymbol{a}^i$	pure strategy of the row player at row i
$\boldsymbol{c}^j$	pure strategy of the column player at column j
$supp(.)$	support of a vector: number of non-zero elements
v	value of the game

$(\boldsymbol{f}^*, \boldsymbol{y}^*)$	the Nash Equilibrium of the game
$\|.\|_p$	p-norm of a vector
Π_t	effective strategy set at round t
Π	row player (i.e., the learner) 's pure strategy set
C	the column player's pure strategy set
k	size of the effective strategy set at the final time window
α	exploiting rate
η	learning rate
M_t	prediction of the strategy of the environment ($\boldsymbol{x}_t$) at round t
$D_{\mathcal{R}(.,.)}$	Bregman divergence with respect to $\mathcal{R}$
$\mathcal{R}$	regularizer function

Chapter 4: Online Markov Decision Processes Against Strategic Adversary

L	size of adversary's pure loss vectors
Δ_L	the action space of the adversary: a simplex of size L
A	agent's action space
$\|A\|$	size of agent's action space
τ	mixing time constant
π_t	agent's policy at round t
S	state space
$\|S\|$	size of the state space
$P(.\|.)$	transition model
$P^a_{s,s'}$	probability of transitioning from state s to s' by taking the action a
$P(\pi)_{s,s'}$	probability of transitioning from state s to s' by taking the policy π
$\boldsymbol{l}_t(.)$	loss function at round t
$\boldsymbol{d}_{\pi_t}$	stationary distribution of policy π_t (also denote $\boldsymbol{d}_t$)
$\boldsymbol{d}_\Pi$	stationary distribution set from all agent's deterministic policies
Δ_{d_Π}	the action space of the agent at each round: the simplex of size $\|d_\Pi\|$
$\boldsymbol{v}^\pi_t(x,a)$	the probability of (state, action) pair (x,a) at time step t
$\boldsymbol{v}_t(x,a)$	(state, action) pair (x,a) distribution at time t when following $\pi_1, \pi_2 \ldots$
$\eta_{\boldsymbol{l}}(\pi)$	average loss of policy π with respect to the loss $\boldsymbol{l}$
$Q_{\pi,\boldsymbol{l}}(s,a)$	accumulated loss of the agent at the (state, action) pair (s,a)
$\boldsymbol{l}_t^{\pi_t}$	the loss function at round t while the agent follows $\pi_1, \ldots, \pi_T$
$\boldsymbol{f}_t^\pi$	the loss function at round t against the fixed policy π of the agent
$P(\pi)$	state transition matrix
$\|.\|_1$	l_1 norm of a vector
v	the minimax value of the zero-sum game

$\langle\ ,\rangle$	the dot product
$\mathbb{E}$	the expectation
$supp()$	support size of a vector
$(\boldsymbol{d}_{\pi^*}, \boldsymbol{l}^*)$	the minimax equilibrium of the zero-sum game (also known as the Nash Equilibrium)
T_i	time window i
A_t^s	effective strategy set in state s at time t
$BR(.)$	best response strategy
$(\bar{\boldsymbol{l}})$	average loss function
k	number of time windows
$RE(.\|\|.)$	the relative entropy distance

Chapter 1

Introduction

Online learning and prediction of individual sequences have been extensively studied in various fields, including game theory, operational research, and machine learning, owing to their diverse applications in weather forecasting, financial stock analysis, online advertisement placement, online web ranking, and classification (Cesa-Bianchi and Lugosi, 2006; Shalev-Shwartz, 2012). The key idea is to predict the next element of an unknown sequence given some knowledge about past elements and side information. In the classical statistical theory of sequential prediction, researchers often rely on the stationarity assumption, where the elements of the sequence are supposed to follow a stationary stochastic process. Under this assumption, the past observations can be used to estimate the statistical properties and then a prediction rule can be derived to achieve a near-optimal strategy. Two popular algorithms that provably converge to the optimal strategy are ϵ-greedy (Sutton and Barto, 2018) and Upper Confidence Bound (Auer et al., 2002). However, the underlying mechanisms governing the elements of a sequence in many real-world applications may be unknown, whether deterministic, stochastic or adversarially adaptive, rendering the classical approach unsuitable. To address this issue, researchers have proposed the prediction of individual sequences, which abandons stochastic assumptions and treats the sequence elements as products of unknown and unspecific mechanisms (Cesa-Bianchi and Lugosi, 2006). This approach has been extensively studied, resulting in significant algorithmic advancements and relevant properties. These algorithms offer robust performance in adversarial environments, ensuring that on average, the agent does not perform worse than a baseline strategy in hindsight. The most widely-used baseline is the best-fixed strategy in hindsight, with algorithms satisfying this criterion referred to as no-(external) regret algorithms.

No-regret or no-external algorithms are widely used in the fields of online learning and algorithmic game theory due to their attractive worst-case performance guarantees (Cesa-Bianchi and Lugosi, 2006). These algorithms provably guarantee that the average payoff of the strategies played will not be significantly worse than the best-fixed strategy in hindsight, regardless of the encountered sequences. As a result, they are commonly

used in playing against adversaries, strategizing in unknown environments, or solving two-player zero-sum games, leading to average convergence to a Nash Equilibrium (NE) under self-play settings (Zinkevich et al., 2007; Cesa-Bianchi and Lugosi, 2006; Lanctot et al., 2017). However, in order to maintain a small regret bound, no-(external) regret algorithms such as Multiplicative Weights Update (Littlestone and Warmuth, 1994; Freund and Schapire, 1999), Follow the Regularized Leader (Abernethy et al., 2008), Follow the Perturbed Leader (Kalai and Vempala, 2005), and Mirror Descent (Nemirovskij and Yudin, 1983) must keep their learning rate small, resulting in a slow change in the strategy profile. This predictability is further compounded by the fact that the behaviour of no-(external) regret algorithms depend heavily on feedback from the environment. As a result, the sequence of strategies played by no-(external) regret algorithms is highly correlated to its predecessors. Against a no-(external) regret learning opponent, the loss sequence encountered by the learner/player is not entirely arbitrarily adversarial in each round, making the worst-case performance guarantees not attractive for the learner. Therefore, it is desirable to develop a learning algorithm that can exploit the extra structure when playing against an agent following a no-(external) regret algorithm (also known as a strategic adversary: in our work, we will use the two terms *strategic adversary* and *no-(external) regret adversary* interchangeably), and answer the question:

Can we exploit the strategic adversary?

In this thesis, we focus on three important properties that an ideal algorithm should have in order to exploit a strategic adversary:

- Better performance: Exploit the extra information when playing against strategic adversaries to achieve better performance.
- Last round convergence: Convergence to a NE in self-play setting (state-of-the-art solver for a NE).
- Robustness: Maintain no-(external) regret bound in the worst-case scenario.

Better performance against the strategic adversary: It is well-known that famous no-regret algorithms such as Multiplicative Weights Update (Freund and Schapire, 1999) or Follow the Regularized Leader (Abernethy et al., 2008) achieve optimal performance guarantee in a fully adversarial setting (Cesa-Bianchi and Lugosi, 2006). However, given the extra knowledge from the strategic adversary, the worst-case performance guarantee of these famous no-regret algorithms is too pessimistic. Therefore, it is desirable to develop new algorithms that achieve better performance guarantees against the strategic adversary. In the literature of online learning in repeated games, Deng et al. (2019) proposed a fixed strategy for an agent playing against a no-external regret adversary, assuming that the agent knows the game structure such as the payoff matrix and the

utility function. The proposed strategy can ensure a Stackelberg value, which is optimal in certain games, such as general-sum games. However, this approach cannot be applied in many practical scenarios where the environment or game structure is unknown, or the adversary does not follow no-regret algorithms. To overcome these limitations, Chiang et al. (2012) and Rakhlin and Sridharan (2013a) considered a different setting where the agent has access to a prediction M_t of $\boldsymbol{x}_t$ before making a decision at round t. Their proposed algorithm, Optimistic Follow the Regularized Leader (OFTRL), has an external regret that depends linearly on $\sqrt{\sum_{t=1}^{T} \|\boldsymbol{x}_t - M_t\|_*^2}$. However, with an accurate prediction (i.e., $M_t \approx \boldsymbol{x}_t$), one could expect a stronger performance guarantee than the no-external regret of OFTRL. It is worth noting that OFTRL assigns a fixed weight of 1 to the prediction M_t, which can limit the advantage of the additional knowledge in the learning process. Our work demonstrates that playing against a strategic adversary can lead to an accurate prediction of $\boldsymbol{x}_t$, thereby offering an opportunity to enhance the performance of OFTRL. In Chapter 3, we propose improved techniques for leveraging the accurate prediction M_t of $\boldsymbol{x}_t$ to achieve better performance guarantee, as outlined in Theorems 3.14 and 3.18.

Last round convergence: The convergence of average strategies to a minimax equilibrium (i.e., the NE) has been well-established when both players employ no-regret algorithms, with a convergence rate of $\mathcal{O}(T^{-1/2})$ as cited in Freund and Schapire (1999). Further developments in no-regret algorithms by Daskalakis et al. (2011) and Rakhlin and Sridharan (2013b) have led to near-optimal convergence rates of $\mathcal{O}(\frac{\log(T)}{T})$. However, despite extensive literature on no-regret learning, one unsatisfactory result is the average convergence to the NE. That is, in two-player zero-sum games, no-regret algorithms such as Multiplicative Weights Update (MWU) (Freund and Schapire, 1999) or Follow the Regularized Leader (FTRL) (Abernethy et al., 2008) will only lead to average convergence instead of last round convergence to the NE. Specifically, Bailey and Piliouras (2018) has demonstrated that the multiplicative weights update (MWU) algorithm leads to convergence of the last round strategy to the boundary in games with an interior Nash equilibrium point, while Mertikopoulos et al. (2018) has identified Poincaré recurrence as an undesirable feature arising from regularized learning, leading to cyclic behaviour in strategy dynamics. The average convergence will not only increase the computational and memory overhead but also make things difficult when using a neutral network in the solution process in which averaging is not always possible (Bowling et al., 2015). For game theory and modern applications of online learning in optimization such as training Generative Adversarial Networks (Daskalakis et al., 2018), last round convergence plays a vital role in the process, thus it is crucial to develop algorithms that can lead to last round convergence. Recently, Daskalakis and Panageas (2019) have proven the attainment of last round convergence to the minimax equilibrium by both players using the optimistic multiplicative weights update algorithm (OMWU) under the assumption of a unique equilibrium point. However, this result hinges on the calculation of the constant step size of the update mechanism from the game's payoff matrix $\boldsymbol{A}$.

Therefore, a lack of knowledge of $\boldsymbol{A}$ on the row player's part precludes the guarantee of last round convergence by OMWU. Furthermore, in scenarios where the row player employs distinct no-regret algorithms such as MWU or FTRL, which have widespread use in various applications, OMWU cannot ensure last round convergence. This issue prompts the question of whether a robust algorithm exists that is capable of achieving last round convergence to the minimax equilibrium even when playing against different no-regret algorithms. In this thesis, we conduct a thorough investigation of this problem and introduce two algorithms: Last Round Convergence in Asymmetric Games (Algorithm 11) and Last Round Convergence in Online Markov Decision Processes (Algorithm 21). These algorithms can achieve last round convergence against the strategic adversary in the two-player zero-sum game and online Markov decision processes settings, respectively.

Robustness against the general adversary: While our algorithms are designed to play against strategic adversaries, there exists the possibility that the additional information or our predictions may be inaccurate, resulting in suboptimal performance against a general type of adversary. Hence, it is important to develop algorithms that can leverage the feedback structure when playing against strategic adversaries while maintaining the standard no-(external) regret bound against general adversaries. Various works, such as Cesa-Bianchi et al. (2007); Even-Dar et al. (2008); Sani et al. (2014), have explored this goal and introduced a class of algorithms called "Prod" that achieves improved regret bounds against "easy data" while maintaining the no-(external) regret bound in the worst-case scenario. The Prod algorithm combines two separate algorithms, one for easy data and another for the worst-case scenario, and tunes the weight between them based on their performance in each round of the game. Since we can exploit the structure of a strategic adversary, we can view it as an easy data opponent and apply the Prod technique to enhance our algorithm's performance. Another approach, as described in De Rooij et al. (2014), is to use the FlipFlop algorithm, which interleaves the Adahedge algorithm (for general adversary) and the Follow-the-Leader algorithm (for easy data) to achieve the best of both worlds. In our work, we propose several algorithms (e.g., Algorithms 12, 15 and 20) that can exploit the strategic adversary to achieve state-of-the-art performance while maintaining no-regret guarantees against the general adversary.

This thesis aims to provide a comprehensive investigation into the problem of playing against the strategic adversary. To achieve this goal, we examine this problem in three interrelated settings: two-player zero-sum games, online linear optimization, and online Markov Decision Processes. These settings are widely studied in the literature and have important practical applications. In the following sections, we provide an overview of the fundamental concepts and related algorithms for each setting. A more specific literature review will be presented in the relevant chapters.

1.1 Two-player Zero-sum Games

Repeated two-player zero-sum games have been extensively studied in game theory literature. Formally, the game is characterized by an $n \times m$ payoff matrix $\boldsymbol{A}$, with the entries of $\boldsymbol{A}$ assumed to be within the interval $[0, 1]$ without loss of generality. The rows and columns of $\boldsymbol{A}$ correspond to the pure strategies of the row and column players, respectively. At round t, the set of feasible strategies available to the row player is denoted by Δ_n, which consists of all $\boldsymbol{x}_t \in \mathbb{R}^n$ such that $\sum_{i=1}^{n} \boldsymbol{x}_t(i) = 1$ and $\boldsymbol{x}_t(i) \geq 0$ for all $i \in \{1, ..., n\}$. Similarly, the set of feasible strategies for the column player is denoted by Δ_m. We use $\boldsymbol{e}_i$ to denote the pure strategy where $\boldsymbol{e}_i(i) = 1$. If the row (resp. column) player selects a mixed strategy $\boldsymbol{x}_t \in \Delta_n$ (resp. $\boldsymbol{y}_t \in \Delta_m$) at round t, the row player's payoff is $-\boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t$, while the column player's payoff is $\boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t$. Consequently, the row (resp. column) player aims to minimize (resp. maximize) the value of $\boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t$. John von Neumann's minimax theorem (Neumann (1928)), which is fundamental in zero-sum games can be expressed as:

$$\max_{\boldsymbol{y} \in \Delta_m} \min_{\boldsymbol{x} \in \Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} \quad = \quad \min_{\boldsymbol{x} \in \Delta_n} \max_{\boldsymbol{y} \in \Delta_m} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} \quad = \quad v \tag{1.1}$$

for some $v \in \mathbb{R}$. We call a point $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ satisfying the minimax theorem Equation (1.1) *the minimax equilibrium of the game*. We also consider another important type of equilibrium, ϵ-Nash Equilibrium in our work.

Definition 1.1 (ϵ**-Nash Equilibrium**). Assume $\epsilon > 0$. We call a point $(\boldsymbol{x}, \boldsymbol{y}) \in \Delta_n \times \Delta_m$ ϵ-NE if

$$max_{\boldsymbol{y} \in \Delta_m} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} - \epsilon \leq \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} \leq min_{\boldsymbol{x} \in \Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} + \epsilon.$$

Clearly, if the details of the game are known (i.e., the game's payoff matrix $\boldsymbol{A}$), then the adversary can calculate their minimax strategies and follow these strategies in each round. In our thesis, we consider a much weaker scenario, a completely-uncoupled dynamics that can be described as follows:

- Each player only knows their own pure strategies and not the game matrix or the number of strategies their opponent has.
- Players interact in rounds and choose a mixed strategy in each round.
- At the end of each round, each player is informed about the expected payoff they would have received if they had played each of their pure strategies against their opponent's mixed strategy, although the mixed strategy used by the opponent is not disclosed to them (this type of feedback is also known as *full information* feedback).

In the following section, we introduce one of the most famous completely-uncoupled dynamics in game theory, namely *fictitious play*.

1.1.1 Fictitious Play

The Fictitious play (FP) was first introduced by Brown (1949), in which he believed the strategies would ultimately reach the value of a zero-sum game. Robinson (1951) later confirmed the convergence properties of this method in a two-player zero-sum game.

Algorithm 1 Fictitious Play

Input: arbitrary strategy $\boldsymbol{x}_1, \boldsymbol{y}_1$ of the row and column players.
for $t = 1, 2, \ldots$ **do**
  the row player follows $\boldsymbol{x}_{t+1}$ such that $\boldsymbol{x}_{t+1} = \text{argmin}_{i\in[n]} \boldsymbol{e}_i^\top \boldsymbol{A} \left(\sum_{j=1}^t \boldsymbol{y}_j\right)$
  the column player follows $\boldsymbol{y}_{t+1}$ such that $\boldsymbol{y}_{t+1} = \text{argmax}_{i\in[m]} \left(\sum_{j=1}^t \boldsymbol{x}_j\right)^\top \boldsymbol{A}\boldsymbol{e}_i$
end for

Definition 1.2 (Fictitious play Brown (1949))**.** Fictitious play is the completely uncoupled dynamics in which in every round, each player responds with their best strategy to their opponent's historical strategy.

It is well-known that following the FP algorithm, the average strategy of both players will converge to the NE in certain games. In particular, FP will lead to convergence in two-player zero-sum game (Robinson, 1951), potential games (Monderer and Shapley, 1996), general-sum $2 \times N$ games (Berger, 2005) and games that is solvable by iterative elimination of strictly dominated strategies (Nachbar, 1990). However, Brandt et al. (2010) shows that, in an example of a constant-sum game, FP requires an exponentially large number of rounds over the size of the representation of the game to converge. It is consistent with the finding of Daskalakis and Pan (2014) in which, by unravelling the induction step in the convergence proof of Robinson (1951), the convergence rate of FP in a two-player zero-sum game can be bounded by $O(t^{-\frac{1}{m+n-2}})$.

When the exact best responses in the FP dynamic are hard to calculate (e.g., the game matrix $\boldsymbol{A}$ is large), the approximate best responses can be used without affecting the convergence guarantee of the method. In particular, Van der Genugten (2000) proposed the Weakened fictitious play, in which the strategies played at each step only need to be ϵ-best response, with $\epsilon \to 0$ as time progresses. Leslie and Collins (2006) further extended this result to propose generalised weakened fictitious play (GWFP) with more relaxation into the progress. The convergence result of GWFP leads to many applications in the machine learning domain where the exact best response is costly to obtain.

The undesirable exponential convergence property of FP leaves a question of whether there exists learning dynamics that can theoretically achieve a faster convergence rate

to NE in a completely-uncoupled dynamic. In the next sections, we introduce a class of algorithms, namely "no-(external)regret" algorithms that achieve a convergence rate of $O(t^{-\frac{1}{2}})$ in the two-player zero-sum game setting.

1.1.2 Multiplicative Weights Update and its Variants

Along with convergence in the self-play setting, one of the important aspects of a learning algorithm is the average performance (i.e., the cumulative loss). Traditionally, the performance is measured using the *external-regret* bound which is the difference between the cumulative loss of the player and the best-fixed strategy in hindsight. Formally, it is defined as follows:

Definition 1.3 (**No-external regret**)**.** Let $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ be a sequence of mixed strategies played by the column player. An algorithm of the row player that generates a sequence of mixed strategies $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ is called a *no-external regret* algorithm (or no-regret algorithm) if we have

$$\lim_{T\to\infty} \frac{R_T}{T} = 0,$$
$$\text{where } R_T := \min_{\boldsymbol{x}\in\Delta_n} \left(\sum_{t=1}^{T} \boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t - \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}_t \right) \text{ denotes the external regret.}$$

One of the most well-studied no-regret algorithms in the game theory literature is the multiplicative weights update (MWU) method, which can be defined as follows:

Algorithm 2 Multiplicative Weights Update

1: **for** $t = 1, 2, \ldots$ **do**
2: Predict a vector $\boldsymbol{x}_{t+1}$ such that $\boldsymbol{x}_{t+1}(i) = \boldsymbol{x}_t(i) \frac{e^{-\mu_t \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_t}}{\sum_{i=1}^{n} \boldsymbol{x}_t(I) e^{-\mu_t \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_t}}$, $i \in \{1, \ldots n\}$
3: **end for**

Definition 1.4 (**MWU** Freund and Schapire (1999))**.** Let $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ be a sequence of mixed strategies played by the column player. The row player is said to follow the MWU algorithm if strategy $\boldsymbol{x}_{t+1}$ is updated as follows:

$$\boldsymbol{x}_{t+1}(i) = \boldsymbol{x}_t(i) \frac{e^{-\mu_t \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_t}}{Z_t}, \; i \in \{1, \ldots n\},$$
$$\text{where } \begin{cases} Z_t = \sum_{i=1}^{n} \boldsymbol{x}_t(i) e^{-\mu_t \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_t}, \mu_t \in [0, \infty) \text{ is a parameter,} \\ \boldsymbol{e}_i, \; i \in \{1, \ldots, n\}, \;\; \text{is the unit-vector with 1 at the } ith \text{ component.} \end{cases}$$

When T is known in advance, by fixing the learning rate $\mu_t = \sqrt{8 \log(n)/T}$, we can achieve the optimal regret bound for MWU (Theorem 2.2 in (Cesa-Bianchi and Lugosi, 2006)):

$$R_T \leq \sqrt{T \log(n)/2}.$$

When T is unknown, we can apply the following Doubling Trick to bound the regret of MWU.

Algorithm 3 The Doubling Trick

1: **Input:** algorithm A whose parameters depend on the time horizon
2: **for** $n = 0, 1, 2, \ldots$ **do**
3: run A on rounds $t = 2^n, \ldots, 2^{n+1} - 1$
4: **end for**

Definition 1.5 (**The Doubling Trick** Cesa-Bianchi and Lugosi (2006))**.** For an algorithm with the regret $\alpha\sqrt{T}$ with a parameter depends on the time-horizon T. The Doubling Trick restarts the algorithm at round 2^m for m=0,1,2... Following the doubling trick, the total regret is not bigger than the sum of the regret in each part. Thus, we have

$$R_T \leq \sum_{i=1}^{\lceil \log_2(T) \rceil} \alpha\sqrt{2^i} \leq \alpha \frac{1 - \sqrt{2T}}{1 - \sqrt{2}} \leq \frac{\sqrt{2}}{\sqrt{2} - 1} \alpha\sqrt{T}.$$

Therefore, following the Doubling Trick, in case T is unknown, MWU can achieve the external regret bound of

$$(\sqrt{2}/(\sqrt{2} - 1))\sqrt{T \log(n)/2},$$

which is worse than the optimal one when T is known by a factor of $\sqrt{2}/(\sqrt{2} - 1)$.

De Rooij et al. (2014) proposed a better algorithm for unknown T, AdaHedge, a variant of MWU with adaptive learning rate $\mu_t = \log(n)/\Delta_{t-1}$ where Δ_t denotes the cumulative mixability gap:

Definition 1.6 (**AdaHedge** De Rooij et al. (2014))**.** Let $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ be a sequence of mixed strategies played by the column player. The row player is said to follow AdaHedge if $\boldsymbol{x}_{t+1}$ is updated as follows:

$$\boldsymbol{x}_{t+1}(i) = \frac{\exp\left(-\mu_t \sum_{j=1}^{t-1} \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_j\right)}{\sum_{i=1}^{n} \exp\left(-\mu_t \sum_{j=1}^{t-1} \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_j\right)}, \ \forall i \in [n]$$

where n is the number of pure strategies and $\mu_t > 0$ is an adaptive learning rate such as: $\mu_t = \log(n)/\Delta_{t-1}$. Δ_t denotes the cumulative mixability gap, which can be derived from historical data (Equation (5) in De Rooij et al. (2014)).

Then applying Theorem 8 in De Rooij et al. (2014) to our setting with $S = 1$, $L^+ - L^- \leq T$ we have:

$$\begin{aligned} R_T^{AdaHedge} &\leq 2\sqrt{\frac{(L^+ - L^*)(L^* - L^-)}{L^+ - L^-} \log(n)} + \frac{16}{3}\log(n) + 2 \\ &\leq 2\sqrt{\frac{(L^+ - L^-)^2/4}{L^+ - L^-} \log(n)} + \frac{16}{3}\log(n) + 2 \leq \sqrt{T\log(n)} + \frac{16}{3}\log(n) + 2, \end{aligned}$$

which is worse than the optimal one when T is known by a factor of $\sqrt{2}$.

Another variant of MWU is the Linear Multiplicative Weights Update (LMWU), which later plays an important role in the analysis of our algorithms:

Definition 1.7 (**LMWU** Cesa-Bianchi et al. (2007))**.** The row player is said to play the Linear Multiplicative Weights Update if the row player updates the strategy as follows:

$$\boldsymbol{x}_{t+1}(i) = \frac{\boldsymbol{x}_t(i)(1 - \mu_t \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_t)}{\sum_{j=1}^n \boldsymbol{x}_t(j)(1 - \mu_t \boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}_t)} \; \forall i \in \{1, \dots n\}.$$

By properly choosing the learning rate μ_t (e.g., see Theorem 1 in Cesa-Bianchi et al. (2007)), LMWU is a no-(external) regret with the regret bound depends on the sum of the loss square (i.e., $\sum_{j=1}^T (\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_j)^2$). Because of this dependency, LMWU is also called a no-(external) regret algorithm with a second-order regret bound.

Despite their optimal regret bound in the worst-case scenario, MWU and its variants (AdaHedge and LMWU) have an undesirable property: non-last round convergence in a self-play setting. Specifically, Bailey and Piliouras (2018) proved that if both players follow MWU then in the case of interior minimax equilibrium, the strategies will move away from the equilibrium and towards the boundary. This undesirable feature causes many issues in game theory and applications, including unwanted cyclic behaviour in training Generative Adversarial Networks (GANs). Thus, a learning dynamic leading to last round convergence is important in the development of the field (e.g., see Daskalakis et al. (2018) for more details). Daskalakis and Panageas (2019) proved a variant of MWU, Optimistic Multiplicative Weights Update algorithm (OMWU), converge last round to the NE in a self-play setting.

Definition 1.8 (**OMWU** Daskalakis and Panageas (2019))**.** Let $\boldsymbol{y}_1, \boldsymbol{y}_2, ...$ be a sequence of mixed strategies played by the column player. The row player is said to follow the OMWU algorithm if strategy $\boldsymbol{x}_{t+1}$ is updated as follows:

$$\begin{aligned} &\boldsymbol{x}_{t+1}(i) = \boldsymbol{x}_t(i)\frac{e^{-\mu \boldsymbol{e}_i^\top \boldsymbol{A}(2\boldsymbol{y}_t - \boldsymbol{y}_{t-1})}}{Z_t}, \; i \in \{1, \dots n\}, \\ &\text{where} \begin{cases} Z_t = \sum_{i=1}^n \boldsymbol{x}_t(i) e^{-\mu \boldsymbol{e}_i^\top \boldsymbol{A}(2\boldsymbol{y}_t - \boldsymbol{y}_{t-1})}, \mu \in [0, \infty) \text{ is a learning rate,} \\ \boldsymbol{e}_i, \; i \in \{1, \dots, n\}, \;\; \text{is the unit-vector with 1 at the } ith \text{ component.} \end{cases} \end{aligned}$$

When the game $\boldsymbol{A}$ has a unique NE equilibrium, then OMWU with arbitrarily small learning rate μ leads to last round convergence to the NE in the self-play setting (e.g., Theorem 1.1 in Daskalakis and Panageas (2019)). Furthermore, Wei et al. (2020) proves that OMWU can achieve a linear last round convergence rate with a *universal constant* learning rate. The works around OMWU have motivated us to derive the convergence properties for our algorithms.

Along with last round convergence, our work also focuses on achieving better performance against the strategic adversary. In order to do that, we consider a stronger notion of regret compared to external regret:

Definition 1.9 (**Dynamic Regret** Besbes et al. (2015))**.** Let $\boldsymbol{y}_1, \boldsymbol{y}_2, \dots$ be a sequence of mixed strategies played by the column player. An algorithm of the row player that generates a sequence of mixed strategies $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots$ is called a *no-dynamic regret* algorithm if we have

$$\lim_{T\to\infty} \frac{DR_T}{T} = 0, \quad \text{where } DR_T := \sum_{t=1}^{T} \left(\boldsymbol{x}_t^\top \boldsymbol{A} \boldsymbol{y}_t - \min_{\boldsymbol{x}\in\Delta_n} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y}_t \right).$$

The no-dynamic regret is a much stronger type of regret compared to the normal no-regret where the regret $R_T = \min_{\boldsymbol{x}\in\Delta_n} \frac{1}{T} \left(\sum_{t=1}^{T} (\boldsymbol{x}_t - \boldsymbol{x})^\top \boldsymbol{A} \boldsymbol{y}_t \right)$ (i.e., the best-fixed strategy on average) is considered instead of DR_T (i.e., the best strategy at each round). In the literature, the no-dynamic regret is desirable but impossible to achieve with the current state-of-the-art algorithms in the adversarial symmetric setting.

1.2 Online Linear Optimization

Algorithm 4 Online Linear Optimization

1: **Input:** A convex set $\mathcal{F}$
2: **for** $t = 1, 2, \dots$ **do**
3: predict a vector $\boldsymbol{f}_t \in \mathcal{F}$
4: receive a loss function $l_t(\boldsymbol{f}) := \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle \ \ \forall \boldsymbol{f} \in \mathcal{F}$
5: suffer loss $l_t(\boldsymbol{f}_t)$
6: **end for**

In Chapter 3, we study the online linear optimization setting in which at round t, the learner chooses a strategy $\boldsymbol{f}_t \in \mathcal{F}$, where $\mathcal{F} \subset [0,1]^n$ [1] is a convex compact set. Simultaneously, the environment reviews a loss vector $\boldsymbol{x}_t \in [0,1]^n$ and the learner suffers the loss: $\langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle$.

The goal of the learner is to minimize the total loss after T rounds: $\min_{\boldsymbol{f}_1,\dots\boldsymbol{f}_T} \sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle$.

1.2.1 Follow-the-Leader

The most natural method of learning is to utilize the vector that has the lowest loss from all previous rounds in each online iteration. In online learning literature, this method is usually referred to as Follow-the-Leader (FTL).

[1] All the results remains true for bounded domain of strategy and loss vector.

Algorithm 5 Follow-the-Leader

1: **for** $t = 1, 2, \ldots$ **do**
2: predict a vector $\boldsymbol{f}_t := \text{argmin}_{\boldsymbol{f} \in \mathcal{F}} \sum_{i=1}^{t-1} l_i(\boldsymbol{f})$
3: **end for**

Definition 1.10 (**FTL** Shalev-Shwartz (2012))**.** Let $l_1, l_2, ...$ be a sequence of loss functions generated by the environment. The learner is said to follow the Follow-the-leader algorithm if strategy $\boldsymbol{f}_t$ is updated as follows:

$$\boldsymbol{f}_t := \operatorname*{argmin}_{\boldsymbol{f} \in \mathcal{F}} \sum_{i=1}^{t-1} l_i(\boldsymbol{f}).$$

Against a stochastic setting (i.e., stationary stochastic assumption of loss function), FTL guarantees a constant regret and therefore has been studied extensively in this setting (De Rooij et al., 2014; McMahan, 2011; Huang et al., 2016). In an adversarial setting, when the loss function l_t has the quadratic structure: $l_t(\boldsymbol{f}) = \frac{1}{2}\|\boldsymbol{f} - \boldsymbol{x}_t\|_2^2$, then by following FTL, the learner can guarantee an external regret of $4(\log(T) + 1)$ (e.g., see Corollary 2.2 in Shalev-Shwartz (2012) for more detail). However, in the setting of online linear optimization, FTL fails to guarantee a low external regret bound. The following example in Shalev-Shwartz (2012) demonstrates why FTL can possibly achieve a high regret against an adversary.

Example 1.1 (**Failure of FTL**)**.** *Consider* $\mathcal{F} = [-1, 1] \in \mathbb{R}$ *and the sequence of linear loss functions such that* $l_t(\boldsymbol{f}) = \boldsymbol{x}_t \boldsymbol{f}$ *where*

$$\boldsymbol{x}_t = \begin{cases} -0.5, & \text{if } t = 1 \\ 1, & \text{if } t \text{ is even} \\ -1, & \text{if } t > 1 \text{ and } t \text{ is odd.} \end{cases}$$

Following the prediction rule of FTL, $\boldsymbol{f}_t = 1$ *for odd* t *values and* $\boldsymbol{f}_t = -1$ *for even* t *values, leading to the cumulative loss of* T *for the FTL algorithm. On the other hand, the cumulative loss of a fixed strategy* $\boldsymbol{f}_t = 0$ *is* 0*. Therefore, the external regret of FTL is* T*.*

The failure of FTL in the aforementioned scenario can be explained by the fact that its predictions are inconsistent - the value of $\boldsymbol{f}_t$ changes significantly with each iteration, even though only one loss function was added to the optimization problem's objective. On the other hand, FTL performs well in the case of a quadratic game (e.g., see Corollary 2.2 in Shalev-Shwartz (2012)) because the value of $\boldsymbol{f}_{t+1}$ remains "near" $\boldsymbol{f}_t$. To resolve this instability, FTL can be improved by incorporating regularization, which will be the Follow-the-Regularized-Leader (FTRL) (Abernethy et al., 2008).

1.2.2 Follow-the-Regularized-Leader and its Variants

The Follow-the-Regularized-Leader (FTRL) algorithm (Shalev-Shwartz and Singer, 2006) is an adaptation of the standard FTL method in which the objective is to minimize the accumulated loss over all previous rounds, along with a regularization component. The purpose of this regularization component is to stabilize the solution obtained. Formally, for a regularization function R, a β-strongly convex function with respect to $\|.\|_p$ norm, we define the FTRL algorithm as follow

Algorithm 6 Follow-the-Regularized-Leader

Input: learning rate $\eta > 0$, regularizer function $R(\boldsymbol{f})$, $\boldsymbol{f}_1 = \text{argmin}_{\boldsymbol{f}\in\mathcal{F}} R(\boldsymbol{f})$

Output: next strategy update

$$\boldsymbol{f}_{t+1} = \underset{\boldsymbol{f}\in\mathcal{F}}{\text{argmin}}\, F_{t+1}(\boldsymbol{f}) = \sum_{i=1}^{t} l_i(\boldsymbol{f}) + \frac{R(\boldsymbol{f})}{\eta}$$

For the FTRL algorithm, it is natural that different regularization functions will lead to different algorithms with different regret bounds. In the remaining of this section, we will examine the two most famous variants of FTRL.

When applying FTRL with the Euclidean regularization function

$$R(\boldsymbol{f}) = \frac{1}{2\eta}\|\boldsymbol{f}\|_2^2$$

and the learning space $\mathcal{F} = \mathbb{R}^d$, it is easy to verify that the update strategy will have the simple form

$$\boldsymbol{f}_{t+1} = \boldsymbol{f}_t - \eta \boldsymbol{x}_t.$$

Since $\boldsymbol{x}_t$ is the gradient of l_t, this update rule is also called the Online Gradient Descent (OGD) (Hazan, 2015).

Algorithm 7 Online Gradient Descent

Input: learning rate $\eta > 0$, $\mathcal{F} = \mathbb{R}^d$, loss function $l_t(\boldsymbol{f}) = \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle$

Output: next strategy update

$$\boldsymbol{f}_{t+1} = \boldsymbol{f}_t - \eta \nabla l_t(\boldsymbol{f}_t).$$

Consider $\mathcal{F}_1 := \{\boldsymbol{f} : \|\boldsymbol{f}\| \leq B\}$ and let L be such that $\frac{1}{T}\sum_{i=1}^{T} \|\boldsymbol{x}_i\|_2^2 \leq L^2$. Then by choosing the learning rate $\eta = \frac{B}{L\sqrt{2T}}$ we have the regret bound for OGD [2]

$$R_T(\mathcal{F}_1) := min_{\boldsymbol{f}\in\mathcal{F}_1} R_T(\boldsymbol{f}) \leq BL\sqrt{2T}.$$

[2] For the full proof, see, for example, Theorem 2.4 in Shalev-Shwartz (2012).

The learning rate η in the above analysis depends on the time horizon T. When T is unknown, we can apply the Doubling Trick in Algorithm 3 to derive a similar bound. OGD can also be generalized to different loss functions (i.e., Lipschitz functions) to achieve a sublinear regret bound. More details can be found in Corollary 2.7 in Shalev-Shwartz (2012).

OGD is used in many applications in practice due to its simple strategy update and its near-optimal performance guarantee in the adversarial setting. When apply to the problem of prediction with expert advice (i.e., the action space is a probability simplex $\boldsymbol{f} \in [0,1]^d$), the regret bound of OGD will be $\sqrt{2dT}$ [3]. By choosing a suitable regularization function, FTRL can provide a more efficient bound of $\sqrt{\log(d)T/2}$.

When applying FTRL with the Entropic regularization of the following form

$$R(\boldsymbol{f}) = \sum_{i=1}^{d} \boldsymbol{f}(i) \log(\boldsymbol{f}(i)),$$

it is easy to show that the update strategy of FTRL recovers the famous multiplicative weights update (MWU) algorithm in Algorithm 2. Thus, FTRL with Entropic regularization can enjoy the MWU's optimal regret bound of $O(\sqrt{\log(d)T})$ against an adversary.

While FTRL is a generalization of many famous algorithms (e.g., OGD, MWU), it also inherits an unfavourable property from them: non-last round convergence. More specifically, although the no-regret properties of FTRL can guarantee the convergence to the coarse correlated equilibrium (CCE) of the empirical frequency of play (i.e., a time-average convergence) (Cesa-Bianchi and Lugosi, 2006), it has been shown by Mertikopoulos et al. (2018) that FTRL's behaviour in zero-sum games with an interior equilibrium is Poincaré recurrent, which means that nearly every path repeatedly revisits any (arbitrarily small) starting point neighbourhood. This cycling behaviour is resilient to the agents' choice of regularization method, and it applies to all positive affine transformations of zero-sum games, including strictly competitive games (Adler et al., 2009), even though these changes produce different playing paths. Additionally, the cycling behaviour persists in the context of networked competition (i.e., constant-sum polymatrix games (Cai et al., 2016)).

Apart from learning the convergence dynamic of FTRL, many researchers have tried to improve the performance guarantee (i.e., regret bound) of FTRL-type algorithms. Rakhlin and Sridharan (2013a) studies the setting of online linear optimization in which at round $t+1$, the learner has access to the prediction M_{t+1} of $\boldsymbol{x}_{t+1}$, the behaviour of the environment. Under this setting, Rakhlin and Sridharan (2013a) suggests a new algorithm, Optimistic Follow the Regularized Leader (OFTRL) that can theoretically exploit the extra prediction.

[3] For detail proof, see Corollary 2.13 in Shalev-Shwartz (2012)

Algorithm 8 Optimistic Follow-the-Regularized-Leader

Input: learning rate $\eta > 0$, regularizer function $R(\boldsymbol{f})$, $\boldsymbol{f}_1 = \text{argmin}_{\boldsymbol{f} \in \mathcal{F}} R(\boldsymbol{f})$

Output: next strategy update

$$\boldsymbol{f}_{t+1} = \underset{\boldsymbol{f} \in \mathcal{F}}{\text{argmin}} F_{t+1}(\boldsymbol{f}) = \langle \boldsymbol{f}, \sum_{i=1}^{t} \boldsymbol{x}_i + M_{t+1} \rangle + \frac{R(\boldsymbol{f})}{\eta}$$

With a suitable learning rate, Rakhlin and Sridharan (2013a) shows that the regret bound of the OFTRL algorithm will be

$$O\left(\sqrt{\sum_{t=1}^{T} \|\boldsymbol{x}_t - M_t\|^2}\right).$$

When $M_{t+1} = 0$, OFTRL becomes identical to the FTRL algorithm developed by Abernethy et al. (2008). However, when M_{t+1} is not equal to zero, the algorithm is like predicting the next step and including it in the goal. If the prediction is correct, OFTRL will suffer no regret. One should note here that the no-regret performance in the case of correct prediction is not strong since an agent can achieve a no-dynamic regret performance in this case. Thus, it is natural to develop a new algorithm that can fully exploit the correct prediction while maintaining a no-regret property in the worst-case scenario. This problem will be discussed thoroughly in Chapter 3.

1.2.3 Optimistic Mirror Descent

Chiang et al. (2012); Rakhlin and Sridharan (2013a) study the same setting in OFTRL with predictable sequences and suggests a new algorithm, Optimistic Mirror Descent (OMD), a modification of the famous Mirror Descent algorithm (Nemirovskij and Yudin, 1983). Formally, let R be a 1-strongly convex function with respect to a norm $\|.\|$, and let $D_R(.,.)$ denote the Bregman divergence with respect to R. Let $\|.\|_*$ be dual to $\|.\|$. Let M_{t+1} be the prediction of $\boldsymbol{x}_{t+1}$ at round t. Then the update strategy of OMD follows Algorithm 9. Note that when $M_t = 0$, OMD becomes exactly the same as the famous Mirror Descent in Beck and Teboulle (2003).

The intuition behind OMD is as follows. If at round $t+1$, the agent can play $\hat{\boldsymbol{f}}_{t+1} = \text{argmin}_{\boldsymbol{f} \in \mathcal{F}} F_{t+1}(\boldsymbol{f}) = \eta \langle \boldsymbol{f}, \boldsymbol{x}_{t+1} \rangle + D_R(\boldsymbol{f}, \boldsymbol{g}_{t+1})$, then it is easy to show that a small regret can be achieved. However, in reality, the agent can not observe $\boldsymbol{x}_{t+1}$ before making the decision at round $t+1$. Yet, if the agent can make a close prediction of $\boldsymbol{x}_{t+1}$, then it can expect a small regret from this process.

Algorithm 9 Optimistic Mirror Descent

Input: learning rate $\eta > 0$,
$\boldsymbol{f}_1 = \boldsymbol{g}_1 = \operatorname{argmin}_{\boldsymbol{f} \in \mathcal{F}} R(\boldsymbol{f})$.
Output: next strategy update

$$\boldsymbol{g}_{t+1} = \operatorname*{argmin}_{\boldsymbol{g} \in \mathcal{F}} G_{t+1}(\boldsymbol{g}) = \eta \langle \boldsymbol{g}, \boldsymbol{x}_t \rangle + D_R(\boldsymbol{g}, \boldsymbol{g}_t)$$
$$\boldsymbol{f}_{t+1} = \operatorname*{argmin}_{\boldsymbol{f} \in \mathcal{F}} F_{t+1}(\boldsymbol{f}) = \eta \langle \boldsymbol{f}, M_{t+1} \rangle + D_R(\boldsymbol{f}, \boldsymbol{g}_{t+1})$$

Similar to OFTRL, OMD with a suitable learning rate will have the regret bound of

$$O\left(\sum_{t=1}^{T} \|\boldsymbol{x}_t - M_t\|_*^2\right).$$

In Chapter 3, we will provide a modification of OMD to better exploit the prediction M_t of $\boldsymbol{x}_t$. In particular, the learner can achieve no-forward regret, a stronger notion of performance compared to the conventional no-external regret.

Definition 1.11 (**Forward Regret** Saha et al. (2012))**.** The forward regret is defined as:

$$FR_T := \sum_{t=1}^{T} \left(\langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \right), \text{where } \boldsymbol{g}_{t+1} = \operatorname*{argmin}_{\boldsymbol{g} \in \mathcal{F}} G_{t+1}(\boldsymbol{g}) = \langle \boldsymbol{g}, \sum_{s=1}^{t} \boldsymbol{x}_s + \boldsymbol{x}_{t+1} \rangle + \frac{R(\boldsymbol{g})}{\eta}.$$

The following lemma implies that if an algorithm has no-forward regret property, then it is a no-external regret algorithm as well, but not vice versa [4].

Lemma 1.12. *Let $\boldsymbol{g}_t$ be defined as above, then the following relationship holds for any $\boldsymbol{f} \in \mathcal{F}$:*

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \leq \langle \boldsymbol{f}, \sum_{t=1}^{T} \boldsymbol{x}_t \rangle + \frac{R(\boldsymbol{f})}{\eta}.$$

1.3 Online Markov Decision Processes

In Chapter 4, we extend our results in two-player zero-sum games and online linear optimization to a more general framework, namely Online Markov Decision Processes (OMDPs), in which the agent environment dynamic can be expressed as Figure 1.1. Our focus is on studying OMDPs, where at each round $t \in \mathbb{N}$, the adversary selects the loss function $\boldsymbol{l}_t$ based on the agent's policy history $\{\pi_1, \pi_2, \ldots, \pi_{t-1}\}$. We consider OMDPs with a finite state space S, a finite action set A for the agent at each state, and a fixed transition model P. The agent's starting state, x_1, follows a distribution μ_0

[4] See 3.25 for the proof of this lemma.

over S. Given state $x_t \in S$ at time t, the agent selects an action $a_t \in A$, and moves to a new random state x_{t+1} determined by the fixed transition model $P(x_{t+1}|x_t, a_t)$. The agent simultaneously receives an immediate loss $\boldsymbol{l}_t(x_t, a_t)$, where the loss function $\boldsymbol{l}_t : S \times A \to R$ is bounded in $[0,1]^{|A|\times|S|}$ and is selected by the adversary from a simplex Δ_L. Here, $\{\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_L\}$ are the loss vectors of the adversary, and Δ_L denotes the set of all probability distributions over $\{\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_L\}$. We assume a zero-sum game setting, where the adversary receives the loss of $-\boldsymbol{l}_t(x_t, a_t)$ at round t. We consider the popular full-information feedback (Even-Dar et al., 2009; Dick et al., 2014), where the agent can observe the loss function $\boldsymbol{l}_t$ after each round t. When there is only one state in the process, OMDPs reduce to the previous setting of online linear optimization, which we discuss in Chapter 3.

Several scholars have examined OMDPs in an oblivious environment in which the loss function can be chosen arbitrarily. The algorithm's effectiveness is evaluated through external regret, which is the difference between the overall loss and the optimal stationary policy in hindsight. In Chapter 4, we consider a subclass of non-oblivious environment, the strategic adversary (i.e., the adversary follows a no-(external) regret algorithm to update the loss functions). In the presence of a non-oblivious environment, the formal definition of no-external regret is inadequate as the adversary can adjust to the agent's actions. In our research, we utilize the same methodology as in Arora et al. (2012a) and focus on policy regret. Specifically, the objective of the agent is to minimize policy regret relative to the best-fixed policy in hindsight:

$$R_T(\pi) = \mathbb{E}_{X,A}\left[\sum_{t=1}^{T} \boldsymbol{l}_t^{\pi_t}(X_t, A_t)\right] - \mathbb{E}_{X,A}\left[\sum_{t=1}^{T} \boldsymbol{l}_t^{\pi}(X_t^{\pi}, A_t^{\pi})\right], \tag{1.2}$$

where $\boldsymbol{l}_t^{\pi_t}$ denotes the loss function at time t while the agent follows $\pi_1, \ldots, \pi_T$ and $\boldsymbol{l}_t^{\pi}$ is the adaptive loss function against the fixed policy π of the agent. We say that the agent achieves sublinear policy regret (i.e., no-policy regret property) with respect to the best fixed strategy in hindsight if $R_T(\pi)$ satisfies:

$$\lim_{T\to\infty} \max_{\pi} \frac{R_T(\pi)}{T} = 0.$$

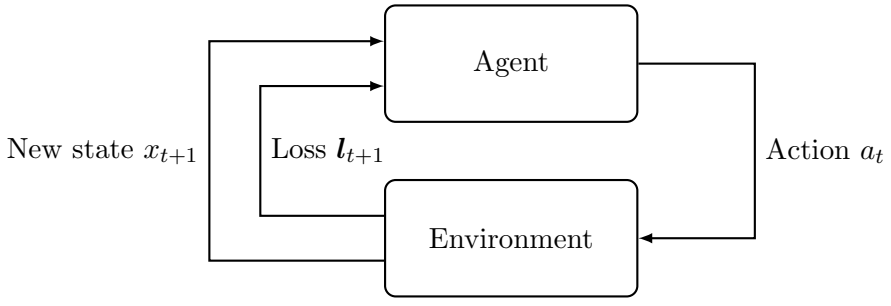


FIGURE 1.1: Agent-Environment Interaction in OMDPs

We use the notation $P(\pi)$ to represent the state transition matrix produced by policy π, where $P(\pi)_{s,s'} = \sum_{a \in A} \pi(a|s) P^a_{s,s'}$. Throughout our work, we assume the mixing time assumption, which is a commonly accepted assumption in OMDPs according to various prior studies (Even-Dar et al., 2009; Dick et al., 2014; Neu et al., 2013):

Assumption 1 (Mixing time). There exists a constant $\tau > 0$ such that for all distributions $\boldsymbol{d}$ and $\boldsymbol{d}'$ over the state space, any policy π,

$$\left\| \boldsymbol{d} P(\pi) - \boldsymbol{d}' P(\pi) \right\|_1 \leq e^{-1/\tau} \left\| \boldsymbol{d} - \boldsymbol{d}' \right\|_1,$$

where $\|\boldsymbol{x}\|_1$ denotes the l_1 norm of a vector $\boldsymbol{x}$.

Let $\boldsymbol{v}_t^\pi(x, a)$ represent the probability of pair (x, a) at time step t when following policy π with initial state x_1. According to the mixing time assumption, for any initial states, $\boldsymbol{v}_t^\pi$ will eventually converge to a stationary distribution $\boldsymbol{d}_\pi$ as t approaches infinity. We can denote the stationary distribution set from all agent's deterministic policies as $\boldsymbol{d}_\Pi$. When an agent follows an algorithm A that utilizes $\pi_1, \pi_2, \ldots$ at each time step, we use $\boldsymbol{v}_t(x, a) = \mathbb{P}\left[X_t = x, A_t = a\right]$ and $\boldsymbol{d}_t = \boldsymbol{d}_{\pi_t}$, despite some minor misuse of notation. As a result, the regret in Equation (1.2) can be formulated as

$$R_T(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} \left\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{v}_t \right\rangle\right] - \mathbb{E}\left[\sum_{t=1}^{T} \left\langle \boldsymbol{l}_t^{\pi}, \boldsymbol{v}_t^{\pi} \right\rangle\right].$$

The mixing time assumption allows us to define the average loss of policy π in an online MDP with loss $\boldsymbol{l}$ as $\eta_{\boldsymbol{l}}(\pi) = \langle \boldsymbol{l}, \boldsymbol{d}_\pi \rangle$ and the accumulated loss $Q_{\pi,\boldsymbol{l}}(s, a)$ is defined as

$$Q_{\pi,\boldsymbol{l}}(s, a) = \mathbb{E}\left[\sum_{t=1}^{\infty} \left(\boldsymbol{l}(s_t, a_t) - \eta_{\boldsymbol{l}}(\pi)\right) \middle| s_1 = s, a_1 = a, \pi\right].$$

In the next sections, we introduce two families of algorithms in the case of the oblivious adversary. The understanding of these algorithms is fundamental in developing new algorithms against non-oblivious strategy in Chapter 4.

1.3.1 Markov Decision Process-Expert

Algorithm 10 Markov Decision Process-Expert (MDP-E)

1: **Input:** Expert algorithm B_s (i.e., MWU) for each state
2: **for** $t = 1$ to ∞ **do**
3: Using algorithm B_s with set of expert A and the feedback $Q_{\pi_t,\boldsymbol{l}_t}(s, .)$ for each state s
4: Output π_{t+1} and observe $\boldsymbol{l}_{t+1}$
5: **end for**

In the OMDPs setting, the agent needs to choose a policy from a finite set of fixed policies in each round of play. Thus, a naive approach could be applying a standard no-regret algorithm (e.g., MWU) to the set of all fixed policies and following this algorithm. However, there are several drawbacks to this approach. Firstly, we are dealing with a huge number of fixed policies. Specifically, for an MDP with state space S and action space A, the number of fixed policies is $|A|^{|S|}$, which renders the naive no-regret's approach computationally infeasible. Secondly, using no-regret algorithms comes with another issue: the policy's current reward is influenced by past actions, which is not typical in standard settings. Additionally, when applying typical regret algorithms, the regret bounds are usually logarithmic in the number of policies, meaning they have a linear relationship with the number of states. It would be beneficial to have a more efficient regret bound that does not depend on the state space's size, which is usually large. Markov Decision Process-Expert (MDP-E) (Even-Dar et al., 2009) resolves the above problem by efficiently incorporating the benefits of existing no-regret algorithms into a more adversarial reinforcement learning setting, in which the environment could change obliviously over time.

The idea of MDP-E (Even-Dar et al., 2009) comes from the observation of the performance difference lemma: the global regret of the agent can be decomposed into local regret in each state with appropriate feedback. Formally, the performance difference between two stationary distributions of strategy π and π^* against the loss function $\boldsymbol{l}$ can be expressed as

$$\langle \boldsymbol{l}, \boldsymbol{d}_{\pi} \rangle - \langle \boldsymbol{l}, \boldsymbol{d}_{\pi^*} \rangle = \mathbb{E}_{s \in \boldsymbol{d}_{\pi^*}} [Q_{\pi, \boldsymbol{l}}(s, \pi) - Q_{\pi, \boldsymbol{l}}(s, \pi^*)].$$

Therefore, summing up the above inequalities l_t and d_{π_t} for $t = 1, \ldots, T$ we have

$$\sum_{t=1}^{T} \langle \boldsymbol{l}_t, \boldsymbol{d}_{\pi_t} \rangle - \langle \boldsymbol{l}, \boldsymbol{d}_{\pi^*} \rangle = \sum_{s \in S} \boldsymbol{d}_{\pi^*}(s) \sum_{t=1}^{T} [Q_{\pi_t, \boldsymbol{l}_t}(s, \pi_t) - Q_{\pi_t, \boldsymbol{l}_t}(s, \pi^*)].$$

Therefore, by bounding the regret in each state s with the appropriate feedback $Q_{\pi_t, \boldsymbol{l}_t}(s, \pi_t)$, the agent can bound the regret with respect to the stationary distribution.

Leveraging on the performance difference lemma, MDP-E (Even-Dar et al., 2009) follows a no-regret algorithm (e.g., MWU) in each state with the feedback $Q_{\pi_t, \boldsymbol{l}_t}(s, \pi_t)$ so that it can bound the regret with respect to the stationary distribution. Furthermore, in order to quantify the difference between the actual performance and the stationary distribution, the no-regret algorithm used in MDP-E needs to have another property: slowly updating the strategy profile (for example, MWU and FTRL satisfy this property). Therefore, MDP-E with a suitable no-regret algorithm will lead to a sublinear regret bound (Even-Dar et al., 2009) of

$$O\left(\sqrt{\tau^4 T \log(|A|)}\right).$$

Despite the strong sublinear regret, one limitation of MDP-E is that it requires the exact action-value function $Q_{\pi,\boldsymbol{l}}$ in order to update its strategy. Therefore, the algorithm will be inefficient when the action-value function is hard or expensive to calculate. In order to resolve that, Abbasi-Yadkori et al. (2019); Cai et al. (2020) use least squares policy evaluation (LSPE) to estimate the action-value function $Q_{\pi,\boldsymbol{l}}$, creating new algorithms with more practical use.

While MDP-E can work well with function approximation for the action-value function, extending it with some constraints such as uncertainties in the transition model will be challenging due to the nature of the algorithm. In the next section, we introduce another class of OMDPs algorithm that can effectively handle model constraints and further extensions.

1.3.2 Online Relative Entropy Policy Search

Compared to the MDP-E line of work, the Online Relative Entropy Policy Search (O-REPS) algorithm (Zimin and Neu, 2013; Dick et al., 2014) improves on the suboptimal dependency of the regret bound on the mixing time parameter τ. To achieve this, O-REPS views the OMDP problem as an online linear optimization problem and demonstrates that the resulting methods can be implemented efficiently. The first observation in O-REPS is that the set of all stationary distribution $\boldsymbol{d}_\Pi$ is a convex polytope in $\mathbb{R}^{|U|}$ and can be expressed as a set of linear constraints (U denotes all the (state, action) pair so $|U| = |A| \times |S|$):

$$\boldsymbol{d}_\Pi = \left[\boldsymbol{d} \in [0,1]^{|U|} : \sum_{u \in U} \boldsymbol{d}(u) = 1, \sum_{a \in A(s)} \boldsymbol{d}(s,a) = \sum_{u \in U} \boldsymbol{d}(u) P(s|u), s \in S \right].$$

Then by applying an online linear optimization algorithm with the loss function $\langle \boldsymbol{l}_t, \boldsymbol{d}_{\pi_t} \rangle$ and the convex decision set $\boldsymbol{d}_\Pi$, O-REPS can have a sublinear regret bound with respect to the stationary distribution. The remaining job is to derive the connection between the actual regret and the regret of the stationary distribution, which follows (see Lemma 1 in Dick et al. (2014) for the full proof)

$$\mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{l}_t(X_t, A_t)\right] - \mathbb{E}_{X,A}\left[\sum_{t=1}^{T} \boldsymbol{l}_t(X_t^\pi, A_t^\pi)\right] \leq \sum_{t=1}^{T} \mathbb{E}[\langle \boldsymbol{l}_t, \boldsymbol{d}_{\pi_t} - \boldsymbol{d}_\pi \rangle] + T(\tau+1)k + 4\tau + 4,$$

for any $k \geq \mathbb{E}[\|\boldsymbol{d}_{\pi_t} - \boldsymbol{d}_{\pi_{t-1}}\|]$, $t = 2, \dots T$.

Given that it is possible to retrieve a policy from a stationary distribution, it is sufficient to identify a sequence of $\boldsymbol{d}_1, \dots \boldsymbol{d}_T \in \boldsymbol{d}_\Pi$ that changes gradually and leads to a small initial value for the bound. Using the famous Mirror Descent with approximate projections (Dick et al., 2014), O-REPS can achieve the following regret bound in the

full information setting

$$O(\sqrt{\tau T \log(|A|)}),$$

which is a substantial improvement in the mixing time dependency compared to the MDP-E algorithm (Even-Dar et al., 2009). Another improvement of O-REPS compared to MDP-E is that it can easily accommodate uncertainties in the transition model by extending the convex decision set $\boldsymbol{d}_\Pi$ (see for example Rosenberg and Mansour (2019)). In Chapter 4, we focus more on extending our result in the MDP-E line of work and leave the extension on O-REPS as an important future work.

1.4 Structure of the Thesis and Contributions

This thesis contains three main chapters that consider the problem of learning against the strategic adversary under different settings: Two-player Zero-Sum Games, Online Linear Optimization, and Online Markov Decision Processes. Together, we extensively derive different theoretical and empirical results that broaden our understanding of the strategic adversary. Since our goal is to create a cohesive thesis while ensuring that each chapter remains independent, we reintroduce some notations and definitions with minor modifications in each chapter to align with the specific context. Our hope is that this approach will help readers fully grasp our contributions. Figure 1.2 summarizes our contributions in each setting.

Firstly, in Chapter 2, we consider the two-player zero-sum games setting and develop a new algorithm, Last Round Convergence in Asymmetric algorithm (LRCA), a no-dynamic regret algorithm that leads to last round convergence against the strategic

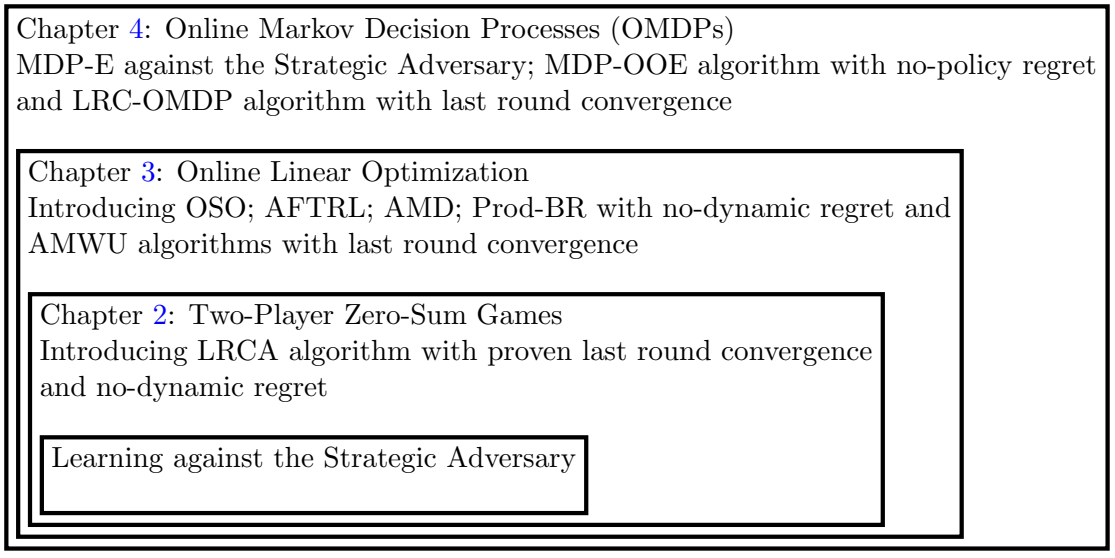


FIGURE 1.2: Contributions of our thesis

adversary. This is the first algorithm to achieve both last round convergence and no-dynamic regret against the strategic adversary in the literature.

Secondly, in Chapter 3, we consider a more general setting compared to the two-player zero-sum games: online linear optimization. In this setting, we further exploit the property of the strategy adversary and propose Online Single Oracle, a combination of no-external regret algorithms and double oracle from game theory that achieves a better external regret. Furthermore, we develop three online learning algorithms, namely Accurate Follow the Regularized Leader (AFTRL), Accurate Mirror Descent (AMD) and Prod-Best Response (Prod-BR), that can intensively exploit the extra knowledge from playing against the strategic adversary. Our algorithms are the first to consider forward regret and achieve O(1) regret against the strategic adversary. A special case of AFTRL, Accurate Multiplicative Weights Update (AMWU) leads to last round convergence in the self-play setting with a better rate compared to state-of-the-art algorithms (e.g., MWU, OMWU).

Thirdly, in Chapter 4, we study online Markov decision processes (OMDPs), a general setting that covers many practical applications. Under this setting, we first show that MDP-E is a no-policy regret algorithm against the strategic adversary. In real-world games where the support size of a NE is small, we introduce MDP-Online Oracle Expert (MDP-OOE) which provably achieves a better policy regret compared to MDP-E. In convergence dynamic analysis, we show that our algorithm, Last-Round Convergence in OMDPs (LRC-OMDP), achieves last round convergence to a NE against the strategic adversary. Finally, Chapter 5 concludes the work and proposes several directions for future research.

Chapter 2

Last Round Convergence to NE Against Strategic Adversary

This chapter is dedicated to investigating the problem of learning against a strategic adversary in the context of repeated two-player zero-sum games. Specifically, we consider the scenario in which the row player (i.e. the strategic adversary) employs a no-regret algorithm to minimize her regret while repeatedly playing the game, and focus on developing a no-dynamic regret algorithm for the column player to achieve last round convergence to a minimax equilibrium. Our proposed algorithm is designed to efficiently handle a broad range of popular no-regret algorithms that the row player may use, including the multiplicative weights update algorithm, general follow-the-regularized-leader, and any no-regret algorithms that satisfy the stability property. Our analysis demonstrates the effectiveness of our approach, making significant contributions to the literature on learning against strategic adversaries in the two-player zero-sum games setting.

2.1 Introduction

Repeated two-player zero-sum games form one of the most studied classes of repeated games in game theory. In this setting, thanks to Blackwell's famous approachability theorem, if a player's strategies are generated by algorithms (i.e., policies) with a special property called "no-regret", one can prove that, on average, that player does not perform worse than the best-fixed strategy in hindsight. A direct implication of this result is that if both players choose to play such no-regret algorithms, their average payoffs will converge to the game's minimax value. Put differently, the players' strategies will converge to a minimax equilibrium on average (see e.g., Cesa-Bianchi and Lugosi (2006) or Arora et al. (2012b) for more details). It can also be easily shown that this (on-average) convergence holds independently from the prior information that each player has about the payoff matrix $\boldsymbol{A}$. That is, no matter how much prior information a player

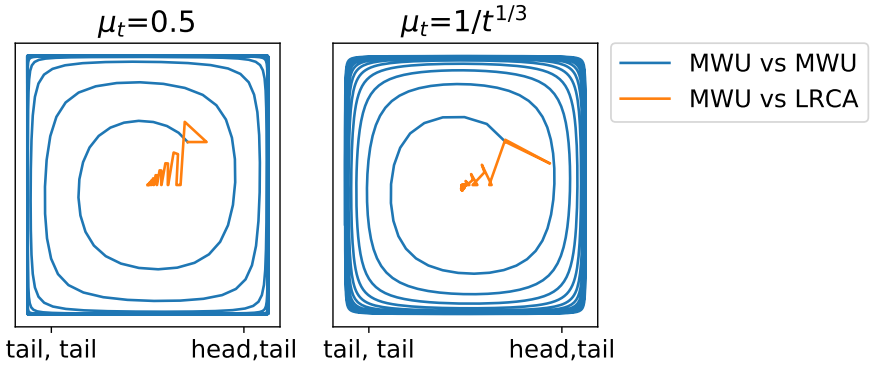


FIGURE 2.1: Player Strategies Spiraling Outwards In MWU vs Last Round Convergence in LRCA in Matching Pennies after 2500 iterations with the Same Initial Condition.

has about the game, she cannot exploit the other player's average payoff if the latter uses a no-regret algorithm.

In this chapter, we consider a shift of interest for the column player and investigate whether she can achieve that in the repeated two-player zero-sum games setting. Along with optimizing the usual performance measure (i.e., no-regret property), the column player also wants to keep her strategy stable while repeatedly play the game. This is motivated by the fact that changing strategies through repeated games might be undesirable. For example, changing the (mixed) strategy of a company will increase the cost of operation to implement the new mixed strategy (e.g., as a result of having to hire new equipment and employees). Therefore, the company often aims not only to maximize the revenue (i.e., the average payoff) but also to reduce the cost of operation by having a stable strategy. For another example, consider a government-owned company, for whom, along with the average benefit, keeping the market stable is one of the key goals in order to increase social welfare. Finally, in system design, the designer (the column player) will want the participant (the row player) to play a certain strategy so that the system is well-behaved.

In the online learning literature, maximizing the average payoff and achieving the system's stability are often viewed as conflicting goals. That is, if all the player in a system follows a selfish behaviour (e.g., an FTRL no-regret algorithm) to maximize their payoff, then the dynamic of the system could become chaotic, and last round convergence never happens (see, e.g., Mertikopoulos et al. (2018) for more details). Figure 2.1 demonstrates a simple game of matching pennies where the dynamic of two selfish players using Multiplicative Weight Update(MWU)(i.e., the blue trajectory) leads to outwards spirals with different step size μ. Thus, the Nash equilibrium point(i.e., the centre point) can never be achieved in this situation. The question is, whether there is a way to achieve both no regret and stability in a system.

In this chapter, we show that it is possible to exploit strategic adversaries to achieve both stability and no-dynamic regret; that is, the regret compared to the best action of each round. In the general fully adversarial setting, it is impossible to achieve no-dynamic regret property and therefore the state-of-the-art algorithms instead measure the success by comparing the regret with the best fixed strategy in the hindsight. By looking deeper into the adversarial setting and analysing the behaviour of strategic players, one can achieve a much stronger concept of regret: dynamic regret.

The intuition behind this result can be explained as follows: If the row player believes that the goal for both players is to maximize their average payoffs (i.e., a fully adversary setting since the column player tries minimise the payoff of the row player in zero-sum game), then she will typically choose to play a no-regret "type" algorithm to achieve good average performance. Being aware of this, the column player can now choose an algorithm that exploits this information to have no-dynamic regret and last round convergence. We should note here, however, that it is not trivial how this can be efficiently done. For example, if the column player keeps playing the same strategy (i.e., the minimax equilibrium), then while the system might achieve stability as the strategy of the row player will converge to the best response, this is not a no-regret algorithm and therefore, far away from being a no-dynamic regret algorithm.

Motivated by the abovementioned challenge, we propose a new algorithm that achieves no-dynamic regret for the column player in the case the row player is a strategic adversary. In contrast to normal no-regret algorithms which take best fixed strategy as the milestone, dynamic regret (e.g., see Besbes et al. (2015)) compares the regret with the optimal strategy in the hindsight. Thus, dynamic regret is a much stronger concept than normal regret, especially in situation where every fixed strategy perform poorly in the game. In adversarial setting, we show that one player can leverage the other strategic player's behaviour to achieve a no-dynamic regret algorithm. In the general case, we introduce a method for the column player to have no-regret property against random strategies of the row player while still maintaining no-dynamic regret property against a strategic row player.

Furthermore, while on-average convergence has been extensively studied, it is still an open question whether last round convergence can be achieved, especially when the row player is also playing a no-regret algorithm (see Section 2.2 for more details). Against this background, we show that our algorithm, called the *L*ast *R*ound *C*onvergence in *A*symmetric games (LRCA), provably achieves last round convergence to a minimax equilibrium of the corresponding game. As shown in Figure 2.1, our Last Round Convergence Algorithm (i.e., the orange trajectory) converges to the Nash equilibrium of the Matching Pennies game while playing against the MWU with different step size μ. We prove that in our setting if the column player follows LRCA and the row player follows an algorithm from a wide set of common no-regret algorithms, then last round convergence to the minimax equilibrium of the game can be achieved. Note that in the case

the horizon of play is unknown, the row player needs to employ a decreasing learning rate to make the algorithm no-regret. It means that the new observation feedback will be discounted compare to the old feedback. Lin et al. (2020) argues that this discounted new feedback is counter intuitive and unjustifiable from economic principles. Thus, it is important to consider the no-regret learning dynamic where the learner does not impose decreasing step size. In this chapter, we allow the row player to play different type of algorithms, including μ-regret algorithms (i.e., constant step size) in which even the average convergence in self-play is not yet known.

Overall this chapter has two main contributions. **First**, by allowing different strategies between the column and row player, in Section 2.5, we propose an algorithm that leads to last round convergence in many situations, which were proven not to hold (i.e., there is no last round convergence) in symmetric information settings. **Second**, we show that by using the algorithm, the column player can achieve no-dynamic regret property; see Section 2.6 for more details. This answer the question of how to achieve both maximizing the average payoff and stability against the strategic adversary in a repeated game.

2.2 Related Work

It is well-known that if both players use no-regret algorithms, their average strategies converge to a minimax equilibrium with the convergence rate of $\mathcal{O}(T^{-1/2})$; cf. Freund and Schapire (1999). Daskalakis et al. (2011) and Rakhlin and Sridharan (2013b) have further improved this result by developing no-regret algorithms with a near-optimal convergence rate of $\mathcal{O}(\frac{\log(T)}{T})$. However, despite the extensive literature on no-regret algorithms, these algorithms typically provide on-average convergence only, but not last round convergence. For example, Bailey and Piliouras (2018) proved that in games with an interior Nash equilibrium point, if the players use the multiplicative weights update (MWU) algorithm, then the last round strategy converges to the boundary. In addition, Mertikopoulos et al. (2018) showed that by using regularized learning, the system's behaviour is Poincaré recurrent; that is, there is a loop in the strategy dynamics of the players. This undesirable feature causes many issues in game theory and applications, including unwanted cyclic behaviour in training Generative Adversarial Networks (GANs). Thus, a learning dynamic leading to last round convergence is of importance in the development of the field (see, e.g., Daskalakis et al. (2018) for more details). Note that in a recent paper, Daskalakis and Panageas (2019) proved that if both players use the optimistic multiplicative weights update algorithm (OMWU), then we have last round convergence to the minimax equilibrium if this equilibrium point is unique. This last round convergence result also requires another restrictive assumption, namely: The constant step size of the update mechanism has to be calculated from the payoff matrix $\boldsymbol{A}$ of the game. Therefore, if the row player does not know the matrix $\boldsymbol{A}$ of the game, then OMWU cannot guarantee last round convergence (as it requires both players to

know matrix $\boldsymbol{A}$). Besides, if the row player plays different no-regret algorithms such that MWU or FTRL, which are widely used in many applications, then OMWU cannot lead to the last round convergence either. This raises the question of whether there could be a robust algorithm, when playing against different no-regret algorithms, converging at the last round to minimax equilibrium.

2.3 Key Assumptions

To proceed with the development of this chapter, we make the following two assumptions:

1. The column player can get an arbitrarily close estimation of her minimax equilibrium.
2. The row player is a strategic adversary (i.e. follows a no-regret "type" algorithm).

The rationale of these assumptions can be explained as follows: Assumption 1 may arise from asymmetric information two-player zero-sum games in which the column player knows the matrix $\boldsymbol{A}$ of the game. In this case, the column player can calculate the exact minimax equilibrium using linear programming. Realistic examples for this setting include problems from the security games domain, where an attacker can store the feedback from past observations and analyze the behaviour of the system. Thus, the attacker could know the matrix $\boldsymbol{A}$ of the game. Another example comes from the perspective of a new company that enters an existing business market. In this market, every strategy and payoff of the players are revealed. Therefore, when a new company enters the market, it can anticipate what the payoff for a particular action of its strategies is. Thus, the new incomer knows the matrix $\boldsymbol{A}$ of the game. Note that the asymmetric game assumption might appear in many other applications, and hence we argue that this setting deserves attention from the online learning research community.

In the case of symmetric information games (i.e., both players have the same prior information about the game), if the row player follows a no-regret type algorithm, the column player can first use a no-regret algorithm to estimate the minimax equilibrium. Note that in this estimation phase, the column player cannot guarantee the no-dynamic regret property like in the case of an asymmetric game. See Section 2.5.5 for more details.

Assumption 2 comes from the vanilla property of no-regret algorithms: without prior information, a player will not do worse than the best-fixed strategy in hindsight by following a no-regret algorithm. In this chapter, we allow the row player to deviate from a no-regret algorithm in a certain way; that is, she can choose a fixed learning rate.

We also consider the full information feedback (see, e.g., Bailey and Piliouras (2018), Daskalakis et al. (2011), Freund and Schapire (1999)).[1]

Note that our setting differs from Daskalakis and Panageas (2019) in the following ways: we require neither the knowledge of the update step size nor the uniqueness of the minimax equilibrium. In addition, our result does not require the row player to follow the fixed learning rate OMWU. As such, we argue that our result can be applied to more real-world applications, due to its more reasonable and realistic assumptions (see Section 2.3 for more detailed discussions).

2.4 Problem Formulations & Preliminaries

Consider a repeated two-player zero-sum game. This game is described by a $n \times m$ payoff matrix $\boldsymbol{A}$ and w.l.o.g we assume the entries of $\boldsymbol{A}$ in $[0, 1]$. The rows and columns of $\boldsymbol{A}$ represent the *pure* strategies of the *row* and *column* players, respectively. We define the set of feasible strategies of the row player, at round t, by $\Delta_n := \{\boldsymbol{x}_t \in \mathbb{R}^n \mid \sum_{i=1}^n \boldsymbol{x}_t(i) = 1,\ \boldsymbol{x}_t(i) \geq 0\ \forall i \in \{1, ..., n\}\}$. The set of feasible strategies of the column player, denoted by Δ_m, is defined in a similar way. At round t, if the row (resp. column) player chooses a mixed strategy $\boldsymbol{x}_t \in \Delta_n$ (resp. $\boldsymbol{y}_t \in \Delta_m$), then the row player's payoff is $-\boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t$, while the column player's payoff is $\boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t$. Thus, the row (resp. column) player aims to minimise (resp. maximise) the quantity $\boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t$ (resp. $\boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t$). John von Neumann's minimax theorem (Neumann (1928)), founding stone in zero-sum games states that

$$\max_{\boldsymbol{y}\in\Delta_m} \min_{\boldsymbol{x}\in\Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} \quad = \quad \min_{\boldsymbol{x}\in\Delta_n} \max_{\boldsymbol{y}\in\Delta_m} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} \quad = \quad v \tag{2.1}$$

for some $v \in \mathbb{R}$. We call a point $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ satisfying the minimax theorem Equation (2.1) *the minimax equilibrium of the game*. Throughout this chapter, we use the notation $f(\boldsymbol{x}) := \max_{\boldsymbol{y}\in\Delta_m} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}$. Since $\boldsymbol{A}$ is a non-zero matrix with entries in $[0, 1]$, we have $f(\boldsymbol{x}) \geq 0$. Note that $(\boldsymbol{x}, \boldsymbol{y}^*)$ which satisfy $f(\boldsymbol{x}) - v \leq \epsilon$ are ϵ-Nash equilibria (i.e., $max_{\boldsymbol{y}\in\Delta_m} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} - \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} \leq \epsilon$ and $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} - min_{\boldsymbol{x}\in\Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} \leq \epsilon$) and $\epsilon = 0$ implies $\boldsymbol{x}_l$ is the Nash equilibrium of the row player. Similarly, if $\min_{\boldsymbol{x}\in\Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} = v$, then $\boldsymbol{y}$ is also a minimax equilibrium strategy. Next, we define the concept of a *no-dynamic regret* that will play an important role in this chapter.

Definition 2.1 (Besbes et al. (2015))**.** Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ be a sequence of mixed strategies played by the row player. An algorithm of the column player that generates a sequence

[1] Note that the main focus of this chapter is on the investigation of the benefit of having asymmetric information. Thus, the analysis of other feedback cases, such as bandit or semi-bandit, is out of scope and remains part of future work.

of mixed strategies $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ is called a *no-dynamic regret* algorithm if we have

$$\lim_{T\to\infty} \frac{DR_T}{T} = 0, \quad \text{where} \quad DR_T := \sum_{t=1}^{T} \left(\max_{\boldsymbol{y}\in\Delta_m} \boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y} - \boldsymbol{x}_t^\top \boldsymbol{A}\boldsymbol{y}_t \right).$$

Here, the no-dynamic regret property is a stronger notion, compared to the usual no-regret property, as the latter, defined by $R_T = \max_{\boldsymbol{y}\in\Delta_m} \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{x}_t{}^\top \boldsymbol{A}(\boldsymbol{y}-\boldsymbol{y}_t)$, is benchmarked against the best-fixed strategy in hindsight.

Note that no-dynamic regret is typically impossible to achieve with current state-of-the-art algorithms in the adversarial symmetric setting. We will show that in our setting we can design an algorithm that can achieve the no-dynamic regret property.

Finally, it is important to mention that in this chapter, we will use the Kullback-Leibler divergence to understand the behaviour of the row player's strategies.

Definition 2.2 (Kullback and Leibler (1951))**.** The relative entropy or K-L divergence between two vectors X_1 and X_2 in Δ_n is defined as

$$RE(X_1 \| X_2) = \sum_{i=1}^{n} X_1(i) \log \left(\frac{X_1(i)}{X_2(i)} \right).$$

The Kullback-Leibler divergence is always non-negative. Furthermore, from Gibbs's inequality (Mitrinovic and Vasic (1970)) we can show that $RE(X_1 \| X_2) = 0$ if and only if $X_1 = X_2$ almost everywhere.

2.5 Last Round Convergence to Minimax Equilibrium

We first start with the investigation of last round convergence in asymmetric information cases. In particular, we present the Last Round Convergence of Asymmetric games (LRCA) algorithm for the column player. We then show that our algorithm is robust to many no-regret algorithms that can be played by the row player, namely: MWU/LMWU, general FTRL and stable no-regret algorithms(i.e., it provides last round convergence when played against these algorithms). Under Assumption 1, we first study the case where the column player knows the exact minimax equilibrium $\boldsymbol{y}^*$ and the value v of the game (i.e. the column player knows the matrix $\boldsymbol{A}$ of the game). We then consider the case where only estimation of $\boldsymbol{y}^*$ and v are available to the column player in Section 2.5.5.

For a sequence of strategies $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ played by the row player, the LRCA algorithm (see Algorithm 11) for the column player can be described as follows: At each odd round, the column player plays the minimax equilibrium strategy, $\boldsymbol{y}^*$, so that in the next round,

she can not only predict the distance between the current strategy of the row player and a minimax equilibrium but also prevent the row player from deviating the current strategy. Then, at the following even round, the column player chooses a strategy such that the feedback to the row player, $\boldsymbol{A}\boldsymbol{y}_t$, is a direction towards a minimax equilibrium strategy of the row player. Depending on the distance between the current strategy of the row player and a minimax equilibrium (which is measured by $f(\boldsymbol{x}_{t-1}) - v$), the column player chooses a suitable step size α_t so that the strategy of the row player will approach a minimax equilibrium. Note here that β is a constant number and we can fix $\beta = n^2$ so that our LRCA algorithm is robust against different no-regret algorithms that we consider in this chapter. In order to obtain a tighter regret convergence rate, we choose two different β in the case of MWU/LMWU and FTRL.

Algorithm 11 (LRCA) will work for a large set of learning rates, including the constant learning rate case. Simpler algorithms, such that "fictitious play" or "best response to the last feedback" will fail to converge in the simple case of constant learning rate and do not have the no-dynamic regret property in Section 2.6.

In Algorithm 11, every odd round the column player keeps playing the same strategy $\boldsymbol{y}^*$ and thus the row player can realize and exploit this pattern. To avoid this scenario, the column player can randomly choose two successive strategies such that: $\boldsymbol{y}_{2k-1} + \boldsymbol{y}_{2k} = \boldsymbol{y}^* + (1 - \alpha_{2k})\boldsymbol{y}^* + \alpha_{2k}\boldsymbol{e}_{2k}$ where α_{2k} and $\boldsymbol{e}_{2k}$ are chosen according to Algorithm 11. By following this method, the cumulative feedback received by the row player in the odd round will stay the same and thus the analysis of Algorithm 11 remains correct. We will prove in the following subsections that if the column player follows LRCA and the row player uses one of the aforementioned no-regret algorithms, last round convergence to the minimax equilibrium will be achieved.

Algorithm 11 *L*ast *R*ound *C*onvergence in *A*symmetric algorithm (LRCA)

Input: Current iteration t, past feedback $\boldsymbol{x}_{t-1}^\top \boldsymbol{A}$ of the row player
Output: Strategy $\boldsymbol{y}_t$ for the column player
if $t = 2k-1,\ k \in \mathbb{N}$ **then**
  $\boldsymbol{y}_t = \boldsymbol{y}^*$
else if $t = 2k,\ k \in \mathbb{N}$ **then**
  $\boldsymbol{e}_t := \operatorname{argmax}_{e \in \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots \boldsymbol{e}_m\}} \boldsymbol{x}_{t-1}^\top A\boldsymbol{e}$; $f(\boldsymbol{x}_{t-1}) := \max_{\boldsymbol{y} \in \Delta_m} \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}$
  $\alpha_t := \frac{f(\boldsymbol{x}_{t-1}) - v}{\beta}$
  $\boldsymbol{y}_t := (1 - \alpha_t)\boldsymbol{y}^* + \alpha_t \boldsymbol{e}_t$
end if

2.5.1 No-Regret Algorithms with Stability Property

We first prove that LRCA can work with a general class of no-regret algorithms, should the no-regret algorithm of the row player have a "stability" property defined below:

Definition 2.3. A no-regret algorithm is *stable* if $\forall t : \boldsymbol{y}_t = \boldsymbol{y}^* \implies \boldsymbol{x}_{t+1} = \boldsymbol{x}_t$.

The stability property is natural and can be explained as follows. Intuitively, in a good situation where the column player follows $\boldsymbol{y}^*$ and all the rewards $\boldsymbol{A}\boldsymbol{y}^*$ are equal (i.e., $\boldsymbol{A}\boldsymbol{y}^* = [v, v, \ldots, v]$), the row player's all strategy will provide the same reward and there is no incentive to change the current strategy. Thus, the next strategy of the row player will be the same as the current strategy (i.e., $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$) and the column player can use this information to exploit the row player in the next iteration.

There are many no-regret algorithms with the stability property. For example, we prove below that a wide class of no-regret algorithms, Follow The Regularized Leader (FTRL), are stable:

Lemma 2.4. *Suppose the row player follows an FTRL algorithm with regularizer $R(\boldsymbol{x})$ defined as:*

$$\boldsymbol{x}_t = \operatorname*{argmin}_{\boldsymbol{x} \in \Delta_n} \mu \boldsymbol{x}^\top \left(\sum_{i=1}^{t-1} \boldsymbol{A}\boldsymbol{y}_i \right) + R(\boldsymbol{x}).$$

If there exists a fully-mixed minimax equilibrium strategy for the row player, then FTRL is stable.

Proof. As there exists a fully-mixed minimax equilibrium strategy for the row player, we have $\boldsymbol{A}\boldsymbol{y}^* = [v, \ldots, v]^T$. Thus, we have

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}^* = v \quad \forall \boldsymbol{x} \in \Delta_n.$$

When the column player follows the minimax strategy, the minimization for $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$ only differ by a constant term v, so their solutions are the same. □

Note that the FTRL framework, with appropriately chosen $R(\boldsymbol{x})$, can recover many popular no-regret algorithms, such as online mirror descent, multiplicative weights update, and Hannan's algorithm (a.k.a. Follow the Perturbed Leader). See, e.g., (McMahan, 2011; Arora et al., 2012b) for more details. Following the stability property, we prove the LRCA can lead to ϵ-Nash equilibrium, where ϵ can be chosen arbitrarily small:

Theorem 2.5. *Assume that the row player follows a stable no-regret algorithm and n is the row player's strategy dimension. Then, by following LRCA, for any $\epsilon > 0$, there exists $l \in \mathbb{N}$ such that $\frac{\mathcal{R}_l}{l} = \mathcal{O}(\frac{\epsilon^2}{n})$ and $f(\boldsymbol{x}_l) - v \leq \epsilon$.*

Proof. We will prove the theorem by contradiction. Suppose there exists $\epsilon > 0$ such that:

$$f(\boldsymbol{x}_l) - v > \epsilon, \ \forall l \in \mathbb{N}.$$

Then, following the update rule of Algorithm 1 (LRCA) we have:

$$\boldsymbol{y}_{2k-1} = \boldsymbol{y}^* \; ; \; \alpha_{2k} = \frac{f(\boldsymbol{x}_{2k-1}) - v}{\beta} > \frac{\epsilon}{\beta}.$$

By the stability property, as $\boldsymbol{y}_{2k-1} = \boldsymbol{y}^*$, we then have: $\boldsymbol{x}_{2k-1} = \boldsymbol{x}_{2k}$. Following the update rule of Algorithm 1 (LRCA):

$$\begin{aligned}
\boldsymbol{x}_{2k}^T \boldsymbol{A} \boldsymbol{y}_{2k} &= \boldsymbol{x}_{2k-1}^T \boldsymbol{A} \left((1 - \alpha_{2k}) \boldsymbol{y}^* + \alpha_{2k} \mathbf{e}_{2k} \right) \\
&\geq (1 - \alpha_{2k}) v + \alpha_{2k} f(\boldsymbol{x}_{2k-1}) \qquad (2.2a) \\
&> (1 - \alpha_{2k}) v + \alpha_{2k} (v + \epsilon) \qquad (2.2b) \\
&\geq v + \frac{\epsilon^2}{\beta},
\end{aligned}$$

Where inequality (2.2a) is due to

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{y}^* \geq v \; \forall \boldsymbol{x} \in \Delta_n,$$

and where inequality (2.2b) comes from the assumption that $f(\boldsymbol{x}_l) - v > \epsilon$. We then have:

$$\frac{1}{T} \sum_{t=1}^{\top} \boldsymbol{x}_t^T \boldsymbol{A} \boldsymbol{y}_t \geq \frac{v + \left(v + \frac{\epsilon^2}{\beta} \right)}{2} = v + \frac{\epsilon^2}{2\beta}.$$

We also note that, from the definition of the value of the game, we have:

$$\min_i \frac{1}{T} \sum_{t=1}^{\top} \mathbf{e}_i^T \boldsymbol{A} \boldsymbol{y}_t = \min_i \mathbf{e}_i^T \boldsymbol{A} \frac{\sum_{t=1}^{\top} \boldsymbol{y}_t}{T} \leq v.$$

Thus, we have:

$$\lim_{T \to \infty} \min_i \frac{1}{T} \sum_{t=1}^{\top} \mathbf{e}_i^T \boldsymbol{A} \boldsymbol{y}_t - \frac{1}{T} \sum_{t=1}^{\top} \boldsymbol{x}_t^T \boldsymbol{A} \boldsymbol{y}_t \leq v - \left(v + \frac{\epsilon^2}{2\beta} \right) = -\frac{\epsilon^2}{2\beta},$$

contradicting to the definition of a no-regret algorithm:

$$\lim_{T \to \infty} \min_i \frac{1}{T} \sum_{t=1}^{\top} \mathbf{e}_i^T \boldsymbol{A} \boldsymbol{y}_t - \frac{1}{T} \sum_{t=1}^{\top} \boldsymbol{x}_t^T \boldsymbol{A} \boldsymbol{y}_t = 0.$$

□

For no-regret algorithms with optimal regret bound $\mathcal{R}_l = O(\sqrt{l})$, following Theorem 2.5, the players will reach an ϵ-Nash equilibrium in at most $\mathcal{O}(\frac{1}{\epsilon})^4$ rounds. Due to the full information feedback assumption, the column player will know when the row player plays an ϵ-Nash equilibrium strategy. Depending on the number of rounds, the column player can lead the row player to play any ϵ-Nash equilibrium, after that switching from

LRCA to $\boldsymbol{y}^*$ so the row player will remain to play the ϵ-Nash equilibrium (due to the stability property).

With the stability property, we prove that LRCA can drive the row player to play an ϵ-Nash equilibrium, where ϵ can be chosen arbitrarily small. However, in situations where the stability property does not hold (i.e., $\boldsymbol{A}\boldsymbol{y}^* \neq [v, v, \ldots, v]$ or the row player follows μ-regret algorithms with constant step size), we need different analyses for LRCA. In the following sections, we provide refined analyses for the LRCA algorithm with respect to specific algorithms followed by the row player.

2.5.2 Last Round Convergence under MWU/LMWU

One of the most well-studied no-regret algorithms in the game theory literature is the multiplicative weights update (MWU) method, which can be defined as follows:

Definition 2.6 (Freund and Schapire (1999))**.** Let $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ be a sequence of mixed strategies played by the column player. The row player is said to follow the MWU algorithm if strategy $\boldsymbol{x}_{t+1}$ is updated as follows:

$$\boldsymbol{x}_{t+1}(i) = \boldsymbol{x}_t(i)\frac{e^{-\mu_t \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_t}}{Z_t},\ i \in \{1, \ldots n\},$$
$$\text{where } \begin{cases} Z_t = \sum_{i=1}^n \boldsymbol{x}_t(i) e^{-\mu_t \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_t}, \mu_t \in [0, \infty) \text{ is a parameter,} \\ \boldsymbol{e}_i,\ i \in \{1, \ldots, n\}, \text{ is the unit-vector with 1 at the } ith \text{ component.} \end{cases}$$

Bailey and Piliouras (2018) proved that if both players follow MWU then in the case of interior minimax equilibrium, the strategies will move away from the equilibrium and towards the boundary(e.g., the blue trajectory in Figure 2.1).

A variant of MWU is the Linear Multiplicative Weight Update(LMWU), which is also a no-regret algorithm with a suitable step size:

Definition 2.7. The row player is said to play the LMWU if the row player updates the strategy as follows:

$$\boldsymbol{x}_{t+1}(i) = \frac{\boldsymbol{x}_t(i)(1 - \mu_t \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_t)}{\sum_{j=1}^n \boldsymbol{x}_t(j)(1 - \mu_t \boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}_t)}\ \forall i \in \{1, \ldots n\}.$$

In this subsection, we prove that Algorithm 11 (LRCA) played by the column player will lead to last round convergence in the case of MWU/LMWU. The following lemma shows that the relative entropy between the strategy of the row player and the minimax equilibrium is non-increasing.

Lemma 2.8. *Assume that the row player follows the MWU/LMWU algorithm with a non-increasing step size μ_t such that there exists $t' \in \mathbb{N}$ with $\mu_{t'} \leq \frac{1}{3}$. If the column*

player follows LRCA with $\beta \geq 2$ then

$$RE\left(\boldsymbol{x}^* \| \boldsymbol{x}_{2k-1}\right) - RE\left(\boldsymbol{x}^* \| \boldsymbol{x}_{2k+1}\right) \geq \frac{1}{2}\mu_{2k}\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) \;\; \forall k \in \mathbb{N}: \;\; 2k \geq t',$$

where RE denotes the relative entropy, which is defined in Definition 2.2.

Proof. First, we provide the proof in the case of MWU. Following Definition 2 of relative entropy we have:

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&= (RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k})) + (RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1})) \\
&= \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k+1}(i)}\right) - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k}(i)}\right)\right) + \\
&\quad \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k}(i)}\right) - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k-1}(i)}\right)\right) \\
&= \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}_{2k}(i)}{\boldsymbol{x}_{2k+1}(i)}\right)\right) + \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}_{2k-1}(i)}{\boldsymbol{x}_{2k}(i)}\right)\right).
\end{aligned}$$

Following the update rule of the multiplicative weights update algorithm in Definition 3.1 we have:

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&= \left(\mu_{2k}\boldsymbol{x}^{*\top}\boldsymbol{A}\boldsymbol{y}_{2k} + \log(Z_{2k})\right) + \left(\mu_{2k-1}\boldsymbol{x}^{*\top}\boldsymbol{A}\boldsymbol{y}_{2k-1} + \log(Z_{2k-1})\right) \\
&\leq \left(\mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k}(i) e^{-\mu_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k}}\right)\right) + (\mu_{2k-1}v + \log(Z_{2k-1})) \qquad (2.3a) \\
&= \left(\mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k-1}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k-1}} e^{-\mu_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k}}\right) - \log(Z_{2k-1})\right) \\
&+ (\mu_{2k-1}v + \log(Z_{2k-1})),
\end{aligned}$$

where Inequality (2.3a) is due to the fact that $\boldsymbol{x}^{*\top}\boldsymbol{A}\boldsymbol{y} \leq v \; \forall y \in \Delta_m$. Thus,

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&\leq \left(\mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k-1}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}^*} e^{-\mu_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k}}\right)\right) + \mu_{2k-1}v \\
&\leq \left(\mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k-1}v} e^{-\mu_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k}}\right)\right) + \mu_{2k-1}v \qquad (2.4a) \\
&= \mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k}}\right),
\end{aligned}$$

where Inequality (2.4a) is the result of the inequality:

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}^* \geq v \quad \forall \boldsymbol{x} \in \Delta_n.$$

Now, using the update rule of Algorithm 1 (LRCA)

$$y_{2k} = (1-\alpha_{2k})\boldsymbol{y}^* + \alpha_{2k}\boldsymbol{e}_{2k},$$

we then have:

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&\leq \mu_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{2k}}\right) \\
&= \mu_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}((1-\alpha_{2k})\boldsymbol{y}^* + \alpha_{2k}\boldsymbol{e}_{2k})}\right) \\
&\leq \mu_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k}((1-\alpha_{2k})v + \boldsymbol{e}_i^\top \boldsymbol{A}(\alpha_{2k}\boldsymbol{e}_{2k}))}\right).
\end{aligned} \tag{2.5a}$$

The Inequality (2.5a) holds as:

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}^* \geq v \quad \forall \boldsymbol{x} \in \Delta_n.$$

This leads to

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&\leq \mu_{2k}\alpha_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k}\alpha_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{e}_{2k}}\right) \\
&\leq \mu_{2k}\alpha_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i)(1-(1-e^{-\mu_{2k}\alpha_{2k}})\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{e}_{2k}\right) \qquad (2.6a) \\
&= \mu_{2k}\alpha_{2k} v + \log\left(1-(1-e^{-\mu_{2k}\alpha_{2k}})\boldsymbol{x}_{2k-1}{}^\top \boldsymbol{A}\boldsymbol{e}_{2k}\right) \\
&\leq \mu_{2k}\alpha_{2k} v - (1-e^{-\mu_{2k}\alpha_{2k}})\boldsymbol{x}_{2k-1}{}^\top \boldsymbol{A}\boldsymbol{e}_{2k} \qquad (2.6b) \\
&= \mu_{2k}\alpha_{2k} v - (1-e^{-\mu_{2k}\alpha_{2k}}) f(\boldsymbol{x}_{2k-1}),
\end{aligned}$$

where Inequalities (2.6*a*, 2.6*b*) are due to

$$\beta^x \leq 1-(1-\beta)x \quad \forall \beta \geq 0\ x \in [0,1] \text{ and } \log(1-x) \leq -x \ \ \forall x < 1.$$

We can develop Inequality (2.6b) further as

$$
\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \leq \mu_{2k}\alpha_{2k}v - \left(1 - e^{-\mu_{2k}\alpha_{2k}}\right) f(\boldsymbol{x}_{2k-1}) \\
&\leq \mu_{2k}\alpha_{2k}v - \left(1 - \left(1 - \mu_{2k}\alpha_{2k} + \frac{1}{2}(\mu_{2k}\alpha_{2k})^2\right)\right) f(\boldsymbol{x}_{2k-1}) \qquad (2.7a) \\
&= -\mu_{2k}\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) + \frac{1}{2}(\mu_{2k}\alpha_{2k})^2 f(\boldsymbol{x}_{2k-1}) \\
&\leq -\mu_{2k}\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) + \frac{1}{2}\mu_{2k}\alpha_{2k}\mu_{2k}\frac{f(\boldsymbol{x}_{2k-1}) - v}{f(\boldsymbol{x}_{2k-1})} f(\boldsymbol{x}_{2k-1}) \qquad (2.7b) \\
&\leq -\mu_{2k}\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) + \frac{1}{2}\mu_{2k}\alpha_{2k}\ (f(\boldsymbol{x}_{2k-1}) - v) \qquad (2.7c) \\
&= -\frac{1}{2}\mu_{2k}\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) \leq 0.
\end{aligned}
$$

Here, Inequality (2.7a) is due to $e^x \leq 1 + x + \frac{1}{2}x^2 \ \ \forall x \in [-\infty, 0]$, Inequality (2.7b) comes from the definition of α_t:

$$\alpha_t = \frac{f(\boldsymbol{x}_{t-1}) - v}{\beta},\ \beta \geq 2,\ f(\boldsymbol{x}_{2k-1}) \leq 1$$

and finally Inequality (2.7c) comes from the choice of k at the beginning of the proof, i.e., $\mu_{2k} \leq 1$.

We now consider the LMWU case. From the step size assumption of LMWU algorithm, we have:

$$\exists t \in \mathbb{N} \text{ such that } \mu_t \leq \frac{1}{3} \text{ and } \lim_{i=t}^{\infty} \mu_i = \infty.$$

Using the update rule of LMWU in Definition 3.3 we obtain

$$\frac{\boldsymbol{x}_{m+1}(1)}{\boldsymbol{x}_m(1)} : \ldots : \frac{\boldsymbol{x}_{m+1}(n)}{\boldsymbol{x}_m(n)} = (1 - \mu_m {\boldsymbol{e}_1}^\top \boldsymbol{A}\boldsymbol{y}_m) : \ldots : (1 - \mu_m {\boldsymbol{e}_n}^\top \boldsymbol{A}\boldsymbol{y}_m)\ \forall m.$$

Take m equal t and $t-1$ and time the equations side by side we obtain

$$
\begin{aligned}
&\frac{\boldsymbol{x}_{t+1}(1)}{\boldsymbol{x}_{t-1}(1)} : \frac{\boldsymbol{x}_{t+1}(2)}{\boldsymbol{x}_{t-1}(2)} : \ldots : \frac{\boldsymbol{x}_{t+1}(n)}{\boldsymbol{x}_{t-1}(n)} = (1 - \mu_t {\boldsymbol{e}_1}^\top \boldsymbol{A}\boldsymbol{y}_t)(1 - \mu_{t-1} {\boldsymbol{e}_1}^\top \boldsymbol{A}\boldsymbol{y}_{t-1}) : \\
&(1 - \mu_t {\boldsymbol{e}_2}^\top \boldsymbol{A}\boldsymbol{y}_t)(1 - \mu_{t-1} {\boldsymbol{e}_2}^\top \boldsymbol{A}\boldsymbol{y}_{t-1}) : \ldots : (1 - \mu_t {\boldsymbol{e}_n}^\top \boldsymbol{A}\boldsymbol{y}_t)(1 - \mu_{t-1} {\boldsymbol{e}_n}^\top \boldsymbol{A}\boldsymbol{y}_{t-1}) \\
&\implies \boldsymbol{x}_{t+1}(i) = \frac{\boldsymbol{x}_{t-1}(i)(1 - \mu_t {\boldsymbol{e}_i}^\top \boldsymbol{A}\boldsymbol{y}_t)(1 - \mu_{t-1} {\boldsymbol{e}_i}^\top \boldsymbol{A}\boldsymbol{y}_{t-1})}{\sum_{j=1}^n \boldsymbol{x}_{t-1}(j)(1 - \mu_t {\boldsymbol{e}_j}^\top \boldsymbol{A}\boldsymbol{y}_t)(1 - \mu_{t-1} {\boldsymbol{e}_j}^\top \boldsymbol{A}\boldsymbol{y}_{t-1})} \quad \forall i \in 1, 2, \ldots n.
\end{aligned}
$$

Note that for t is event, $\boldsymbol{y}_{t-1} = \boldsymbol{y}^*$ in LRCA-1 algorithm. For any i such that : ${\boldsymbol{e}_i}^\top \boldsymbol{A}\boldsymbol{y}^* = v$ we have:

$$
\begin{aligned}
\frac{\boldsymbol{x}_{t+1}(i)}{\boldsymbol{x}_{t-1}(i)} &= \frac{(1 - \mu_{t-1} {\boldsymbol{e}_i}^\top \boldsymbol{A}\boldsymbol{y}^*)(1 - \mu_t {\boldsymbol{e}_i}^\top \boldsymbol{A}\boldsymbol{y}_t)}{\sum_{j=1}^n \boldsymbol{x}_{t-1}(j)(1 - \mu_{t-1} {\boldsymbol{e}_j}^\top \boldsymbol{A}\boldsymbol{y}^*)(1 - \mu_t {\boldsymbol{e}_j}^\top \boldsymbol{A}\boldsymbol{y}_t)} \\
&= \frac{(1 - \mu_{t-1}v)(1 - \mu_t {\boldsymbol{e}_i}^\top \boldsymbol{A}\boldsymbol{y}_t)}{\sum_{j=1}^n \boldsymbol{x}_{t-1}(j)(1 - \mu_{t-1} {\boldsymbol{e}_j}^\top \boldsymbol{A}\boldsymbol{y}^*)(1 - \mu_t {\boldsymbol{e}_j}^\top \boldsymbol{A}\boldsymbol{y}_t)} \\
&= \frac{(1 - \mu_t {\boldsymbol{e}_i}^\top \boldsymbol{A}\boldsymbol{y}_t)}{\sum_{j=1}^n \boldsymbol{x}_t(j)\frac{1 - \mu_{t-1} {\boldsymbol{e}_j}^\top \boldsymbol{A}\boldsymbol{y}^*}{1 - \mu_{t-1}v}(1 - \mu_t \boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}_t)} \geq \frac{(1 - \mu_t {\boldsymbol{e}_i}^\top \boldsymbol{A}\boldsymbol{y}_t)}{\sum_{j=1}^n \boldsymbol{x}_{t-1}(j)(1 - \mu_t {\boldsymbol{e}_j}^\top \boldsymbol{A}\boldsymbol{y}_t)}.
\end{aligned}
$$

The last inequality is due to $\boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}^* \geq v \ \ \forall j \in \{1, \ldots, n\}$.

We also have for any j such that : ${\boldsymbol{e}_j}^\top \boldsymbol{A}\boldsymbol{y}^* > v$ then $\boldsymbol{x}^*(j) = 0$ for any minimax equilibrium strategy $\boldsymbol{x}^*$. Therefore, we have:

$$RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t-1}) - RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t+1}) = \sum_{i=1}^{n} \boldsymbol{x}^*(i) log\left(\frac{\boldsymbol{x}_{t+1}(i)}{\boldsymbol{x}_{t-1}(i)}\right)$$
$$\geq \sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{(1-\mu_t {\boldsymbol{e}_i}^\top \boldsymbol{A}\boldsymbol{y}_t)}{\sum_{j=1}^{n} \boldsymbol{x}_{t-1}(j)(1-\mu_t {\boldsymbol{e}_j}^\top \boldsymbol{A}\boldsymbol{y}_t)}\right) = \sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{(1-\mu_t {\boldsymbol{e}_i}^\top \boldsymbol{A}\boldsymbol{y}_t)}{1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t}\right).$$

Applying inequality $log(x) \geq (x-1) - (x-1)^2 \ \forall x \geq 0.5$ to the above equation, we obtain

$$RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t-1}) - RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t+1}) \geq \sum_{i=1}^{n} \boldsymbol{x}^*(i) \left(\frac{(1-\mu_t {\boldsymbol{e}_i}^\top \boldsymbol{A}\boldsymbol{y}_t)}{1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t} - 1 - \left(\frac{(1-\mu_t {\boldsymbol{e}_i}^\top \boldsymbol{A}\boldsymbol{y}_t)}{1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t} - 1\right)^2\right)$$
$$= \frac{\mu_t(\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t - {\boldsymbol{x}^*}^\top \boldsymbol{A}\boldsymbol{y}_t)}{1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t} - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \frac{\mu_t^2(\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t - {\boldsymbol{e}_i}^\top \boldsymbol{A}\boldsymbol{y}_t)^2}{(1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t)^2}.$$

Now, following Algorithm 1 (LRCA), we have: $\boldsymbol{y}_t = (1-\alpha_t)\boldsymbol{y}^* + \alpha_t \boldsymbol{e}_t$. For j such that $\boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}^* > v$, we have $\boldsymbol{x}^*(j) = 0$. We can simplify the above equation accordingly and use the Cauchy theorem to obtain

$$\begin{aligned} &RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t-1}) - RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t+1}) \geq \\ &\frac{\mu_t(1-\alpha_t)(\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}^* - v)}{1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t} - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \frac{2\mu_t^2(1-\alpha_t)^2(\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}^* - v)^2}{(1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t)^2} \\ &+ \frac{\mu_t\alpha_t(\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{e}_t - {\boldsymbol{x}^*}^\top \boldsymbol{A}\boldsymbol{e}_t)}{1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t} - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \frac{2\mu_t^2\alpha_t^2(\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{e}_t - \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{e}_t)^2}{(1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t)^2}. \end{aligned} \tag{2.8}$$

For $\mu_t \leq \frac{1}{3}$ we have:

$$\frac{\mu_t(1-\alpha_t)(\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}^* - v)}{1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t} - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \frac{2\mu_t^2(1-\alpha_t)^2(\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}^* - v)^2}{(1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t)^2} \geq 0.$$

We also have:

$$\frac{(\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{e}_t - \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{e}_t)^2}{(1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t)^2} \leq \frac{1}{(1-\mu_t)(1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t)}.$$

Follow the Inequality (2.8), we obtain

$$RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t-1}) - RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t+1}) \geq \frac{\mu_t\alpha_t(\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{e}_t - {\boldsymbol{x}^*}^\top \boldsymbol{A}\boldsymbol{e}_t)}{1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t} - \frac{2\mu_t^2\alpha_t^2}{(1-\mu_t)(1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t)}.$$

By definition of α_t in LRCA-1 algorithm

$$\alpha_t \leq \frac{\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{e}_t - v}{2} \leq \frac{\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{e}_t - \boldsymbol{x}^{*\top} \boldsymbol{A}\boldsymbol{e}_t}{2},$$

along with $\mu_t \leq \frac{1}{3}$ we have:

$$\frac{1}{2}\frac{\mu_t \alpha_t (\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{e}_t - \boldsymbol{x}^{*\top} \boldsymbol{A}\boldsymbol{e}_t)}{1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t} \geq \frac{2\mu_t^2 \alpha_t^2}{(1-\mu_t)(1-\mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t)}.$$

Thus, we have:

$$\begin{aligned} RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t-1}) - RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t+1}) &\geq \frac{1}{2}\frac{\mu_t \alpha_t (\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{e}_t - \boldsymbol{x}^{*\top} \boldsymbol{A}\boldsymbol{e}_t)}{1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t} \\ &\geq \frac{1}{2}\frac{\mu_t \alpha_t (\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{e}_t - v)}{1 - \mu_t \boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t} \geq \frac{\mu_t \alpha_t (\boldsymbol{x}_{t-1}^\top \boldsymbol{A}\boldsymbol{e}_t - v)}{2} \geq 0 \; \forall t = 2k. \end{aligned}$$

□

This lemma can be used to prove the following result:

Theorem 2.9. *Let $\boldsymbol{A}$ be an $n \times m$ non-zero matrix with entries in $[0,1]$. Assume that the row player follows the MWU/LMWU algorithm with a non-increasing step size μ_t such that $\lim_{T\to\infty} \sum_{t=1}^{T} \mu_t = \infty$ and there exists $t' \in \mathbb{N}$ with $\mu_{t'} \leq \frac{1}{3}$. If the column player plays LRCA then there exists a minimax equilibrium $\bar{\boldsymbol{x}}^*$, such that $\lim_{t\to\infty} RE(\bar{\boldsymbol{x}}^* \| \boldsymbol{x}_t) = 0$ and thus $lim_{t\to\infty}\, \boldsymbol{x}_t = \bar{\boldsymbol{x}}^*$ almost everywhere and $lim_{t\to\infty}\, \boldsymbol{y}_t = \boldsymbol{y}^*$.*

Proof. Let $\boldsymbol{x}^*$ be a minimax equilibrium strategy of the row player ($\boldsymbol{x}^*$ may not be unique). Since μ_t is a non-increasing step size, there exists t' such that $\mu_t \leq \frac{1}{3}$ for all $t \geq t'$. Following Lemma 2.8, for all $k \in \mathbb{N}$ such that $2k \geq t'$, we have

$$RE(\boldsymbol{x}^* \| \boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^* \| \boldsymbol{x}_{2k-1}) \leq -\frac{1}{2}\mu_{2k}\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v). \tag{2.9}$$

Thus, the sequence of relative entropy $RE(\boldsymbol{x}^* \| \boldsymbol{x}_{2k-1})$ is non-increasing for all $k \geq \frac{t'}{2}$. As the sequence is bounded below by 0, it has a limit for any minimax equilibrium strategy $\boldsymbol{x}^*$. Since t' is a finite number and $\sum_{t=1}^{\infty} \mu_t = \infty$, we have $\sum_{t=t'}^{\infty} \mu_t = \infty$. Thus,

$$\lim_{T\to\infty} \sum_{k=\left\lceil \frac{t'}{2} \right\rceil}^{T} \mu_{2k} = \infty.$$

We will prove that $\forall \epsilon > 0$, $\exists h \in \mathbb{N}$ such that following LRCA for the column player and MWU/LMWU algorithm for the row player, the row player will play strategy $\boldsymbol{x}_h$ at round h and $f(\boldsymbol{x}_h) - v \leq \epsilon$. In particular, we prove this by contradiction. That is,

suppose that $\exists \epsilon > 0$ such that $\forall h \in \mathbb{N},\ f(\boldsymbol{x}_h) - v > \epsilon$. Then $\forall k \in \mathbb{N}$,

$$\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) = \frac{(f(\boldsymbol{x}_{2k-1}) - v)^2}{\beta} > \frac{e^2}{\beta}.$$

Let k vary from $\left\lceil \frac{t'}{2} \right\rceil$ to T in equation (2.9). By summing over k, we obtain:

$$\begin{aligned} RE(\boldsymbol{x}^* \| \boldsymbol{x}_{2T+1}) &\leq RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t'}) - \frac{1}{2} \sum_{k=\left\lceil \frac{t'}{2} \right\rceil}^{T} \mu_{2k} \alpha_{2k} (f(\boldsymbol{x}_{2k-1}) - v) \\ &\leq RE(\boldsymbol{x}^* \| \boldsymbol{x}_{t'}) - \frac{1}{2} \frac{e^2}{\beta} \sum_{k=\left\lceil \frac{t'}{2} \right\rceil}^{T} \mu_{2k}. \end{aligned}$$

Since $\lim_{T\to\infty} \sum_{k=\left\lceil \frac{t'}{2} \right\rceil}^{T} \mu_{2k} = \infty$ and $RE(\boldsymbol{x}^* \| \boldsymbol{x}_{T+1}) \geq 0$, which contradicts our assumption about $\forall h \in \mathbb{N},\ f(\boldsymbol{x}_h) - v > \epsilon$.

Now, we take a sequence of $\epsilon_k > 0$ such that $\lim_{k\to\infty} \epsilon_k = 0$. Then for each k, there exists $\boldsymbol{x}_{t_k} \in \Delta_n$ such that $v \leq f(\boldsymbol{x}_{t_k}) \leq v + \epsilon_k$. As Δ_n is a compact set and $\boldsymbol{x}_{t_k}$ is bounded then following the Bolzano-Weierstrass theorem, there is a convergence subsequence $\boldsymbol{x}_{\bar{t}_k}$. The limit of that sequence, $\bar{\boldsymbol{x}}^*$, is a minimax equilibrium strategy of the row player (since $f(\bar{\boldsymbol{x}}^*) = f(\lim_{k\to\infty} \boldsymbol{x}_{\bar{t}_k}) = \lim_{k\to\infty} f(\boldsymbol{x}_{\bar{t}_k}) = v$). Combining with the fact that $RE(\bar{\boldsymbol{x}}^* \| \boldsymbol{x}_{2k-1})$ is non-increasing for $k \geq \left\lceil \frac{t'}{2} \right\rceil$ and $RE(\bar{\boldsymbol{x}}^* \| \bar{\boldsymbol{x}}^*) = 0$, we have $\lim_{k\to\infty} RE(\bar{\boldsymbol{x}}^* \| \boldsymbol{x}_{2k-1}) = 0$. We also note that

$$\begin{aligned} RE(\bar{\boldsymbol{x}}^* \| \boldsymbol{x}_{2k}) - RE(\bar{\boldsymbol{x}}^* \| \boldsymbol{x}_{2k-1}) &= \mu_{2k-1} \bar{\boldsymbol{x}}^{*\top} \boldsymbol{A} \boldsymbol{y}_{2k-1} + \log \left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k-1} \boldsymbol{e}_i^\top \boldsymbol{A} \boldsymbol{y}^*} \right) \\ &\leq \mu_{2k-1} v + \log \left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu_{2k-1} v} \right) = 0, \end{aligned}$$

following the fact that $\boldsymbol{x}^{*\top} \boldsymbol{A} \boldsymbol{y} \leq v$ for all $\boldsymbol{y} \in \Delta_m$ and $\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y}^* \geq v$ for all $\boldsymbol{x} \in \Delta_n$. Thus, we have $\lim_{k\to\infty} RE(\bar{\boldsymbol{x}}^* \| \boldsymbol{x}_{2k}) = 0$ as well. Subsequently, $\lim_{t\to\infty} RE(\bar{\boldsymbol{x}}^* \| \boldsymbol{x}_t) = 0$, which concludes the proof. □

The optimal step size α_t in the case of MWU is $\alpha_t = \frac{f(\boldsymbol{x}_{t-1}) - v}{\mu_t f(\boldsymbol{x}_{t-1})}$. However, in order to make LRCA robust against other algorithms of the row player, we choose the step size as shown in the algorithm. In LRCA Algorithm 11, if we have $f(\boldsymbol{x}_t) - v \leq \epsilon$, then

$$min_{\boldsymbol{x} \in \Delta_n} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y}_t \geq (1 - \alpha_t) v \geq (1 - \epsilon) v \implies v - min_{\boldsymbol{x} \in \Delta_n} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y}_t \leq \epsilon.$$

It is easy to show that these inequalities imply $(\boldsymbol{x}_t, \boldsymbol{y}_t)$ is 2ϵ-nash equilibrium. Follow Lemma 2.8 in the case of constant learning rate $\mu_t = \mu$, we have the complexity of the algorithm in order to achieve ϵ-nash equilibrium is $\mathcal{O}(\frac{\log(n)/\mu}{\epsilon^2})$.

2.5.3 Last Round Convergence under FTRL

We now consider a more general form of no-regret algorithms for the strategic adversary, namely Follow the Regularized Leader (e.g., see Abernethy et al. (2008)).

Definition 2.10. The row player is said to play the FTRL with σ-strongly convex regularizer: $F(\boldsymbol{x})$ if the row player updates the strategy as follows:

$$\boldsymbol{x}_t = \operatorname*{argmin}_{\boldsymbol{x}\in\Delta_n} G_t(\boldsymbol{x}) = \boldsymbol{x}^\top \left(\sum_{i=1}^{t-1} \boldsymbol{A}\boldsymbol{y}_i\right) + \frac{1}{\mu}F(\boldsymbol{x}).$$

FTRL covers a large set of well-known no-regret algorithms. For instance, if the negative entropy function is used as the regularizer, then FTRL results in a fixed step-size Multiplicative Weight Update. In the case of Euclidean regularizer, the FTRL becomes the famous Online Mirror Descent with lazy projection(e.g. see Shalev-Shwartz (2012)). We now have an analysis of last round convergence when play against the general FTRL:

Theorem 2.11. *Assume that the row player follows the FTRL with σ-strongly convex regularizer: $F(\boldsymbol{x})$ with fixed step such that $\mu \leq 1$ and $\sigma \geq 1$. Then if the column player follows the Algorithm 11 (LRCA) with $\beta \geq n^2$, there will be last round convergence to the minimax equilibrium.*

Proof. Let $\boldsymbol{x}^*$ be a minimax equilibrium of the row player. Denote $H_t(\boldsymbol{x}^*) = G_t(\boldsymbol{x}^*) - G_t(\boldsymbol{x}_t)$, following properties of strongly convex function we have:

$$H_t \geq \frac{\sigma}{2\mu}\|\boldsymbol{x}^* - \boldsymbol{x}_t\|^2.$$

Thus, if $H_t(\boldsymbol{x}^*)$ converges to 0 then we have $\boldsymbol{x}_t$ converges to $\boldsymbol{x}^*$. We will prove that

$$H_{t-1}(\boldsymbol{x}^*) - H_{t+1}(\boldsymbol{x}^*) \geq \frac{(f(\boldsymbol{x}_{t-1}) - v)^2}{2n^2} \quad \forall t = 2k.$$

From the definition of H_t we have:

$$\begin{aligned}
&H_{t-1}(\boldsymbol{x}^*) - H_{t+1}(\boldsymbol{x}^*) = (G_{t+1}(\boldsymbol{x}_{t+1}) - G_{t-1}(\boldsymbol{x}_{t-1})) - (G_{t+1}(\boldsymbol{x}^*) - G_{t-1}(\boldsymbol{x}^*)) \\
&= (G_{t-1}(\boldsymbol{x}_{t+1}) - G_{t-1}(\boldsymbol{x}_{t-1}) + {\boldsymbol{x}_{t+1}}^\top \boldsymbol{A}(\boldsymbol{y}_{t-1} + \boldsymbol{y}_t)) - {\boldsymbol{x}^*}^\top \boldsymbol{A}(\boldsymbol{y}_{t-1} + \boldsymbol{y}_t) \\
&\geq \frac{\sigma}{2\mu}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|^2 + {\boldsymbol{x}_{t+1}}^\top \boldsymbol{A}(\boldsymbol{y}_{t-1} + \boldsymbol{y}_t) - {\boldsymbol{x}^*}^\top \boldsymbol{A}(\boldsymbol{y}_{t-1} + \boldsymbol{y}_t) \qquad (2.10a)
\end{aligned}$$

where the last inequality derives from the strongly convex property of G_{t-1}. We note that as $\boldsymbol{x}^* = \operatorname{argmin}_{\boldsymbol{x}\in\Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}^*$, the following inequality holds

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}^* \geq {\boldsymbol{x}^*}^\top \boldsymbol{A}\boldsymbol{y}^* = v \;\forall \boldsymbol{x} \in \Delta_n.$$

Plug it in the inequality (2.10a) and note that $\boldsymbol{y}_{t-1} = \boldsymbol{y}^*$ for an even t, then we have:

$$\begin{aligned}
&H_{t-1}(\boldsymbol{x}^*) - H_{t+1}(\boldsymbol{x}^*) \geq \frac{\sigma}{2\mu}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|^2 + (\boldsymbol{x}_{t+1} - \boldsymbol{x}^*)^\top \boldsymbol{A}\boldsymbol{y}_t \\
&= \frac{\sigma}{2\mu}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|^2 + (\boldsymbol{x}_{t+1} - \boldsymbol{x}^*)^\top((1-\alpha_t)\boldsymbol{y}^* + \alpha_t \boldsymbol{e}_t) && \text{(2.11a)} \\
&\geq \frac{\sigma}{2\mu}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|^2 + \alpha_t(\boldsymbol{x}_{t+1} - \boldsymbol{x}^*)^\top \boldsymbol{A}\boldsymbol{e}_t && \text{(2.11b)} \\
&= \frac{\sigma}{2\mu}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|^2 + \alpha_t(\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1})^\top \boldsymbol{A}\boldsymbol{e}_t + \alpha_t(\boldsymbol{x}_{t-1} - \boldsymbol{x}^*)^\top \boldsymbol{A}\boldsymbol{e}_t \\
&\geq \frac{\sigma}{2\mu}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|^2 - \alpha_t\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|\ \|\boldsymbol{A}\boldsymbol{e}_t\|_* + \alpha_t(f(\boldsymbol{x}_{t-1}) - v). && \text{(2.11c)}
\end{aligned}$$

Equalities (2.11a,2.11b) come from the definition of $\boldsymbol{y}_t$. We have the inequalities (2.11c) as the result of,

$$\boldsymbol{a}^\top \boldsymbol{b} \leq \|\boldsymbol{a}\|\|\boldsymbol{b}\|_*,$$

where $\|\ \|_*$ denotes the dual norm. For vector $\boldsymbol{a}$ such that $\{\boldsymbol{a}|\ 0 < \boldsymbol{a}(i) \leq 1\ \forall i \in [n]\}$ we have:

$$\|\boldsymbol{a}\|_* \leq n.$$

Substitute this into the inequalities (2.11c) we have:

$$\begin{aligned}
&H_{t-1}(\boldsymbol{x}^*) - H_{t+1}(\boldsymbol{x}^*) \geq \frac{\sigma}{2\mu}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\|^2 - n\alpha_t\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\| + \alpha_t(f(\boldsymbol{x}_{t-1}) - v) \\
&= \left(\sqrt{\frac{\sigma}{2\mu}}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_{t-1}\| - \frac{n\alpha_t}{2\sqrt{\frac{\sigma}{2\mu}}}\right)^2 + \alpha_t(f(\boldsymbol{x}_{t-1}) - v) - \frac{n^2\alpha_t^2\mu}{2\sigma} \\
&\geq \alpha_t(f(\boldsymbol{x}_{t-1}) - v) - \frac{n^2\alpha_t^2\mu}{2\sigma} \geq \alpha_t(f(\boldsymbol{x}_{t-1}) - v) - \frac{n^2\alpha_t^2}{2}. && \text{(2.12a)}
\end{aligned}$$

Now, from LRCA Algorithm 11 we have

$$\alpha_t = \frac{f(\boldsymbol{x}_{t-1}) - v}{n^2},$$

then inequality (2.12a) implies:

$$H_{t-1}(\boldsymbol{x}^*) - H_{t+1}(\boldsymbol{x}^*) \geq \frac{\alpha_t}{2}(f(\boldsymbol{x}_{t-1}) - v) = \frac{(f(\boldsymbol{x}_{t-1}) - v)^2}{2n^2} \geq 0\ \forall t \text{ even}.$$

Following the same argument in the proof of Theorem 2.9, we have the last round convergence result. □

Note that we only need an upper bound for μ and a lower bound for σ in order to prove the Theorem 2.11. The FTRL with negative entropy regularizer becomes the MWU with constant step size μ. However, when μ varies in each update, then the two algorithms can be significantly different and thus the analysis in Theorem 2.9 is necessary. From

the analysis of Theorem 2.11, the complexity of the algorithm in order to achieve ϵ-Nash equilibrium is $\mathcal{O}(\frac{n^2}{\epsilon^2})$.

2.5.4 Last Round Convergence under Optimistic MWU

When following the Optimistic Multiplicative Weight Update algorithm Daskalakis and Panageas (2019), the row player will have the following strategy update:

$$\boldsymbol{x}_{t+1}(i) = \boldsymbol{x}_t(i)\frac{e^{-2\mu \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_t + \mu \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{t-1}}}{\sum_{j=1}^n \boldsymbol{x}_t j e^{-2\mu \boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}_t + \mu \boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}_{t-1}}} \quad \forall i \in \{1 \dots n\}.$$

We note that in Algorithm 11, we just use one "stabilizing" strategy $\boldsymbol{y}_{t-1} = \boldsymbol{y}^*$ before we exploit the strategy of the row player. However, in the case of Optimistic Multiplicative Weight Update algorithm, we need two "stabilizing" strategies. It will not change the result of LRCA in other cases, but it will make it slower to converge to the minimax equilibrium of the row player. In this case, let $\boldsymbol{y}_{3k} = \boldsymbol{y}_{3k-1} = \boldsymbol{y}^*$ and $\boldsymbol{y}_{3k+1} = (1-\alpha)\boldsymbol{y}^* + \alpha \boldsymbol{e}_{3k+1}$ where $\boldsymbol{e}_{3k+1} = \text{argmax}_{\boldsymbol{y}\in\Delta_m} \boldsymbol{x}_{3k}^\top \boldsymbol{A}\boldsymbol{y}, \quad \forall k \in \mathbb{N}$. Then we have:

$$\boldsymbol{x}_{3k+3}(i) = \boldsymbol{x}_{3k}(i)\frac{e^{-2\mu \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{3k+2} - \mu \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{3k+1} - \mu \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{3k} + \mu \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{3k-1}}}{\sum_{j=1}^n \boldsymbol{x}_{3k}(j) e^{-2\mu \boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}_{3k+2} - \mu \boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}_{3k+1} - \mu \boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}_{3k} + \mu \boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}_{3k-1}}}.$$

Following that we then have:

$$\begin{aligned}
&RE(\boldsymbol{x}^* \| \boldsymbol{x}_{3k}) - RE(\boldsymbol{x}^* \| \boldsymbol{x}_{3k+3}) = \sum_{i=1}^n \boldsymbol{x}^*(i) \log(\frac{\boldsymbol{x}_{3k+3}(i)}{\boldsymbol{x}_{3k}(i)}) \\
&= \sum_{i=1}^n \boldsymbol{x}^*(i) \log(\frac{e^{-2\mu \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}^* - \mu \boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{y}_{3k+1}}}{\sum_{j=1}^n \boldsymbol{x}_{3k}(j) e^{-2\mu \boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}^* - \mu \boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}_{3k+1}}}) \\
&= -2\mu v - \mu \boldsymbol{x}^{*\top} \boldsymbol{A}\boldsymbol{y}_{3k+1} - \log(\sum_{j=1}^n \boldsymbol{x}_{3k}(j) e^{-2\mu \boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}^* - \mu \boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}_{3k+1}}) \\
&\geq -2\mu v - \mu v - (-2\mu v) - \log(\sum_{j=1}^n \boldsymbol{x}_{3k}(j) e^{-\mu \boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}_{3k+1}}) \\
&= -\mu v - \log(\sum_{j=1}^n \boldsymbol{x}_{3k}(j) e^{-\mu \boldsymbol{e}_j^\top \boldsymbol{A}\boldsymbol{y}_{3k+1}}).
\end{aligned}$$

Now it comes back the exact step in the proof of Theorem 2.9 and with the same chosen step size we will have the last round convergence.

2.5.5 Convergence with Minimax Equilibrium Estimation

It is well-known that if both players follow a no-regret algorithm, then the average strategy will converge to a minimax equilibrium Cesa-Bianchi and Lugosi (2006). Bailey and Piliouras (2019) considered a more interesting setting where both players use a constant step size gradient algorithm(i.e., algorithms with a constant regret). They proved that in the case of 2×2 matrix $\boldsymbol{A}$, there will be average convergence to minimax equilibrium. Further, their experimental results suggest that the result holds true for every size of matrix $\boldsymbol{A}$. In this section, we consider a symmetric game in which the row player follows the Multiplicative Weight Update Algorithm. Without the knowledge of the matrix $\boldsymbol{A}$, the column player first plays a no-regret algorithm and collects the historical average strategy: an estimation of $\boldsymbol{y}^*$. After having the estimation, the column player then follows the Algorithm 11. We prove that the strategies of the row and column player will converge to an arbitrarily small ball containing the minimax equilibrium.

Theorem 2.12. *Assume that the row player follows the MWU algorithm with a fixed step size $\mu > 0$. For any $\lambda > 0$, there exists $\epsilon > 0$ such that if the column player follows LRCA with the approximations of $\boldsymbol{y}^*$, v as $\bar{\boldsymbol{y}}$, $\bar{v}$ and $\min_{\boldsymbol{x} \in \Delta_n} \boldsymbol{x}^\top \boldsymbol{A} \bar{\boldsymbol{y}} > v - \epsilon$ with $v + \epsilon > \bar{v} > v - \epsilon$, then there exist T and such that for every $t > T$, there is a minimax equilibrium $\boldsymbol{x}^*$ such that $RE(\boldsymbol{x}^* \| \boldsymbol{x}_t) < \lambda$.*

Proof of Theorem 2.12. We first provide some lemmas before proving the theorem.

Lemma 2.13. *Let $\boldsymbol{A}$ be a matrix of a two-players zero-sum game with entries in [0,1]. For all $\lambda > 0$, there exists $\epsilon > 0$ such that:*

$$if \max_{\boldsymbol{y} \in \Delta_m} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y} \leq v + \epsilon \implies \text{There exists a minimax equilibrium } \boldsymbol{x}^* \text{ such that } ||\boldsymbol{x} - \boldsymbol{x}^*|| < \lambda.$$

Proof. Let denote $f(\boldsymbol{x}) = \max_{\boldsymbol{y} \in \Delta_m} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y}$. Consider a closed and bounded set:

$$S = \{\boldsymbol{x} \in \Delta_n |\ ||\boldsymbol{x} - \boldsymbol{x}^*|| \geq \lambda \ \forall \text{ equilibria points } \boldsymbol{x}^*\}.$$

$f(\boldsymbol{x})$ is a continuous function on the closed and bounded set S, therefore it achieves a minimum v' in S. Since the construction of S, we have $v' - v > 0$. Pick $0 < \epsilon < v' - v$, then

$$\forall \boldsymbol{x} \in \Delta_n,\ max_{\boldsymbol{y} \in \Delta_m} \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y} \leq v + \epsilon \leq v' \implies \boldsymbol{x} \notin S \implies \exists \boldsymbol{x}^* \text{such that} ||\boldsymbol{x} - \boldsymbol{x}^*|| < \lambda.$$

□

Lemma 2.14. *Let $\boldsymbol{x}$ be a point in the Δ_n. Then for every $\lambda > 0$, there exists $\epsilon > 0$ such that $\forall \boldsymbol{y} \in \Delta_n$*

$$||\boldsymbol{x} - \boldsymbol{y}|| < \epsilon \implies Re(\boldsymbol{x} || \boldsymbol{y}) \leq \lambda.$$

Proof. For $x_i = 0$, we have $x_i \log(\frac{x_i}{y_i}) = 0$ so w.l.o.g, we assume $x_i > 0 \ \forall i \in [n]$. Let $x_k = \min_{j\in[n]} x_j$. Pick $0 < \epsilon < x_k$ such that

$$\log(\frac{x_k}{x_k - \epsilon}) \leq \lambda.$$

With the assumption that $||\boldsymbol{x} - \boldsymbol{y}|| < \epsilon$, we have $y_i \geq x_i - \epsilon$. Then,

$$RE(\boldsymbol{x}||\boldsymbol{y}) = \sum_{i=1}^{n} x_i \log(\frac{x_i}{y_i}) \leq \sum_{i=1}^{n} x_i \log(\frac{x_i}{x_i - \epsilon}) \leq \sum_{i=1}^{n} x_i \log(\frac{x_k}{x_k - \epsilon}) = \log(\frac{x_k}{x_k - \epsilon}) \leq \lambda.$$

□

Now, we can prove the above theorem. Following the Definition of relative entropy we have:

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&= (RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k})) + (RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1})) \\
&= \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k+1}(i)}\right) - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k}(i)}\right)\right) + \\
&\quad \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k}(i)}\right) - \sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}^*(i)}{\boldsymbol{x}_{2k-1}(i)}\right)\right) \\
&= \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}_{2k}(i)}{\boldsymbol{x}_{2k+1}(i)}\right)\right) + \left(\sum_{i=1}^{n} \boldsymbol{x}^*(i) \log\left(\frac{\boldsymbol{x}_{2k-1}(i)}{\boldsymbol{x}_{2k}(i)}\right)\right).
\end{aligned}$$

Following the update rule of the multiplicative weights update algorithm we have:

$$\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&= \left(\mu \boldsymbol{x}^{*\top} \boldsymbol{A} \boldsymbol{y}_{2k} + \log(Z_{2k})\right) + \left(\mu \boldsymbol{x}^{*\top} \boldsymbol{A} \boldsymbol{y}_{2k-1} + \log(Z_{2k-1})\right) \\
&\leq \left(\mu v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k}(i) e^{-\mu \boldsymbol{e}_i^\top \boldsymbol{A} \boldsymbol{y}_{2k}}\right)\right) + (\mu v + \log(Z_{2k-1})) \qquad (2.13a) \\
&= \left(\mu v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu \boldsymbol{e}_i^\top \boldsymbol{A} \boldsymbol{y}_{2k-1}} e^{-\mu \boldsymbol{e}_i^\top \boldsymbol{A} \boldsymbol{y}_{2k}}\right) - \log(Z_{2k-1})\right) \\
&+ (\mu v + \log(Z_{2k-1})),
\end{aligned}$$

where Inequality (2.13a) is due to the fact that $\boldsymbol{x}^{*\top} \boldsymbol{A} \boldsymbol{y} \leq v \ \forall \boldsymbol{y} \in \Delta_m$. Now, using the update rule of Algorithm (LRCA)

$$y_{2k} = (1 - \alpha_{2k})\bar{\boldsymbol{y}} + \alpha_{2k} \boldsymbol{e}_{2k},$$

we then have:

$$
\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&\leq \left(\mu v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu \boldsymbol{e}_i^\top \boldsymbol{A}\bar{\boldsymbol{y}}} e^{-\mu \boldsymbol{e}_i^\top \boldsymbol{A}((1-\alpha_{2k})\bar{\boldsymbol{y}} + \alpha_{2k}\boldsymbol{e}_{2k})}\right)\right) + \mu v \\
&\leq \left(\mu v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu(v-\epsilon)} e^{-\mu(1-\alpha_{2k})(v-\epsilon) - \mu\alpha_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{e}_{2k}}\right)\right) + \mu v \qquad (2.14a) \\
&= \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu\alpha_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{e}_{2k}}\right),
\end{aligned}
$$

where Inequality (2.14a) is the result of the inequality:

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y}^* \geq v \quad \forall \boldsymbol{x} \in \Delta_n.$$

This leads to

$$
\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\
&\leq \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i) e^{-\mu\alpha_{2k}\boldsymbol{e}_i^\top \boldsymbol{A}\boldsymbol{e}_{2k}}\right) \\
&\leq \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2k-1}(i)(1 - (1 - e^{-\mu\alpha_{2k}})\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{e}_{2k}\right) \qquad (2.15a) \\
&= \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k} v + \log\left(1 - (1 - e^{-\mu\alpha_{2k}})\boldsymbol{x}_{2k-1}{}^\top \boldsymbol{A}\boldsymbol{e}_{2k}\right) \\
&\leq \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k} v - (1 - e^{-\mu\alpha_{2k}})\boldsymbol{x}_{2k-1}{}^\top \boldsymbol{A}\boldsymbol{e}_{2k} \qquad (2.15b) \\
&= \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k} v - (1 - e^{-\mu\alpha_{2k}}) f(\boldsymbol{x}_{2k-1}),
\end{aligned}
$$

where Inequalities (2.15*a*, 2.15*b*) are due to

$$\beta^x \leq 1 - (1-\beta)x \quad \forall \beta \geq 0 \; x \in [0,1] \text{ and } \log(1-x) \leq -x \;\; \forall x < 1.$$

We can develop Inequality (2.15b) further as

$$
\begin{aligned}
&RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \leq \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k} v - \left(1 - e^{-\mu\alpha_{2k}}\right) f(\boldsymbol{x}_{2k-1}) \\
&\leq \epsilon(2\mu - \mu\alpha_{2k}) + \mu\alpha_{2k} v - \left(1 - \left(1 - \mu\alpha_{2k} + \frac{1}{2}(\mu\alpha_{2k})^2\right)\right) f(\boldsymbol{x}_{2k-1}) \qquad (2.16a) \\
&= \epsilon(2\mu - \mu\alpha_{2k}) - \mu\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) + \frac{1}{2}(\mu\alpha_{2k})^2 f(\boldsymbol{x}_{2k-1}) \\
&\leq \epsilon(2\mu - \mu\alpha_{2k}) - \mu\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v) + \frac{1}{2}\mu\alpha_{2k}\mu\frac{f(\boldsymbol{x}_{2k-1}) - \bar{v} + \epsilon}{f(\boldsymbol{x}_{2k-1})} f(\boldsymbol{x}_{2k-1}) \qquad (2.16b) \\
&\leq \epsilon(2\mu - \mu\alpha_{2k}) - \frac{1}{2}\mu\alpha_{2k}(f(\boldsymbol{x}_{2k-1}) - v - 2\epsilon) \qquad (2.16c) \\
&\leq \epsilon(2\mu - \mu\alpha_{2k}) - \frac{1}{2}\mu(f(\boldsymbol{x}_{2k-1}) - v)(f(\boldsymbol{x}_{2k-1}) - v - 2\epsilon).
\end{aligned}
$$

Here, Inequality (2.16a) is due to $e^x \leq 1 + x + \frac{1}{2}x^2 \quad \forall x \in [-\infty, 0]$, Inequality (2.16b) comes from the definition of α_t:

$$\alpha_t = \frac{f(\boldsymbol{x}_{t-1}) - \bar{v} + \epsilon}{f(\boldsymbol{x}_{t-1})},$$

and finally, Inequality (2.16c) comes from the choice of k at the beginning of the proof, i.e., $\mu_{2k} \leq 1$. If

$$f(\boldsymbol{x}_{2k-1}) - v \geq 3\sqrt{\epsilon} \text{ and } \epsilon \leq \frac{1}{4},$$

then we have

$$\begin{aligned} &RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) \\ &\leq \epsilon(2\mu) - \frac{1}{2}3\sqrt{\epsilon}(2\sqrt{\epsilon}) \\ &\leq -\epsilon\mu < 0. \end{aligned}$$

Using lemma 2.13 and lemma 2.14, pick ϵ such that if $f(\boldsymbol{x}_{2k-1}) - v < 3\sqrt{\epsilon}$, then

$$RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) < \lambda_1 < \lambda - 3\epsilon\mu.$$

Since $RE()$ is non-negative, there exists k such that $f(\boldsymbol{x}_{2k-1}) - v < 3\sqrt{\epsilon}$. It leads to $RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) < \lambda - \epsilon\mu$. If $f(\boldsymbol{x}_{2k+1}) - v < 3\sqrt{\epsilon}$, then

$$RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) < \lambda_1 < \lambda - \epsilon\mu.$$

If $f(\boldsymbol{x}_{2k-1}) - v > 3\sqrt{\epsilon}$, then

$$RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) < RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) + 2\epsilon\mu < \lambda_1 + 2\epsilon\mu < \lambda - \epsilon\mu$$

$$RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+3}) < RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) - \epsilon\mu < \lambda_1 + \epsilon\mu < \lambda - \epsilon\mu.$$

Following this process, we then have the K-L distance $RE(\boldsymbol{x}^*||\boldsymbol{x}_{2l-1}) < \lambda - \epsilon\mu < \lambda$ for all $l \geq k$.

For the even round, for all $l \geq k$ we have:

$$\begin{aligned} RE(\boldsymbol{x}^*||\boldsymbol{x}_{2l}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2l-1}) &= \mu \boldsymbol{x}^{*\top} \boldsymbol{A} \boldsymbol{y}_{2l-1} + \log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2l-1}(i) e^{\mu \boldsymbol{e}_i{}^\top \boldsymbol{A}\bar{\boldsymbol{y}}}\right) \\ &\leq \mu v + log\left(\sum_{i=1}^{n} \boldsymbol{x}_{2l-1}(i) e^{-\mu(v-\epsilon)}\right) \\ &= \mu\epsilon. \end{aligned}$$

This implies that

$$RE(\boldsymbol{x}^*||\boldsymbol{x}_{2l}) < RE(\boldsymbol{x}^*||\boldsymbol{x}_{2l-1}) + \mu\epsilon < (\lambda - \mu\epsilon) + \mu\epsilon = \lambda \; \forall \; l \geq k.$$

□

Next, we prove that the LRCA algorithm is a no-dynamic regret algorithm under mild conditions.

2.6 No-dynamic Regret Algorithm

In this section, we first show that if the column player wants to achieve both the no-regret and stability properties, then the row player's strategy needs to converge to the minimax equilibrium. We then show that LRCA is a no-dynamic regret algorithm for the column player when the row player follows the aforementioned no-regret algorithms. In a general case, we suggest a method to combine our LRCA algorithm with another no-regret algorithm (such that Adahedge De Rooij et al. (2014)) so that the new algorithm will still have no-regret property against random sequences of the row player while maintaining no-dynamic regret in the specific situation.

Lemma 2.15. *Suppose that the row player is a strategic adversary who follows a common no-regret algorithm such as MWU, OMD, FTRL, LMWU or OMWU. Then, the column player cannot achieve last round convergence and the no-regret property if the row player's strategy does not converge to a minimax equilibrium of the game.*

Proof. Suppose that the column player achieves both stability and no-regret property. The strategy of the column player will then converge, say to $\hat{y}$. Following the property of common no-regret algorithms, the strategy of the row player will also converge to a single best response $\hat{\boldsymbol{x}}$ to $\hat{\boldsymbol{y}}$:

$$\hat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}\in\Delta_n} \boldsymbol{x}^\top \boldsymbol{A}\hat{\boldsymbol{y}}.$$

Furthermore, since the strategy of the column player is no-regret, we must also have

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y}\in\Delta_m} \hat{\boldsymbol{x}}^\top \boldsymbol{A}\boldsymbol{y}.$$

Therefore, by definition, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ is a minimax equilibrium of the game. □

Our algorithm LRCA satisfies the sufficient condition in Lemma 2.15. Next, we prove the no-dynamic regret property of the algorithm, clarifying the design of LRCA.

Theorem 2.16. *Assume that the row player is a strategic adversary who follows the above-mentioned no-regret type algorithms: MWU/LMWU, FTRL. If there exists a fully mixed minimax strategy for the row player, then by following LRCA, the column player will achieve the no-dynamic regret property with the dynamic regret satisfying* $R_T \leq DR_T = \mathcal{O}\left(\sqrt{\log(n)}T^{3/4}\right)$. *Furthermore, in the case the row player uses a constant learning rate* μ, *we have* $DR_T = \mathcal{O}(\frac{n}{\sqrt{\mu}}T^{1/2})$.

Proof. We first prove the theorem in the case the row player follows the MWU/LMWU algorithm.

For the odd round $2k-1$, the dynamic regret of the column player at round $2k-1$ will satisfy

$$DR^{2k-1} = \max_{i\in 1,..m} \boldsymbol{x}_{2k-1}^\top \boldsymbol{A}\boldsymbol{e}_i - \boldsymbol{x}_{2k-1}^\top \boldsymbol{A}\boldsymbol{y}^* \leq f(\boldsymbol{x}_{2k-1}) - v.$$

For the even round $2k$, considering the existence of the fully mixed minimax equilibrium of the row player, we then have $\boldsymbol{A}\boldsymbol{y}^* = vI_1$ (I_1 is a vector of all 1 element) and thus $\boldsymbol{x}_{2k} = \boldsymbol{x}_{2k-1}$. Therefore $DR^{2k} \leq f(\boldsymbol{x}_{2k-1}) - v$.

Combining the case of odd and even round, we derive

$$DR_T \leq 2\sum_{k=1}^{T/2}(f(\boldsymbol{x}_{2k-1}) - v).$$

Now, following Lemma 2.8 in the case $n \geq 8$, we have

$$\begin{aligned}
&\frac{1}{2}\mu_{2k}\frac{(f(\boldsymbol{x}_{2k-1}) - v)^2}{2} \leq RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k-1}) - RE(\boldsymbol{x}^*||\boldsymbol{x}_{2k+1}) \\
\implies &\sum_{k=1}^{T/2}\mu_{2k}(f(\boldsymbol{x}_{2k-1}) - v)^2 \leq 4RE(\boldsymbol{x}^*||\boldsymbol{x}_1) \leq 4\log(n).
\end{aligned}$$

Using the Cauchy–Schwarz inequality, we can then derive that

$$\sum_{k=1}^{T/2}(f(\boldsymbol{x}_{2k-1}) - v) \leq 2\sqrt{\log(n)}\sqrt{\sum_{k=1}^{T/2}\frac{1}{\mu_{2k}}} \implies DR_T \leq 4\sqrt{\log(n)}\sqrt{\sum_{k=1}^{T/2}\frac{1}{\mu_{2k}}}.$$

If the row player follows the constant step size μ, then we have

$$DR_T \leq \frac{2\sqrt{2}\sqrt{\log(n)}}{\sqrt{\mu}}T^{1/2} = \mathcal{O}(\frac{n}{\sqrt{\mu}}T^{1/2}).$$

If the row player follows a decreasing step size $\mu_k = \sqrt{8\log(n)/k}$ (Cesa-Bianchi and Lugosi (2006)) to make the algorithm no-regret, then we have

$$DR_T \leq \log(n)^{1/4}T^{3/4} = \mathcal{O}(\sqrt{\log(n)}T^{3/4}).$$

Indeed, for any sequence of step size μ_k such that $\sum_{k=1}^{T/2}\frac{1}{\mu_{2k}} \leq \mathcal{O}(T^{3/2})$, the theorem holds.

We continue the proof in the case of FTRL. W.l.o.g, assume that $\max_{\boldsymbol{x}\in\Delta_n} F(\boldsymbol{x}) = 1$. Following the proof of Theorem 2.11 we have

$$\sum_{k=1}^{T/2}(f(\boldsymbol{x}_{2k-1}) - v)^2 \leq \frac{2n^2}{\mu} \implies \sum_{k=1}^{T/2}(f(\boldsymbol{x}_{2k-1}) - v) \leq \frac{1}{\sqrt{\mu}}T^{1/2}n.$$

Using the same argument as the case of MWU, we then have:

$$DR_T \leq \frac{2}{\sqrt{\mu}} T^{1/2} n = \mathcal{O}(\frac{n}{\sqrt{\mu}} T^{1/2}).$$

□

In the case of the row player using a constant learning rate, LRCA achieves the average dynamic regret of $\mathcal{O}(T^{-1/2})$, better than state-of-the-art no-regret algorithms which obtain the same average but in the normal regret R_T.

In the general case where the column player does not know whether the row player uses the following algorithm to achieve the no-regret property in any situation while maintaining the no-dynamic regret property against the no-regret algorithm of the row player: The idea is to put LCRA on top of another no-regret algorithm. When the regret of LCRA exceeds a certain threshold, we swap to the chosen algorithm. If the row player follows a no-regret algorithm then the LRCA regret will never exceed the threshold; thus we will have no-dynamic regret. By doing that, the column player sacrifices the optimal rate of no-regret in the worst case in order to achieve a much better no-dynamic regret in the case the row player follows a no-regret algorithm. The new Algorithm 12 will have the regret $R_T = O(\sqrt{\log(n)} T^{3/4})$ against random sequence strategies of the row player while maintaining no-dynamic regret against the no-regret algorithm of the row player.

Algorithm 12 Combination of LRCA and Adahedge algorithm

Input: Current iteration t, past feedback $\boldsymbol{x}_{t-1}^\top \boldsymbol{A}$ of the row player, total regret up to time t: R_t
Output: Strategy $\boldsymbol{y}_t$ for the column player
if $R_t \leq \sqrt{\log(n)} t^{3/4}$ **then**
  Follow Algorithm 11 (LRCA)
else
  Follow Adahedge algorithm De Rooij et al. (2014) onwards
end if

2.7 Conclusion

The main focus of this chapter centres around the crucial concept of achieving last round convergence against a strategic adversary in the two-player zero-sum game setting. We demonstrate that, by taking into account the asymmetric goals of the players, a natural method can be implemented to achieve last round convergence, which had previously been unattainable in the symmetric setting. This will strengthen the study of no-regret algorithms in the theoretical community and open to more interesting problems in which last round convergence can be achieved (e.g., see Dinh (2022); Bishop

et al. (2021), Daskalakis and Panageas (2019)). As researchers increasingly embrace no-regret algorithms in online learning, there has been a growing shift towards prioritizing last round convergence of strategies over average convergence. Our approach sheds light on how Nash equilibrium can emerge organically in dynamic environments, and moreover, confers stability upon the system, thereby providing system designers with an advantageous property to leverage. In the next chapter, we extend our understanding of the strategic adversary to a broader setting: online linear optimization.

Chapter 3

Achieving Better Regret Against Strategic Adversary

In this chapter, we study the problem of online learning in the presence of a strategic adversary under the online linear optimization setting. By leveraging the additional knowledge gained from the adversary's behaviour, we develop novel algorithms that offer improved performance guarantees. Firstly, we present the Online Single Oracle algorithm, which combines no-(external) regret algorithms and double oracle from game theory to exploit the strategic adversary with an external regret bound of $\mathcal{O}(\sqrt{k\log(k)T})$ in normal-form game, where the size of the effective strategy set k is often linearly dependent on the support size of the Nash Equilibrium. Secondly, we introduce Accurate Follow the Regularized Leader (AFTRL) and Prod-Best Response (Prod-BR), two new online learning algorithms that intensively leverage this additional knowledge while ensuring no-regret property in the worst-case scenario of having inaccurate extra knowledge. AFTRL achieves $O(1)$ external regret or $O(1)$ forward regret against the strategic adversary, in contrast to the $O(\sqrt{T})$ dynamic regret of Prod-BR. To the best of our knowledge, our algorithm is the first to consider forward regret and achieve $O(1)$ regret against strategic adversaries. When playing zero-sum games with Accurate Multiplicative Weights Update (AMWU), a special case of AFTRL, we achieve last round convergence to the NE. We also provide numerical experiments to support our theoretical results, which demonstrate that our methods offer significantly better regret bounds and rate of last round convergence than the state-of-the-art algorithms, such as Multiplicative Weights Update (MWU) and its optimistic counterpart, OMWU.

3.1 Introduction

No-regret algorithms are popular in the online learning and algorithmic game theory literature due to their attractive worst-case performance guarantees (Cesa-Bianchi and Lugosi, 2006). In particular, using these algorithms to choose the strategies to play provably guarantees the average payoff will not be (significantly) worse than the best-fixed strategy in the hindsight, regardless of the sequences encountered. Due to this property, these no-regret algorithms are commonly used in playing against adversary and solving two-player zero-sum games, in which it will eventually lead to average convergence to a Nash Equilibrium (NE) under self-play settings (Zinkevich et al., 2007; Lanctot et al., 2017; Dinh et al., 2022). However, in order to keep the regret bound small, no-regret algorithms (e.g., Multiplicative Weights Update (Freund and Schapire, 1999), Follow the Regularized Leader (Abernethy et al., 2008) and Mirror Descent (Nemirovskij and Yudin, 1983)) need to keep their learning rate small, leading to a slow change in the strategy profile. This makes the sequence of strategies played by no-regret algorithms predictable since each strategy profile will be correlated to its predecessors. Thus, against a no-regret learning opponent, the loss sequence encountered by the learner/player is not entirely arbitrarily adversarial in each round and therefore the worst-case performance guarantees are too pessimistic for the learner. Therefore, in situations such as playing against no-regret algorithms (a.k.a. strategic adversaries), it is desirable to develop a learning algorithm that can exploit the extra structure while maintaining the no-regret property in the worst-case scenario and answer the question:

Can we exploit strategic adversaries?

Besides aiming for better regret bounds, we are also interested in last round convergence. In more detail, despite extensive literature on no-regret learning, one unsatisfactory result is the average convergence to the NE. That is, in two-player zero-sum games, no-regret algorithms such as Multiplicative Weights Update (MWU) (Freund and Schapire, 1999) or Follow the Regularized Leader (FTRL) (Abernethy et al., 2008) will only lead to average convergence instead of last round convergence to the NE. In fact, recent results in Bailey and Piliouras (2018); Mertikopoulos et al. (2018) show that MWU and FTRL will lead to divergence from the NE in many situations. The average convergence will not only increase the computational and memory overhead but also make things difficult when using a neutral network in the solution process in which averaging is not always possible (Bowling et al., 2015). For game theory and modern applications of online learning in optimization such as training Generative Adversarial Networks (Daskalakis et al., 2018), last round convergence plays a vital role in the process, thus it is crucial to develop algorithms that can lead to last round convergence.

To investigate both of the above-mentioned goals in this chapter, under the setting of online linear optimization, **firstly**, by conducting no-regret analysis (Freund and Schapire,

1999) within the DO framework (McMahan et al., 2003), we propose the *Online Single Oracle (OSO)* algorithm which inherits the key benefits from both sides. It is the first DO method that enjoys the no-regret property and can exploit strategic adversaries during game play. Importantly, OSO achieves a regret of $\mathcal{O}(\sqrt{k\log(k)T})$ where k, the size of effective strategy set, is upper-bounded by the total number of pure strategies n and often $k \ll n$ holds in practice. **Secondly**, we develop a new algorithm, Accurate Follow the Regularized Leader (AFTRL), that can exploit strategic adversaries to achieve $O(1)$ external regret or $O(1)$ forward regret while maintaining state-of-the-art regret bound of $O\left(\sqrt{\sum_{t=1}^{T}\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q^2}\right)$ in the worst-case scenario. We also show the generality of our method by extending the result to another online learning class and propose a new algorithm, Accurate Mirror Descent (AMD) with a similar forward regret bound for it. To the best of our knowledge, we are the first to consider *intensive exploitation* and achieve $O(1)$ *forward regret* against the strategic adversary. **Thirdly**, we explore the idea of (A,B)-Prod algorithm in Sani et al. (2014) and suggest a new algorithm, Prod-Best Response (Prod-BR) that achieves a stronger performance guarantee in our setting. In particular, Prod-BR achieves $O(\sqrt{T})$ dynamic regret against the strategic adversary while maintaining $O(\sqrt{T\log(T)})$ external regret in the worst case. **Fourthly**, in a particular case of AFTRL with entropy regularizer, called Accurate Multiplicative Weights Update (AMWU), we prove that this new algorithm will lead to last round convergence in two-player zero-sum games, thus can be an efficient game-solver in many practical applications. In addition, this provides novel contributions to the last round convergence literature. **Finally**, to demonstrate the practical efficiency of AMWU, we show that our algorithm significantly outperforms MWU and OMWU (Rakhlin and Sridharan, 2013a; Daskalakis and Panageas, 2019) on a number of random matrix games and meta games such as Connect Four or Disc (Czarnecki et al., 2020) by a large margin, achieving smaller average loss, dynamic regret and faster last round convergence.

3.2 Related Work

Online learning against a strategic adversary: Deng et al. (2019) studies a similar setting in which the agent plays against a no-external regret adversary in a repeated game. Under the assumption that the agent knows the game structure (i.e., payoff matrix, player's utility), Deng et al. (2019) suggested a fixed strategy for the agent (through solving an optimization problem) such that the agent can guarantee a Stackelberg value, which is optimal in certain games (e.g., general-sum games). Although the work in Deng et al. (2019) provides a planning solution against no-external regret adversary, it can not be applied in many practical situations in which the environment or game structure is unknown (i.e., the agent can not calculate the Stackelberg strategy in advance) or the adversary does not follow no-regret algorithms (i.e., there is no performance guarantee against general adversary). Chiang et al. (2012) and Rakhlin and Sridharan (2013a)

study a different setting in which the agent has access to the prediction M_t of $\boldsymbol{x}_t$ before making a decision at round t. [1] The new algorithm, Optimistic Follow the Regularized Leader (OFTRL), has the external regret that depends linearly on $\sqrt{\sum_{t=1}^{T} \|\boldsymbol{x}_t - M_t\|_*^2}$. However, with an accurate prediction (i.e., $M_t \approx \boldsymbol{x}_t$), one could expect a stronger performance guarantee rather than no-external regret of OFTRL. Intuitively, since OFTRL sets a fixed weight 1 for prediction M_t [2], it restricts the advantage of the extra knowledge in the learning process. Our new algorithms (AFTRL and AMD) generalize the work of Rakhlin and Sridharan (2013a) to further exploit the extra knowledge in the learning process while maintaining a no-forward regret property (Saha et al., 2012) in the worst-case scenario.

Last round convergence: While average convergence of no-regret learning dynamics has been studied extensively in game theory and online learning communities (e.g., Freund and Schapire (1999); Cesa-Bianchi and Lugosi (2006)), last round convergence has only been a topic of research in the last few years due to its application in game theory and optimization. This started with the negative results of Bailey and Piliouras (2018); Mertikopoulos et al. (2018), who showed that in games with interior equilibria, if the agents use MWU, then the last round strategy moves away from the NE and towards the boundary. More recently, Daskalakis and Panageas (2019); Wei et al. (2020) proved that in a two-player zero-sum game with unique NE, if both players follow a variant of MWU, called optimistic multiplicative weight update (OMWU), then the dynamic will converge in the last round to the NE. In the asymmetric setting, Dinh et al. (2021) proposed last round convergence in asymmetric games algorithm (LRCA), which requires one agent to have an estimate of the minimax equilibrium and therefore limits the use of the algorithm. In our work, we prove that our method AMWU will converge in last round to the NE of a two-player zero-sum game without such a requirement, and it does this faster than OMWU and MWU.

3.3 Problem Formulations & Preliminaries

We consider the online linear optimization setting in which at round t, the learner chooses a strategy $\boldsymbol{f}_t \in \mathcal{F}$, where $\mathcal{F} \subset [0,1]^n$ [3] is a convex compact set. Simultaneously, the environment reviews a loss vector $\boldsymbol{x}_t \in [0,1]^n$ and the learner suffers the loss: $\langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle$. The goal of the learner is to minimize the total loss after T rounds: $\min_{\boldsymbol{f}_1,\dots \boldsymbol{f}_T} \sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle$, which can be translated into minimizing the following dynamic regret:

[1] As we prove in Lemma 3.13, playing against strategic adversary can result in an accurate prediction of $\boldsymbol{x}_t$.

[2] The exploiting rate α in Algorithm 15.

[3] All the results remains true for bounded domain of strategy and loss vector.

Definition 3.1 (**Dynamic Regret** Besbes et al. (2015))**.** The dynamic regret is defined as:

$$DR_T := \sum_{t=1}^{T} \left(\langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \operatorname*{argmin}_{\boldsymbol{g}_t \in \mathcal{F}} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \right).$$

In situations where there is no knowledge about $\boldsymbol{x}_t$, it is often impossible to achieve no-dynamic regret. Thus, it is more tractable to aim for no-external regret (Cesa-Bianchi and Lugosi, 2006):

Definition 3.2 (**No-external regret**)**.** Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ be a sequence of mixed losses played by the environment. An algorithm of the learner that generates a sequence of mixed strategies $\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots$ is called a *no-external regret* algorithm if we have

$$\lim_{T\to\infty} \frac{R_T}{T} = 0, \text{ where } R_T := \min_{\boldsymbol{f}\in\mathcal{F}} \sum_{t=1}^{T} \left(\langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle \right).$$

The well-known Multiplicative Weights Update (Freund and Schapire, 1999) has this no-external regret property:

Definition 3.3 (Multiplicative Weights Update)**.** Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ be a sequence of loss vectors followed by the environment. The learner is said to follow MWU if $\boldsymbol{f}_{t+1}$ is updated as follows:

$$\boldsymbol{f}_{t+1}(i) = \frac{\boldsymbol{f}_t(i) \exp(-\mu_t \boldsymbol{a}^{i\top} \boldsymbol{x}_i)}{\sum_{i=1}^{n} \boldsymbol{f}_t(i) \exp(-\mu_t \boldsymbol{a}^{i\top} \boldsymbol{x}_i)}, \ \forall i \in [n] \tag{3.1}$$

where $\mu_t > 0$ is a parameter, $\boldsymbol{f}_0 = [1/n, \ldots, 1/n]$ and n is the number of pure strategies (a.k.a. experts).

When T is known in advance, by fixing the learning rate $\mu_t = \sqrt{8\log(n)/T}$, we can achieve the optimal regret bound for MWU (Theorem 2.2 in Cesa-Bianchi and Lugosi (2006)):

$$\sqrt{T\log(n)/2}.$$

When T is unknown, we can apply the Doubling Trick to achieve the regret bound of

$$(\sqrt{2}/(\sqrt{2}-1))\sqrt{T\log(n)/2},$$

which is worse than the optimal one by a factor of $\sqrt{2}/(\sqrt{2}-1)$. De Rooij et al. (2014) proposed AdaHedge, a variant of MWU with adaptive learning rate $\mu_t = \log(n)/\Delta_{t-1}$ where Δ_t denotes the cumulative mixability gap [4]. Then following Theorem 8 in De Rooij et al. (2014), the regret for AdaHedge will be bounded by

$$\sqrt{T\log(n)} + 16/3\log(n) + 2,$$

[4] See Appendix A.2 and A.3 for more details about Doubling Trick and AdaHedge algorithm.

which is the worse than the optimal one by a factor of $\sqrt{2}$.

In our work, w.l.o.g we use the optimal regret bound of MWU when T is known to derive our theoretical results. In the case T is unknown, following exactly the same argument with the Doubling Trick or AdHedge algorithm, we can derive similar regret bound up to a constant factor for our algorithms.

In our work, since we assume the learner has extra knowledge about the adversary, the learner can achieve a stronger notion of performance, compared to the conventional no-external regret, namely:

Definition 3.4 (**Forward Regret** Saha et al. (2012))**.** The forward regret is defined as:

$$FR_T := \sum_{t=1}^{T} \left(\langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \right), \text{where } \boldsymbol{g}_{t+1} = \operatorname*{argmin}_{\boldsymbol{g} \in \mathcal{F}} G_{t+1}(\boldsymbol{g}) = \langle \boldsymbol{g}, \sum_{s=1}^{t} \boldsymbol{x}_s + \boldsymbol{x}_{t+1} \rangle + \frac{R(\boldsymbol{g})}{\eta}.$$

In particular, the following lemma implies that if an algorithm has no-forward regret property, then it is a no-external regret algorithm as well, but not vice versa.

Lemma 3.5. *Let $\boldsymbol{g}_t$ be defined as above, then the following relationship holds for any $\boldsymbol{f} \in \mathcal{F}$:*

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \leq \langle \boldsymbol{f}, \sum_{t=1}^{T} \boldsymbol{x}_t \rangle + \frac{R(\boldsymbol{f})}{\eta}.$$

(We provide the full proof in Appendix 3.25).

In Section 3.7, we study a simpler form of online linear optimization, a two-player zero-sum normal-form game, which is often described by a payoff matrix $\boldsymbol{A}$ of size $n \times m$. The rows and columns of $\boldsymbol{A}$ are the pure strategies of the row and the column players, respectively, and we consider n and m to be prohibitively large numbers. We denote the set of pure strategies for the row player as $\Pi := \{\boldsymbol{a}^1, \boldsymbol{a}^2, \dots \boldsymbol{a}^n\}$, and $C := \{\boldsymbol{c}^1, \boldsymbol{c}^2, \dots, \boldsymbol{c}^m\}$ for the column player. We consider $\boldsymbol{A}_{i,j} \in [0, 1]$ to represent the (normalised) loss of the row player when playing a pure strategy $\boldsymbol{a}^i$ against the pure strategy $\boldsymbol{c}^j$ of the column player. The set of mixed strategies for the row-player is $\Delta_n := \left\{ \boldsymbol{f} | \boldsymbol{f} = \sum_{i=1}^{n} x_i \boldsymbol{a}^i, \ \sum_{i=1}^{n} x_i = 1, x_i \geq 0, \forall i \in [n] \right\}$, and for the column player it is $\Delta_m := \{\boldsymbol{y} | \boldsymbol{y} = \sum_{i=1}^{m} y_i \boldsymbol{c}^i, \sum_{i=1}^{n} y_i = 1, y_i \geq 0, \forall i \in [m]\}$. The support of a mixed strategy is written as $\text{supp}(\boldsymbol{f}) := \{\boldsymbol{a}^i \in \Pi | x_i \neq 0\}$, with its size being $|\text{supp}(\boldsymbol{f})|$. At the t-th round, the expected payoff for the joint-strategy profile $(\boldsymbol{f}_t \in \Delta_n, \boldsymbol{y}_t \in \Delta_m)$ is $(-\boldsymbol{f}_t^\top \boldsymbol{A} \boldsymbol{y}_t, \boldsymbol{f}_t^\top \boldsymbol{A} \boldsymbol{y}_t)$.

The NE in two-player zero-sum game $\boldsymbol{A}$ can be expressed by John von Neumann's minimax theorem (Neumann, 1928):

$$\max_{\boldsymbol{y} \in \Delta_m} \min_{\boldsymbol{f} \in \Delta_n} \boldsymbol{f}^\top \boldsymbol{A} \boldsymbol{y} = \min_{\boldsymbol{f} \in \Delta_n} \max_{\boldsymbol{y} \in \Delta_m} \boldsymbol{f}^\top \boldsymbol{A} \boldsymbol{y} = v \tag{3.2}$$

for some $v \in \mathbb{R}$. The point $(\boldsymbol{f}^*, \boldsymbol{y}^*)$ satisfying Equation (3.2) is the NE of the game.

Following the minimax theorem, it is well-known that the minimax equilibrium can be calculated by solving a linear programming problem. However, when the game size is large, it is still computationally expensive to solve the exact NE. The DO method (McMahan et al., 2003) approximates a NE in large-scale zero-sum games by iteratively expanding and solving a series of sub-games (i.e., games with a restricted set of pure strategies). Since the sets of pure strategies of the sub-game are often much smaller than the original game, the NE of the sub-games can be easily solved via approaches such as FP. Based on the NE of the sub-game, each player finds the best response to said NE, and expands their strategy set with this best response. PSRO methods (Lanctot et al., 2017; McAleer et al., 2020) are a generalisation of DO in which RL methods (e.g., Actor-Critic) are adopted to approximate the best-response strategy. In the worst case scenario (e.g., the support size of NE is large), DO may end up restoring the original game and will maintain no advantages over LP solutions.

Although DO can solve large-scale zero-sum games, it requires both players to *coordinate* by finding a NE in the sub-games; this is a problem for DO when applied in real-world scenarios, as it cannot exploit the opponent who can play any non-stationary strategy. OSO addresses this problem by combining DO with tools in online learning.

3.4 Online Single Oracle

In this section, we introduce Online Single Oracle (OSO), a no-regret algorithm followed by individual players that can strategically exploit any non-stationary opponent unlike DO. Compared to the MWU algorithm, OSO can be applied to strategizing against a strategic adversary as it only considers a smaller subset of the full pure strategy space.

This section is organised as follows: we start by introducing OSO and deriving its regret bound. We then discuss the bound on the effective strategy set k, the key element in the regret bound of OSO. Finally, we set out two different questions on the effectiveness and efficiency of the best-response oracle, and analyse OSO's performance when the player only has access to *less-frequent* or *approximate* best-responses oracles.

3.4.1 Online Single Oracle Algorithm

One can think of OSO as an online counterpart to the *Single Oracle* in DO (McMahan et al., 2003) which can achieve the no-regret property. In contrast to classical no-regret algorithms such as MWU (Freund and Schapire, 1999) where the whole set of pure strategies needs considering at each iteration, i.e., Equation (3.1), we propose OSO that only considers a *subset* of the whole strategy set. The key operation is that, at each

Algorithm 13 Online Single Oracle Algorithm

1: **Input:** Player's pure strategy set Π
2: Init. effective strategies set: $\Pi_0 = \Pi_1 = \{\boldsymbol{a}^j\}, \boldsymbol{a}^j \in \Pi$
3: **for** $t = 1$ to T **do**
4: **if** $\Pi_t = \Pi_{t-1}$ **then**
5: Compute $\boldsymbol{f}_t$ by the MWU in Equation (3.1)
6: **else if** $\Pi_t \neq \Pi_{t-1}$ **then**
7: Start a new time window T_{i+1} and
Reset $\boldsymbol{f}_t = \big[1/|\Pi_t|, \ldots, 1/|\Pi_t|\big]$, $\bar{\boldsymbol{x}} = \mathbf{0}$
8: **end if**
9: Observe $\boldsymbol{x}_t$ and update the average loss in T_i: $\bar{\boldsymbol{x}} = \sum_{t \in T_i} \boldsymbol{x}_t / |T_i|$
10: Calculate the best-response: $\boldsymbol{a}_t = \arg\min_{\boldsymbol{f} \in \Pi} \langle \boldsymbol{f}, \bar{\boldsymbol{x}} \rangle$
11: Update the set of strategies: $\Pi_{t+1} = \Pi_t \cup \{\boldsymbol{a}_t\}$
12: **end for**
13: **Output:** $\boldsymbol{f}_T$, Π_T

round t, OSO only considers adding a new strategy if it is the best response to the average loss in a time window (defined formally in the following paragraph). As such, OSO can save on exploration costs by ignoring the pure strategies that have never been the best-response to any, so far observed, average losses, $\bar{\boldsymbol{x}}$.

Our OSO is listed in Algorithm 13. We initialise the OSO algorithm with a random strategy subset Π_0 from the original strategy set Π. Without loss of generality, we assume that Π_0 starts from only one pure strategy (line 2). We call subset Π_t the **effective strategy set** at the timestep t, and define the period of consecutive iterations as one **time window** T_i in which the effective strategy set stays fixed, i.e., $T_i := \big\{t \;\big|\; |\Pi_t| = i\big\}$. At iteration t, we update $\boldsymbol{f}_t$ (line 5) whilst only considering the effective strategy set Π_t (rather than whole set Π); and the best-response is computed against the average loss $\bar{\boldsymbol{x}}$ within the current time window T_i (line 9). Adding a new best-response that is not in the existing effective strategy set will start a new time window (line 7). Notably, despite the design of effective strategy sets, the exact best-response oracle in line 10 still needs to search over the whole strategy set Π, which is a property that we relax through best-response approximation later.

We now present the regret bound of OSO as follows,

Theorem 3.6 (Regret Bound of OSO)**.** *Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T$ be a sequence of loss vectors played by an adversary, and $\langle \cdot, \cdot \rangle$ be the dot product, OSO in Algorithm 13 is a no-regret algorithm with*

$$\frac{1}{T}\Big(\sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{f} \in \Pi} \sum_{t=1}^{T} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle\Big) \leq \frac{\sqrt{k \log(k)}}{\sqrt{2T}},$$

where $k = |\Pi_T|$ is the size of the effective strategy set in the final time window.

Proof. W.l.o.g, we assume the player uses the MWU as the no-regret algorithm and starts with only one pure strategy in Π_0 in Algorithm 13. Since in the final time

window, the effective strategy set has k elements, there are exactly k time windows. Denote $|T_1|, |T_2|, \ldots, |T_k|$ be the lengths of time windows during each of which the subset of strategies the no-regret algorithm considers does not change. In the case of finite set of strategies, k will be finite and we have

$$\sum_{i=1}^{k} |T_k| = T.$$

In the time window with length $|T_i|$, following the regret bound of MWU in Definition 3.3 we have

$$\sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{f} \in \Pi_{|\bar{T}_i|+1}} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle \leq \sqrt{\frac{|T_i|}{2} \log(i)}, \quad \text{where } |\bar{T}_i| = \sum_{j=1}^{i-1} |T_j|. \tag{3.3}$$

In the time window T_i, we consider the full strategy set when we calculate the best response strategy in step 11 of Algorithm 13 and it stays in $\Pi_{|\bar{T}_i|+1}$. Therefore, the inequality (3.3) can be expressed as

$$\sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{f} \in \Pi} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle \leq \sqrt{\frac{|T_i|}{2} \log(i)}. \tag{3.4}$$

Sum up the inequality (3.4) for $i = 1, \ldots k$ we have

$$\begin{aligned}
&\sum_{i=1}^{k} \sqrt{\frac{|T_i|}{2} \log(i)} \geq \sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \sum_{i=1}^{k} \min_{\boldsymbol{f} \in \Pi} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle \\
&\geq \sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{f} \in \Pi} \sum_{i=1}^{k} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle = \sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{f} \in \Pi} \sum_{t=1}^{T} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle.
\end{aligned} \tag{3.5a}$$

Inequality (3.5a) is due to $\sum \min \leq \min \sum$. Using the Cauchy-Schwarz inequality we have

$$\sum_{i=1}^{k} \sqrt{\frac{|T_i|}{2} \log(i)} \leq \sqrt{(\sum_{i=1}^{k} \frac{|T_i|}{2})(\sum_{i=1}^{k} log(i))} = \sqrt{\frac{T}{2}(\sum_{i=1}^{k} log(i))} \leq \sqrt{\frac{Tk \log(k)}{2}}.$$

Along with Inequality (3.5a) we can derive the regret

$$\sqrt{\frac{Tk \log(k)}{2}} \geq \sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{f} \in \Pi} \sum_{t=1}^{T} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle.$$

□

We note here that in line 7 of Algorithm 13, each time OSO enters a new time window, it sets equal weight for every pure strategy in the current effective strategy. Since we

assume a fully adversarial environment, the historical data that the agent learnt in the previous time window does not provide any advantages over the current time window, thus in order to avoid any exploitation, the agent needs to reset the strategy as stated in Algorithm 13. In situations where priority knowledge can be observed through historical data, our OSO algorithm can exploit this knowledge by updating the starting strategy in each time window. We leave this important extension to our future work.

Remark 3.7 (Worst-Case Regret Bound). *Similar to all existing DO type of methods, in the worst-case scenario, OSO has to find all pure strategies, i.e.,* $k = |\Pi|$. *Thus, the regret in the worst case scenario will be:* $\sqrt{|\Pi| \log(|\Pi|)}/\sqrt{2T}$. *However, we believe* $k \ll |\Pi|$ *holds in many practical cases such as against the strategic adversary. In later sections, we provide both theoretical and empirical evidence that real-world games tend to have* $k \ll |\Pi|$.

In the next section, we discuss the relationship between the effective strategy set size k and the full game size.

3.4.2 Size of Effective Strategy Set k

Heuristically, the practical success of OSO and other discussed methods (e.g., DO (McMahan et al., 2003) / PSRO (Lanctot et al., 2017)) is based on the assumption that the support size of the NE is small. Intuitively, since OSO is a no-regret algorithm, should the adversary itself follow a no-regret algorithm, the adversary's average strategy would converge to the NE. Thus, the learner's best-responses with respect to the average loss will include all the pure strategies in the support of the learner's NE. Therefore, under the assumption of DO and PSRO that the support size of NE is small, the effective strategy set size k is potentially a far smaller number than the game size (i.e., n).

Note that the assumption of a NE having a small support size holds true in many situations. In **symmetric** games with random entries (i.e., see Theorem 2.8 in Jonasson et al. (2004)), it has been proved that the expected support size of a NE will be $(\frac{1}{2} + \mathcal{O}(1))n$ where n is the game size; showing that the support size of a NE strategy is only half of the game size. In **asymmetric** games with disproportionate action spaces (e.g., $n \gg m$), we provide the following lemma under which the support of a NE is small.

Lemma 3.8. *In asymmetric games* $\boldsymbol{A}_{n\times m}, n \gg m$, *if the NE* $(\boldsymbol{f}^*, \boldsymbol{y}^*)$ *is unique, then the support size of the NE will follow* $|\operatorname{supp}(\boldsymbol{f}^*)| = |\operatorname{supp}(\boldsymbol{y}^*)| \leq m$.

Proof. Since the size of $\boldsymbol{f}^*$ and $\boldsymbol{y}^*$ are n and m respectively, the size of the support of NE can not exceed the size of the game

$$|\text{support}(\boldsymbol{f}^*)| \leq n;\ |\text{support}(\boldsymbol{y}^*)| \leq m.$$

In the case the game $\boldsymbol{A}$ has a unique Nash equilibrium, following Theorem 1 in Bohnenblust et al. (1950), we have

$$|\text{support}(\boldsymbol{f}^*)| = |\text{support}(\boldsymbol{y}^*)| \leq \min(n, m) = m.$$

Thus, we have proved the lemma. □

In the situation where an asymmetric game $\boldsymbol{A}_{n\times m}$, $n \gg m$ does not has unique NE but it is nondegenerate [5], then following Proposition 3.3 in Roughgarden (2010), we can similarly bound the support of NE by m.

In the case when a dominant strategy exists, we can theoretically bound k by the following lemma:

Definition 3.9 (Strictly Dominant Strategy)**.** A strategy $\hat{\boldsymbol{f}}$ is called a strictly dominant strategy for the row player if:

$$\hat{\boldsymbol{f}}^\top \boldsymbol{A}\boldsymbol{y} < \boldsymbol{f}^\top \boldsymbol{A}\boldsymbol{y} \quad \forall \boldsymbol{f} \in \Pi, \boldsymbol{y} \in C.$$

Lemma 3.10. *Suppose there exists a strictly dominant strategy for the player, then the size of the effective strategy set will be bounded by 2.*

Proof. First we show that a strictly dominant strategy in two-player zero-sum game is a pure strategy. Let $\hat{\boldsymbol{f}}$ be the strictly dominant strategy. By definition of strictly dominant strategy we have

$$\hat{\boldsymbol{f}}^\top \boldsymbol{A}\boldsymbol{y} < \boldsymbol{f}^\top \boldsymbol{A}\boldsymbol{y} \quad \forall \boldsymbol{f} \in \Pi, \boldsymbol{y} \in C.$$

Let $\boldsymbol{a}^1$ be a pure strategy such that

$$\boldsymbol{a}^1 = \underset{\boldsymbol{a}\in\Pi}{\text{argmin}}\, \boldsymbol{a}^\top \boldsymbol{A}\boldsymbol{y}^1,$$

where $\boldsymbol{c}^1$ is a constant vector. If $\hat{\boldsymbol{f}}$ is a mixed strategy then we have $\hat{\boldsymbol{f}} \neq \boldsymbol{a}^1$ and

$$\hat{\boldsymbol{f}}^\top \boldsymbol{A}\boldsymbol{c}^1 \geq \boldsymbol{a}^{1\top} \boldsymbol{A}\boldsymbol{c}^1,$$

contradicts with the definition of strictly dominant strategy. Thus, the strictly dominant strategy in two-player zero-sum game is a pure strategy [6].

Now, after the first iteration, the OSO algorithm will add the best response to the effective strategy set. Since there exists a strictly dominant strategy $\hat{\boldsymbol{f}}$ and it is a pure strategy, $\hat{\boldsymbol{f}}$ will be added to the effective strategy set. From the second iteration, since

[5] No mixed strategy of support size h has more than h pure best responses

[6] With the same argument, a strictly dominant strategy in any normal-form game is a pure strategy.

the strictly dominant strategy $\hat{\boldsymbol{f}}$ is already in the effective strategy set, the best response to any average loss is always in the effective strategy set. Thus, the effective strategy set will not be expanded after iteration 2. In other words, the size of the effective strategy set will be bounded by 2. □

Despite the practical success of our method and the DO/PSRO lines of work, there is no theoretical guarantee about the relationship between the support size of a NE and the performance of the algorithm. In this chapter, we provide a negative result by constructing an example such that the size of the effective strategy set equals the size of the full strategy set, even when the support of NE is small.

Lemma 3.11. *Suppose the players start with the entry $\boldsymbol{A}_{1,1}$ and the game matrix $\boldsymbol{A}$ of the two-players zero-sum game is designed such as*

$$\boldsymbol{A}_{i,i} = 0.5 + \frac{0.1i}{n}\ \forall i \in [n];\quad \boldsymbol{A}_{i,i+1} = 0.9\ \forall i \in [n-1],$$
$$\boldsymbol{A}_{i,j} = 0.8\ \forall j \geq i+2, i \in [n],\quad \boldsymbol{A}_{i,j} = \boldsymbol{A}_{i,i} + \frac{0.1}{2n} \forall j \leq i, i \in [n],$$

where n is the size of the pure strategy set for both players. Then the game has a unique Nash equilibrium with support size of 1 (i.e., the entry $\boldsymbol{A}_{n,n}$) and the effective strategy set in both DO and OSO will reach the size of pure strategy set, that is, $k = n$.

We provide the full proof in Appendix 3.21. The idea is that the the matrix $\boldsymbol{A}$ is designed such that the sub-game NE will change from $\boldsymbol{A}_{i,i}$ to $\boldsymbol{A}_{i,i+1}$ for $i \in [n]$, thus OSO will need to consider the full pure strategy set before reaching the game NE at $\boldsymbol{A}_{n,n}$. Following the same argument, when the players start with the entry $\boldsymbol{A}_{i,i}$, the effective strategy set will be $n - i + 1$ and thus when the players choose the starting entry as uniformly random, the expected size of effective stratetgy set will be: $\mathbb{E}(k) = (n+1)/2$. We would like to highlight that this negative result not only applies to our method, but also to **all** existing DO/PSRO algorithms and their variations.

However, as described in our experiments, we find that the extreme situation shown in Lemma 3.11 rarely occurs in practice. Later in Figure 3.1, we provide empirical evidence to support our claim that $k \ll |\Pi|$ and that there exists a linear relationship between k and the Nash support size in many real-world applications.

3.4.3 OSO with Less-Frequent Best-Response

The first adaptation to the best-response process that we consider is to make calls to the best-response oracle less frequently. Obtaining a best-response strategy can be computationally expensive (Vinyals et al., 2019), and OSO considers adding a new best-response strategy at every iteration. A practical solution is to consider adding a new strategy when the regret in the current time window exceeds a predefined threshold

Algorithm 14 OSO with Less-Frequent Best Response

Input: A set Π pure strategy set of player
$\Pi_0 := \Pi_1$: initial set of effective strategies
for $t = 1$ to ∞ **do**
  if $\Pi_t = \Pi_{t-1}$ **then**
    Following the MWU update in Equation (3.1)
  else if $\Pi_t \neq \Pi_{t-1}$ **then**
    Start a new time window T_{i+1}
    Reset the MWU update in Equation (3.1) with a new initial strategy $\boldsymbol{f}_t$
  end if
  Observe $\boldsymbol{x}_t$ and update the average loss in the current time window T_i
    $\bar{\boldsymbol{x}} = \frac{1}{|T_i|}\sum_{\boldsymbol{f}_t \in T_i} \boldsymbol{x}_t$
  Calculate the best response:
    $\boldsymbol{a} = \arg\min_{\boldsymbol{f}\in\Pi}\langle \boldsymbol{f}, \bar{\boldsymbol{x}}\rangle$,
  if $\min_{\boldsymbol{f}\in\Pi_{|\bar{T}_i|+1}}\langle \boldsymbol{f}, \sum_{j=|\bar{T}_i|}^{t} \boldsymbol{x}_j\rangle - \min_{\boldsymbol{f}\in\Pi}\langle \boldsymbol{f}, \sum_{j=|\bar{T}_i|}^{t} \boldsymbol{x}_j\rangle \geq \alpha^i_{t-|\bar{T}_i|}$ **then**
    Update the strategy set: $\Pi_{t+1} = \Pi_t \cup \boldsymbol{a}$
  else
    $\Pi_{t+1} = \Pi_t$
  end if
  Output the strategy $\boldsymbol{f}_t$ at round t for the player
end for

α. To make OSO account for this, we denote $|\bar{T}_i| := \sum_{h=1}^{i-1} |T_h|$ as the starting point of the time window T_i, and write the threshold at T_i as $\alpha^i_{t-|\bar{T}_i|}$ where $t - |\bar{T}_i|$ denotes the relative position of round t in the time window T_i. We can make OSO add a new strategy only when the following condition is satisfied:

$$\min_{\boldsymbol{f}\in\Pi_t} \left\langle \boldsymbol{f}, \sum_{j=|\bar{T}_i|}^{t} \boldsymbol{x}_j \right\rangle - \min_{\boldsymbol{f}\in\Pi} \left\langle \boldsymbol{f}, \sum_{j=|\bar{T}_i|}^{t} \boldsymbol{x}_j \right\rangle \geq \alpha^i_{t-|\bar{T}_i|}. \tag{3.6}$$

Note that the larger the threshold α, the longer OSO takes to add a new strategy into Π_t. However, choosing a large α will prevent the learner from acquiring the actual best-response, thus increasing the total regret R_T by α. In order to maintain the no-regret property, the α needs to satisfy

$$\lim_{T\to\infty} \frac{\sum_{i=1}^{k} \alpha^i_{T_i}}{T} = 0. \tag{3.7}$$

One choice of α that satisfies Equation (3.7) is $\alpha^i_{t-|\bar{T}_i|} = \sqrt{t - |\bar{T}_i|}$. We show that with the chosen α, the regret bound of Algorithm 14 will be $\mathcal{O}\big(\sqrt{k\log(k)/T}\big)$ (The full proof is given in Appendix 3.22).

3.4.4 Considering ϵ-Best Responses

The second adaptation brings OSO more closely in line with the work of PSRO by considering a non-exact best-response oracle. So far, OSO agents compute the exact best-response to the average loss function $\bar{\boldsymbol{x}}$ (i.e., line 10 in Algorithm 13). Since calculating the exact best-response is often infeasible in large games, an alternative way is to consider an ϵ-best response (e.g., through a RL subroutine similar to PSRO (Lanctot et al., 2017)) to the average loss. By first analysing the convergence of DO, we can derive the regret bound as well as convergence guarantees for an OSO learner in the case of an ϵ-best response oracle.

Theorem 3.12. *Suppose an OSO agent can only access the ϵ-best response in each iteration when following Algorithm 13, if the adversary is strategic, then the average strategy of the agent will converge to an ϵ-NE. Furthermore, the algorithm is ϵ-regret:*

$$\lim_{T\to\infty} \frac{R_T}{T} \leq \epsilon; \quad R_T = \max_{\boldsymbol{f}\in\Delta_\Pi} \sum_{t=1}^{T} \left(\boldsymbol{f}_t^\top \boldsymbol{A}\boldsymbol{y}_t - \boldsymbol{f}^\top \boldsymbol{A}\boldsymbol{y}_t \right).$$

(We provide the full proof in Appendix 3.24).

Theorem 3.12 justifies that in the case of approximate best-responses, OSO learners can still approximately converge to a NE. This results allows for the application of optimisation methods to approximate the best response, which paves the way to use RL algorithm in solving complicated zero-sum games such as StarCraft (Vinyals et al., 2019).

The effectiveness of OSO against a strategic adversary is contingent upon the ability to observe a small effective strategy set. In the following sections, we will introduce another observation that can be leveraged by the learner to exploit a strategic adversary, specifically, the adversary's gradual change in updates.

3.5 Accurate Follow the Regularized Leader

In order to have a no-(external) regret property, popular algorithms such FTRL and OMD need to have small learning rate η (i.e., see (Shalev-Shwartz, 2012)): $\eta = O(\frac{1}{\sqrt{T}})$. From this observation, we can prove the following lemma, which plays an important role in our analyses:

Lemma 3.13. *Let $\boldsymbol{f}_t$, $\boldsymbol{f}_{t+1}$ be two consecutive strategies of no-external regret algorithms (i.e., FTRL, OMD). Then we have for any norm $\|.\|_q$:*

$$\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_q = O(\frac{1}{\sqrt{T}}).$$

The full proof is given in Appendix 3.27.

Now, let R be β-strongly convex function with respect to $\|.\|_p$ norm. W.l.o.g. we assume that $\min_{\boldsymbol{f}\in\mathcal{F}} R(\boldsymbol{f}) = 0$.

Algorithm 15 Accurate Follow the Regularized Leader

Input: learning rate $\eta > 0$, exploiting rate $\alpha \geq 1$, $\boldsymbol{f}_1 = \operatorname{argmin}_{\boldsymbol{f}\in\mathcal{F}} R(\boldsymbol{f})$

Output: next strategy update

$$\boldsymbol{f}_{t+1} = \operatorname*{argmin}_{\boldsymbol{f}\in\mathcal{F}} F_{t+1}(\boldsymbol{f}) = \langle \boldsymbol{f}, \sum_{s=1}^{t} \boldsymbol{x}_s + \alpha \boldsymbol{x}_t \rangle + \frac{R(\boldsymbol{f})}{\eta}$$

The Accurate Follow the Regularized Leader algorithm (AFTRL) contains two important parameters: the exploiting rate α and the learning rate η. While the learning rate η stabilizes the strategy update to avoid exploitation, the exploiting rate α measures the relative weight between the historical data $\sum_{s=1}^{t} \boldsymbol{x}_s$ and the prediction $\boldsymbol{x}_t$. Intuitively, with an accurate prediction $\boldsymbol{x}_t$, a large α will boost the performance of AFTRL since $\boldsymbol{x}_t$ describes the next loss vector $\boldsymbol{x}_{t+1}$ better compared to the historical data $\sum_{s=1}^{t} \boldsymbol{x}_s$. Varying α provides different algorithms in the literature. With $\alpha = 0$, the algorithm becomes the classical FTRL (Abernethy et al., 2008). With $\alpha = 1$, AFTRL recovers the optimistic FTRL method (OFTRL) of Rakhlin and Sridharan (2013a). We can have the following regret bound of the AFTRL algorithm:

Theorem 3.14. *Let $\mathcal{F} \subset [0,1]^n$ be a convex compact set and let R be a β-strongly convex function with respect to $\|.\|_p$ norm and $\min_{\boldsymbol{f}\in\mathcal{F}} R(\boldsymbol{f}) = 0$. Denote $\|.\|_q$ the dual norm with $1/p + 1/q = 1$. Then the AFTRL achieves the external regret of $O(1)$ or forward regret of $O\Big(\sqrt{\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2}\Big)$ against general adversary. More importantly, against a strategic adversary (i.e., no-external regret algorithms such that FTRL, OMD), AFTRL achieves $O(1)$ external regret or $O(1)$ forward regret.*

Proof Sketch. We first prove that for any strategy $\boldsymbol{f}'$ of the environment, AFTRL satisfies:

$$\sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \frac{1}{\alpha} \langle \boldsymbol{f}', \sum_{t=1}^{T} \boldsymbol{x}_t \rangle - \frac{\alpha - 1}{\alpha} \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \leq \frac{1}{\eta\alpha} R(\boldsymbol{f}') + \frac{\eta\alpha}{\beta} \sum_{t=1}^{T} (||\boldsymbol{x}_t - \boldsymbol{x}_{t-1}||_q)^2. \tag{3.8}$$

Define $\boldsymbol{h}_{t+1}$ as follows: $\boldsymbol{h}_{t+1} = \operatorname{argmin}_{\boldsymbol{f}\in\mathcal{F}} H_{t+1}(\boldsymbol{f}) = \langle \boldsymbol{f}, \sum_{s=1}^{t} \boldsymbol{x}_s + \alpha \boldsymbol{x}_{t+1} \rangle + \frac{R(\boldsymbol{f})}{\eta}$.

Intuitively, the strategy $\boldsymbol{h}_{t+1}$ will perform much better than the normal FTRL since the agent can observe one step ahead the strategy of the adversary. Note that we can

decompose the total loss of the agent as follows

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle = \sum_{t=1}^{T}\langle \boldsymbol{f}_t - \boldsymbol{h}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t-1}\rangle + \sum_{t=1}^{T}\langle \boldsymbol{f}_t - \boldsymbol{h}_t, \boldsymbol{x}_{t-1}\rangle + \sum_{t=1}^{T}\langle \boldsymbol{h}_t, \boldsymbol{x}_t\rangle. \tag{3.9}$$

The key step of the proof is that we can prove by induction:

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t - \boldsymbol{h}_t, \boldsymbol{x}_{t-1}\rangle + \sum_{t=1}^{T}\langle \boldsymbol{h}_t, \boldsymbol{x}_t\rangle \le \frac{1}{\alpha}\langle \boldsymbol{f}', \sum_{t=1}^{T}\boldsymbol{x}_t\rangle + \frac{\alpha-1}{\alpha}\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle + \frac{1}{\eta\alpha}R(\boldsymbol{f}'),\ \forall \boldsymbol{f}' \in \mathcal{F}. \tag{3.10}$$

Furthermore, using the property of β-strongly convex function, we can derive:

$$\begin{aligned} &\|\boldsymbol{x}_{t-1} - \boldsymbol{x}_t\|_q \ge \frac{\beta}{\eta\alpha}\|\boldsymbol{h}_t - \boldsymbol{f}_t\|_p \\ &\implies \sum_{t=1}^{T}\langle \boldsymbol{f}_t - \boldsymbol{h}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t-1}\rangle \le \sum_{t=1}^{T}||\boldsymbol{f}_t - \boldsymbol{h}_t||_p||\boldsymbol{x}_t - \boldsymbol{x}_{t-1}||_q \le \frac{\eta\alpha}{\beta}\sum_{t=1}^{T}(||\boldsymbol{x}_{t-1} - \boldsymbol{x}_t||_q)^2. \end{aligned} \tag{3.11}$$

Using Inequality (3.10) and (3.11) in Equality (3.9) we derive the Inequality (3.8).

Let $\boldsymbol{f}^* = \text{argmin}_{\boldsymbol{f}'\in\mathcal{F}}\langle \boldsymbol{f}', \sum_{t=1}^{T}\boldsymbol{x}_t\rangle$ and $\boldsymbol{R} = \max_{\boldsymbol{f}\in\mathcal{F}} R(f)$. Then using Inequality (3.8) with $\boldsymbol{f}' = \boldsymbol{f}^*$ we have

$$\frac{1}{\alpha}(\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \langle \boldsymbol{f}^*, \sum_{t=1}^{T}\boldsymbol{x}_t\rangle) + \frac{\alpha-1}{\alpha}(\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle) \le \frac{1}{\eta\alpha}\boldsymbol{R} + \frac{\eta\alpha}{\beta}\sum_{t=1}^{T}(||\boldsymbol{x}_{t-1} - \boldsymbol{x}_t||_q)^2$$

Now, against a general adversary, if $\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \langle \boldsymbol{f}^*, \sum_{t=1}^{T}\boldsymbol{x}_t\rangle \le 0$ then by definition, AFTRL has $O(1)$ external regret. In case where $\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \langle \boldsymbol{f}^*, \sum_{t=1}^{T}\boldsymbol{x}_t\rangle \ge 0$, using Inequality (3.8) and setting $\eta\alpha = \sqrt{\beta\boldsymbol{R}/(\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2)}$ we have

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \le \frac{\alpha}{\alpha-1}\sqrt{\boldsymbol{R}\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2/\beta} = O(\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2).$$

For unknown bound $\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2$, we can use the Doubling Trick as shown in Appendix 3.26 to achieve a similar regret bound.

Against a no-external regret adversary, using Lemma 3.13, we then have:

$$\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2 = \sum_{t=1}^{T}(O(\frac{1}{\sqrt{T}}))^2 = O(1).$$

Thus, Inequality (3.8) becomes:

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \frac{1}{\alpha}\langle \boldsymbol{f}', \sum_{t=1}^{T}\boldsymbol{x}_t\rangle - \frac{\alpha-1}{\alpha}\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \le \frac{1}{\eta\alpha}\boldsymbol{R} + \frac{\eta\alpha}{\beta}O(1) = O(1).$$

Following a similar reasoning for general adversary, AFTRL achieves $O(1)$ external regret or $O(1)$ forward regret against no-external regret adversary. The full proof is given in Appendix 3.30. □

Remark 3.15 (**AFTRL vs OFTRL**). While both AFTRL and OFTRL share the same idea of exploiting "predictable sequences", they are significantly different. Firstly, the level of dependency on predictable sequences in OFTRL is fixed to 1, whereas AFTRL allows a flexible control over the predictable sequences (i.e., via parameter α). Thus, AFTRL can achieve much better performance in situation of accurate prediction compared to OFTRL, which can be reassured by experiment results in Figure 3.3. Secondly, in the worst case scenario, AFTRL can guarantee a stronger forward regret bound compared to external regret bound of OFTRL in Rakhlin and Sridharan (2013a).

Our techniques can be extended to a different class of algorithm such as Mirror Descent (Nemirovskij and Yudin, 1983). We introduce Accurate Mirror Descent (AMD) with a similar regret bound as AFTRL. Let $\mathcal{R}$ be a β-strongly convex function with respect to a norm $\|.\|_p$, and let $D_{\mathcal{R}(.,.)}$ denote the Bregman divergence with respect to $\mathcal{R}$. Let $\|.\|_q$ be dual to $\|.\|_p$. Then the AMD algorithm can be described as follows.

Algorithm 16 Accurate Mirror Descent

Input: learning rate $\eta > 0$, exploiting rate $\alpha \geq 1$, $\boldsymbol{f}_1 = \boldsymbol{g}_1 = \operatorname{argmin}_{\boldsymbol{f}\in\mathcal{F}} \mathcal{R}(\boldsymbol{f})$
Output: next strategy update

$$\boldsymbol{g}_{t+1} = \operatorname*{argmin}_{\boldsymbol{g}\in\mathcal{F}} G_{t+1}(\boldsymbol{g}) = \eta\langle \boldsymbol{g}, \boldsymbol{x}_t\rangle + D_{\mathcal{R}}(\boldsymbol{g}, \boldsymbol{g}_t)$$
$$\boldsymbol{f}_{t+1} = \operatorname*{argmin}_{\boldsymbol{f}\in\mathcal{F}} F_{t+1}(\boldsymbol{f}) = \eta\langle \boldsymbol{f}, \alpha M_{t+1}\rangle + D_{\mathcal{R}}(\boldsymbol{f}, \boldsymbol{g}_{t+1})$$

The regularizer $R(\boldsymbol{f})$ is a β-strongly convex function with respect of l_p norm, $p \geq 1$. The following theorem provides the regret bound for AMD:

Theorem 3.16. *Let $\mathcal{F}$ be a convex set in a Banach space $\mathcal{B}$. Let $\mathcal{R} : \mathcal{B} \to \mathbb{R}$ be a β-strongly convex function on $\mathcal{F}$ with respect to some norm $\|.\|_p$. Denote $\|.\|_q$ the dual norm with $1/p + 1/q = 1$. For any strategy of the environment and any $\boldsymbol{f}' \in \mathcal{F}$, AMD yields*

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \frac{1}{\alpha}\langle \boldsymbol{f}', \boldsymbol{x}_t\rangle - \frac{\alpha-1}{\alpha}\langle \boldsymbol{g}_{t+1}, \boldsymbol{x}_t\rangle \leq \frac{\eta\alpha}{2\beta}\sum_{t=1}^{T}\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q^2 + \frac{\mathcal{R}_{max}^2}{\eta\alpha},$$

where $\mathcal{R}_{max}^2 = \max_{\boldsymbol{f}\in\mathcal{F}} \mathcal{R}(\boldsymbol{f}) - \min_{\boldsymbol{f}\in\mathcal{F}} \mathcal{R}(\boldsymbol{f})$. The full proof is given in Appendix 3.31.

3.6 Prod with Best Response

While AFTRL gives us a guarantee of no-forward regret, one can wonder whether the agent can achieve a better performance (e.g., no-dynamic regret) given the extra knowledge. In this section, we introduce Prod with Best Response algorithm (Prod-BR) such that the agent can achieve no-dynamic regret against the no-external regret adversary while maintaining a no-external regret performance in the worst case. Our variant Prod-BR algorithm gets motivation from (A,B)-Prod algorithm in Sani et al. (2014), in which we observe that the best response strategy from current feedback can exploit a no-external regret adversary. The Prod-BR runs two separate algorithms (i.e., FTRL and BR) inside the main algorithm. Intuitively, while FTRL maintains a performance guarantee against the worst case scenario, BR algorithm exploits the extra structure against no-external regret adversary and thus make Prod-BR algorithm efficient. Prod-BR can balance between accurate and inaccurate extra knowledge so that the agent can achieve $O(\sqrt{T})$ dynamic regret against no-external regret adversary while maintaining $O(\sqrt{T}\log(T))$ external regret in the worst case scenario.

Algorithm 17 Prod-Best Response algorithm

Input: learning rate $\eta > 0$, $\eta_1 \in (0, 1]$, initial weight $w_{1,R}, w_{1,BR}$, regularizer function $R(.)$
$\boldsymbol{f}_{t+1} = \text{argmin}_{\boldsymbol{f}\in\mathcal{F}} F_{t+1}(\boldsymbol{f}) = \langle \boldsymbol{f}, \sum_{s=1}^{t} \boldsymbol{x}_s\rangle + \frac{R(\boldsymbol{f})}{\eta}$; $BR_{t+1} = \text{argmin}_{\boldsymbol{f}\in\mathcal{F}}\langle f, \boldsymbol{x}_t\rangle$
Output: next strategy update $\boldsymbol{g}_{t+1}$ and next weight $w_{t+1,R}$:

$$\boldsymbol{g}_{t+1} = \frac{w_{t,R}}{w_{t,R} + w_{1,BR}}\boldsymbol{f}_{t+1} + \frac{w_{1,BR}}{w_{t,FTRL} + w_{1,BR}} BR_{t+1}; \quad w_{t+1,R} = w_{t,R}(1+\eta_1\langle BR_{t+1} - \boldsymbol{f}_{t+1}, \boldsymbol{x}_{t+1}\rangle)$$

We first show that in the case where the adversary follows a no-external regret algorithm (i.e., FTRL, OMD) with optimal learning rate, then the best response with respect to the previous feedback can guarantee the agent the following:

Lemma 3.17. *Let $\boldsymbol{x}_t$, $\boldsymbol{x}_{t+1}$ be two consecutive strategies of a no-external regret algorithm (i.e., FTRL, OMD). Then, we have*

$$\langle \boldsymbol{b}, \boldsymbol{x}_{t+1}\rangle - \langle \boldsymbol{c}, \boldsymbol{x}_{t+1}\rangle = O(\frac{1}{\sqrt{T}}), \text{ where } \boldsymbol{b} = \underset{\boldsymbol{f}\in\mathcal{F}}{\text{argmin}}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle \ , \ \boldsymbol{c} = \underset{\boldsymbol{f}\in\mathcal{F}}{\text{argmin}}\langle \boldsymbol{f}, \boldsymbol{x}_{t+1}\rangle.$$

The full proof is given in Appendix 3.32.

We then can prove the following theorem about the performance of Prod-BR algorithm:

Theorem 3.18. *Let the agent follows Prod-BR Algorithm 17 with $\eta = n/\sqrt{2T}$, $\eta_1 = 1/2.\sqrt{\log(T)/T}$ and $w_{1,BR} = 1 - w_{1,R} = 1 - \eta_1$. Then it achieves $O(\sqrt{T\log(T)})$ external regret against a general adversary while maintaining $O(\sqrt{T})$ dynamic regret against the strategic adversary.*

Proof. Following the regret bound analysis of (A,B)-Prod in Corollary 1 in (Sani et al., 2014) we have

$$\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \leq \sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle + 2\sqrt{T\log(T)} \text{ and} \tag{3.12a}$$

$$\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \leq \sum_{t=1}^{T}\langle BR_t, \boldsymbol{x}_t\rangle + 2\log(2). \tag{3.12b}$$

Since the agent uses the optimal learning rate for FTRL inside Algorithm 17, following the regret bound analysis of FTRL (i.e., see Shalev-Shwartz (2012)) we have

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle \leq n\sqrt{2T} \;\; \forall \boldsymbol{f} \in \mathcal{F}.$$

Along with Inequality (3.12a) we have

$$\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle \leq 2\sqrt{T\log(T)} + n\sqrt{2T} = O(\sqrt{T\log(T)}) \;\; \forall \boldsymbol{f} \in \mathcal{F},$$

or Prod-BR achieves $O(\sqrt{T\log(T)})$ external regret against general adversary. For the second part of the proof, using Inequality (3.12b) along with Lemma 3.17 we have

$$\begin{aligned}\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \operatorname*{argmin}_{\boldsymbol{f}\in\mathcal{F}}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle &\leq \sum_{t=1}^{T}\langle BR_t, \boldsymbol{x}_t\rangle - \operatorname*{argmin}_{\boldsymbol{f}\in\mathcal{F}}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle + 2\log(2)\\ = \sum_{t=1}^{T} O(\frac{1}{\sqrt{T}}) + 2\log(2) &= O(\sqrt{T}),\end{aligned}$$

or Prod-BR has $O(\sqrt{T})$ dynamic regret against no-external regret adversary. □

Remark 3.19 (Prod-BR vs AFTRL). In the worst case scenario, AFTRL provides a better performance guarantee over Prod-BR ($O(\sqrt{T})$ vs $O(\sqrt{T\log(T)})$). However, against the strategic adversary, Prod-BR provides a much stronger notion of performance guarantee (no-dynamic regret) compared to no-forward regret of AFTRL. Note that both Prod-BR and AFTRL rely on the small distance between two consecutive strategies of the adversary. While it holds true for many no-external regret algorithms as in Lemma 3.13, there are no-external regret algorithms (i.e., AdaHedge (De Rooij et al., 2014)) such as the distance between two consecutive strategies will have the form: $\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_q = O(1/\sqrt{t})$ where t denotes the current iteration. In this situation, following the same argument, AFTRL achieves $O(1)$ external regret or $O(\log(T))$ forward regret while Prod-BR maintains $O(\sqrt{T})$ dynamic regret.

3.7 Accurate Multiplicative Weights Update with Last Round Convergence

Algorithm 18 Accurate Multiplicative Weights Update

Input: learning rate $\eta > 0$, exploiting rate $\alpha > 0$, $\boldsymbol{f}_1 = \boldsymbol{f}_2 = [1/n, \ldots, 1/n]$

Output: Next update

$$\boldsymbol{f}_{t+1}(i) = \frac{\boldsymbol{f}_t(i) e^{\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_t - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_{t-1})}}{\sum_j \boldsymbol{f}_t(j) e^{\eta((\alpha+1)\boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}_t - \alpha \boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}_{t-1})}}, \tag{3.13}$$

$\boldsymbol{e}_i$ denotes the unit-vector with weight of 1 at i-component

We now turn to the second group of our contributions in this chapter, namely: to ensure last round convergence with this new algorithmic framework. We show that if both players follow Accurate Multiplicative Weights Update (AMWU), a special case of AFTRL with entropy regularizer, then the dynamic converges last round to the NE in zero-sum game with unique NE. [7]

Note here that the uniqueness assumption of NE is generic in the following sense: since the set of zero-sum games with non-unique equilibrium has Lebesgue measure zero (Van Damme, 1991), if the entries of $\boldsymbol{A}$ are independently sampled from some continuous distribution, then with probability one, the game has a unique NE. We leave the relaxation of the uniqueness assumption for future work.

Our main last round convergence result is as follows:

Theorem 3.20 (**Last Round Convergence of AMWU**). *Let* $(\boldsymbol{f}^*, \boldsymbol{y}^*)$ *be a unique Nash Equilibrium of the matrix game* $\boldsymbol{A}$. *Then, with* $\alpha = \eta^{b-1}$ *for* $b \in (0, 1]$ *and sufficiently small* η, *the dynamic of AMWU converges last round to the NE of the game:* $\lim_{t\to\infty}(\boldsymbol{f}_t, \boldsymbol{y}_t) = (\boldsymbol{f}^*, \boldsymbol{y}^*)$.

Proof of Sketch. We break the proof into three main parts. First, we prove that the K-L divergence (Kullback and Leibler, 1951) between the t-th strategy $(\boldsymbol{f}_t, \boldsymbol{y}_t)$ and $(\boldsymbol{f}^*, \boldsymbol{y}^*)$ will decrease by a factor of η^{2+b} unless the strategy $(\boldsymbol{f}_t, \boldsymbol{y}_t)$ is $O(\eta^{b/3})$-close [8]:

$$RE((\boldsymbol{f}^*, \boldsymbol{y}^*)||(\boldsymbol{f}_{t+1}, \boldsymbol{y}_{t+1})) \leq RE((\boldsymbol{f}^*, \boldsymbol{y}^*)||(\boldsymbol{f}_t, \boldsymbol{y}_t)) - \Omega(\eta^{b+2}).$$

[7] With some abuse of notation, in this section we use both $\boldsymbol{f}(i)$ and $\boldsymbol{f}_i$ to denote the i-th element of vector $\boldsymbol{f}$.

[8] We later define it rigorously in Definition 3.33.

The key step is the observation that the quantity $\boldsymbol{f}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t - \boldsymbol{f}_t^\top \boldsymbol{A}\boldsymbol{y}_{t-1}$ can be bounded by:

$$\eta \boldsymbol{f}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t - \eta \boldsymbol{f}_t^\top \boldsymbol{A}\boldsymbol{y}_{t-1} = -\eta^2 \sum_i \boldsymbol{f}_t(i)((\boldsymbol{f}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha \boldsymbol{y}_{t-1}))^2$$
$$- \eta^2 \sum_i \boldsymbol{y}_t(i)((\boldsymbol{y}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}^\top((\alpha \boldsymbol{y}_{t-1} - (\alpha+1)\boldsymbol{y}_t))^2 + O(\eta^{2+b}).$$

From the above result, we then have that if the starting point is uniform (i.e., $\boldsymbol{f}_1 = (1/n, \ldots, 1/n)$ and $\boldsymbol{y}_1 = (1/m, \ldots, 1/m)$), AMWU will reach $O(\eta^{b/3})$-close in at most: $O\left(\frac{\log(nm)}{\eta^{2+b}}\right)$ time steps.

Secondly, we show that $\eta^{b/3}$-close point implies close to the NE with sufficiently small η. The proof comes closely related to the proof of Theorem 3.2 in Daskalakis and Panageas (2019). Thus, for any starting strategy with non-zero element and a sufficient small learning rate η, AMWU can get arbitrarily close to the NE.

Finally, by proving that the spectral radius of the unique minimax equilibrium is less than one, we show that the update dynamic of AMWU is a locally converging on the NE point, meaning that there is last round convergence to the NE if the dynamic leads to a point in the neighborhood of the NE. Now, applying the first and second points to the dynamic of AMWU algorithm with non-zero element starting strategy, we have that AMWU will get arbitrarily close to the NE $(\boldsymbol{f}^*, \boldsymbol{y}^*)$ with a sufficiently small learning rate η. Then, using the locally converging property of AMWU, the last round convergence result in Theorem 3.20 will follow directly.

All the missing proofs can be found in Appendix 3.10.2. □

3.8 Experiment

In this section, we test the performance of our algorithms OSO, AMWU and Prod-BR in several settings. Firstly, we verify the linear dependency between the size of the effective strategy set and the NE support size in random matrix games. Secondly, we test the performance of OSO and relevant algorithms against the strategic adversary in meta games (Czarnecki et al., 2020). Thirdly, we consider an oblivious no-external regret adversary and measure the average loss performance of our algorithms against baselines (i.e., MWU, OMWU) in both random and meta games. Fourthly, we test AMWU and Prod-BR against a non-oblivious no-external regret adversary and measure the average dynamic regret performance. Finally, we test AMWU in a self-play setting and measure the last round convergence rate to the NE.

Size of k vs. Support Size of NE: We consider a set of zero-sum normal-form games of different sizes, the entries of which are sampled from a uniform distribution $\mathbf{U}(0, 1)$.

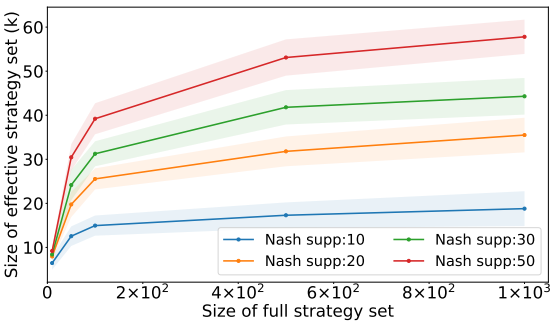


FIGURE 3.1: Sizes of effective strategy set (i.e., k) in cases of an OSO agent playing against an MWU opponent with different sizes of full strategy set and NE support.

We run OSO as the row player against a no-regret column player [9] until convergence, and plot the size of the OSO player's effective strategy set against its full strategy size. We run 20 seeds for each setting. As we can see from Figure 3.1, given a fixed support size of the NE, which is achieved by fixing the number of columns while increasing the number of rows in the game matrix, the size of the effective strategy set k grows as the size of the full strategy set increases, but plateaus quickly. The larger the size of the NE support (not the full strategy set!), the higher this plateau will reach. Clearly, we can tell that the size of OSO's effective strategy set **does not** increase drastically with the full strategy size, but rather depends on the support size of the NE. This result confirms Theorem 3.6 in which we prove that OSO's regret bound depends on k, which is related to the size of the NE support but not the game size. Economically, this is a desired property as OSO can potentially avoid unnecessary computation, in contrast to other no-regret methods that require looping over the full strategy set at each iteration.

OSO against MWU: we also look at the setting of playing against an MWU adversary in Figure 3.2. We can see that OSO outperforms MWU and DO baselines in average performance in almost all 15 games, which confirms the effectiveness of our design compared to the relevant algorithms. Notably, MWU achieves a constant payoff; we believe this is because these games are symmetric and since both players follow MWU with the same learning rate, the payoff will always be the value of the game (thus the ground truth), which OSO will eventually converge to as well.

Performance against oblivious adversary: for a fair average loss performance comparison between AMWU, Prod-BR and the baselines, we consider oblivious MWU adversaries: the agent's historical strategies do not affect the strategy of the MWU adversary. In order to create this non-oblivious adversary, we assume the adversary follows MWU to play against a different opponent rather than the agent and therefore the agent's strategies do not affect the adversary's behaviour. [10]. As we can see in Figure 3.3, AMWU and Prod-BR outperform other baselines by a large margin. In particular, Prod-BR achieves

[9] We choose MWU for column player in our experiment, but we expect other no-regret algorithms would give a similar result.

[10] The detail setting can be found in Appendix 3.11.1

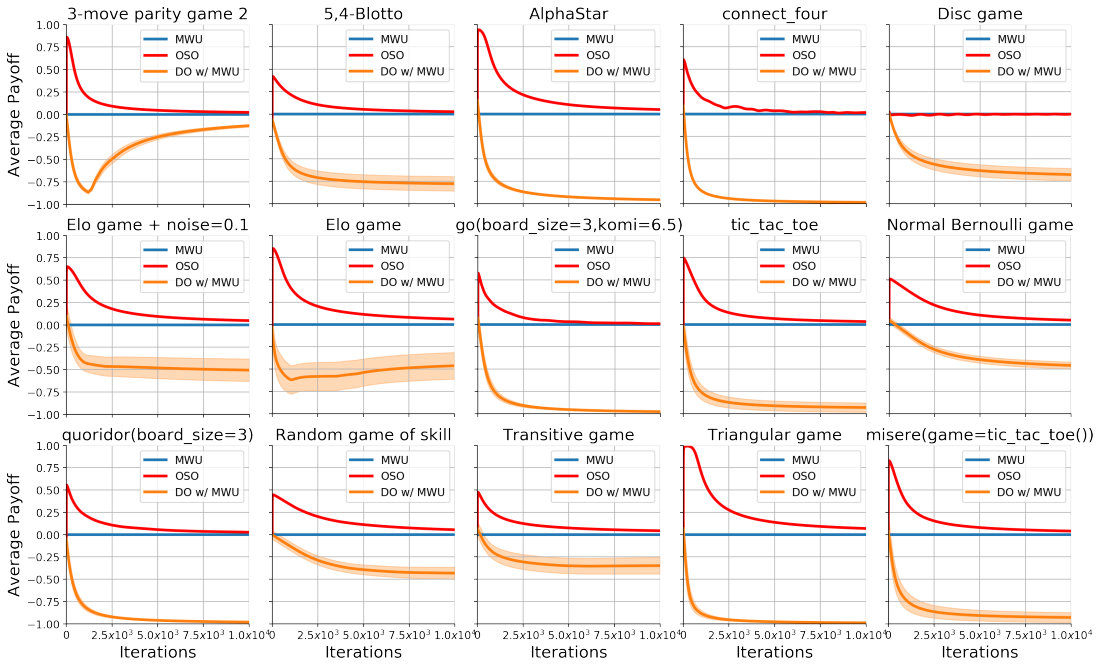


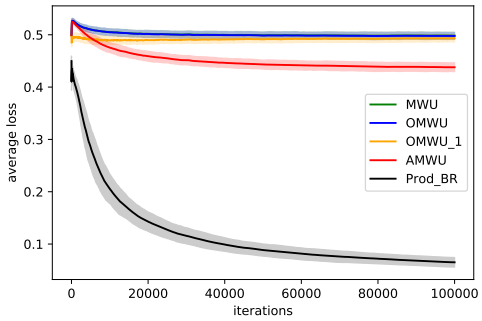
FIGURE 3.2: Performance comparisons against MWU adversary

the smallest average loss compared to AMWU and other baselines. Intuitively, since the agent plays against an oblivious adversary, a better theoretical regret guarantee of AMWU and Prod-BR can imply a better average loss performance as we have shown in this experiment. Therefore, Prod-BR with the best regret bound measure (i.e., dynamic regret) achieves the best performance, followed by AMWU with a forward regret guarantee. An interesting observation is that the performance of MWU is almost identical to OMWU with the same learning rate in our setting, reassuring the point in which OMWU does not exploit enough extra knowledge.

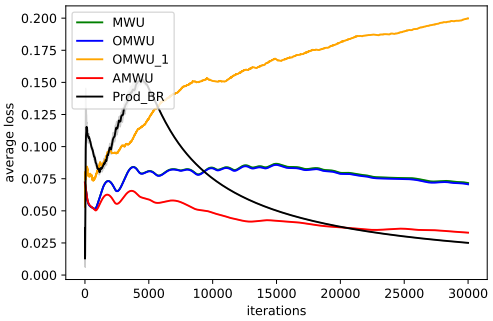
Performance against non-oblivious adversary: we now test our algorithms against non-oblivious adversaries (i.e., the agent's behaviour can change the adversary's strategy) and answer the question: can better theoretical regret bound of AMWU and Prod-BR lead to better regret performance against no-external regret adversary in practice? As we can see in Figure 3.4, AMWU and Prod-BR achieve much smaller average dynamic regret compared to the baselines. This further assures our theoretical results as both AMWU and Prod-BR have better regret bound guarantee against no-external regret adversary compared to the baselines, leading to better regret bound in practice.

Last round convergence: we compare the rate of convergence of AMWU against OWMU and MWU. For a fair comparison, we use a common learning rate $\mu = 0.01$ for all 3 algorithms [11]. As we can see in Figure 3.5, AMWU outperforms OMWU and MWU by a large margin in convergence to the NE. Interestingly, in Connect Four and

[11] The results for other values have similar broad view. See Appendix 3.11.2 for more details.

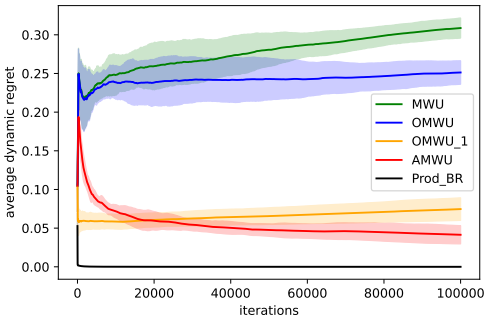


(A) 0.5 learning rate MWU adversary in random game

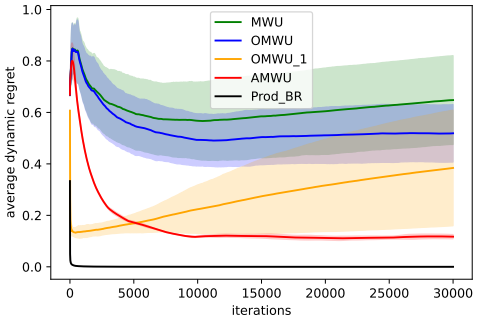


(B) 0.5 learning rate MWU adversary in Connect Four

FIGURE 3.3: Average Loss Against Oblivious MWU adversary



(A) non-oblivious MWU adversary in random game



(B) non-oblivious MWU adversary in Connect Four

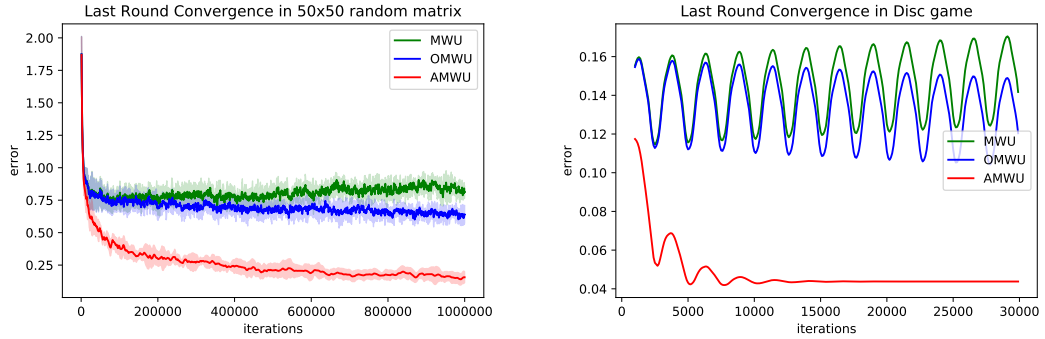
FIGURE 3.4: Average Loss Against Non-Oblivious MWU adversary

Disc meta games, AMWU shows clear convergence pattern whereas OMWU and MWU fluctuate under the same setting (Figure 3.5b).

AMWU vs OMWU: in order to highlight the difference between AMWU and OMWU, we test OWMU$_1$ with the same relative weight between the predictable sequence $\boldsymbol{x}_{t-1}$ and the regularizer $R(\boldsymbol{f})$ as AMWU (i.e., $\eta_{OMWU} = \eta_{AMWU} \times \alpha_{AMWU}$). As we can clearly see in Figure 3.3, AMWU outperform OWMU$_1$ in every game that we consider. We can confirm that AMWU and OMWU are two very different algorithm due to its level of exploiting extra knowledge.

3.9 Conclusion

We study online learning problems in which the learner has extra knowledge about the adversary's behaviour (i.e., strategic adversary). Under this setting, OSO can achieve an external-regret bound of $\mathcal{O}(\sqrt{k\log(k)T})$, where k is the size of the effective strategy set. Furthermore, our algorithms AFTRL and Prod-BR can intensively exploit this extra knowledge to achieve $O(1)$ forward regret and $O(\sqrt{T})$ dynamic regret, respectively. In addition, both AFTRL and Prod-BR retain no-regret properties in the worst case



(A) 50×50 random games (B) Disc meta game

FIGURE 3.5: Last Round Convergence

scenario of inaccurate extra knowledge. Finally, we show that AMWU, a special case of AFTRL, leads to last round convergence in two-player zero-sum games with a unique NE.

3.10 Appendix A: Detail Proofs

Lemma 3.21 (Proof of Lemma 3.11)**.** *Suppose the players start with the entry $\boldsymbol{A}_{1,1}$ and the game matrix $\boldsymbol{A}$ of the two-players zero-sum game is designed such as*

$$\begin{aligned}
&\boldsymbol{A}_{i,i} = 0.5 + \frac{0.1i}{n}\ \forall i \in [n];\quad \boldsymbol{A}_{i,i+1} = 0.9\ \forall i \in [n-1],\\
&\boldsymbol{A}_{i,j} = 0.8\ \forall j \geq i+2, i \in [n],\\
&\boldsymbol{A}_{i,j} = \boldsymbol{A}_{i,i} + \frac{0.1}{2n}\ \forall j \leq i, i \in [n],
\end{aligned}$$

where n is the size of the pure strategy set for both players. Then the game has a unique Nash equilibrium with support size of 1 (i.e., the entry $\boldsymbol{A}_{n,n}$) and the effective strategy set both DO and OSO method will reach the size of pure strategy set: $k = n$.

Proof. First, we show that the game $\boldsymbol{A}$ has a unique Nash equilibrium at entry $\boldsymbol{A}_{n,n}$. Following the design of matrix $\boldsymbol{A}$ we have

$$\boldsymbol{a}^n = \operatorname*{argmin}_{\boldsymbol{a}\in\Pi} \boldsymbol{a}^\top \boldsymbol{A}\boldsymbol{c}^n;\quad \boldsymbol{c}^n = \operatorname*{argmax}_{\boldsymbol{c}\in C} \boldsymbol{a}^{n\top} \boldsymbol{A}\boldsymbol{c}$$

Thus by definition, $(\boldsymbol{a}^n, \boldsymbol{c}^n)$ is the NE of the game and $\boldsymbol{A}_{n,n}$ is the minimax value v of the game. Suppose there is another NE of the game $(\hat{\boldsymbol{a}}, \hat{\boldsymbol{c}})$ such that $\hat{\boldsymbol{a}} \neq \boldsymbol{a}^n$ [12]. Then, by definition of the NE we have

$$\hat{\boldsymbol{a}}^\top \boldsymbol{A}\boldsymbol{c}^n \leq \hat{\boldsymbol{a}}^\top \boldsymbol{A}\hat{\boldsymbol{c}} = v = \boldsymbol{A}_{n,n},$$

since the minimax value v is unique in two-player zero-sum game. However, by the design of matrix $\boldsymbol{A}$, $\boldsymbol{a}^\top \boldsymbol{A}\boldsymbol{c}^n \geq v\ \forall \boldsymbol{a} \in \Pi$ and the equal sign holds true only if $\boldsymbol{a} = \boldsymbol{a}^n$. This leads to a contradiction. Thus, the game $\boldsymbol{A}$ has a unique pure NE with respect to the entry $\boldsymbol{A}_{n,n}$.

Next, we show that Double Oracle method with the NE as best response target will recover the whole pure strategy set in this game.

By definition, the game start with the entry $\boldsymbol{A}_{1,1}$ and the initial effective strategy sets are $\Pi_0 = \{\boldsymbol{a}^1\}$ and $C_0 = \{\boldsymbol{c}^1\}$. Since, $\boldsymbol{A}_{1,1} = \operatorname{argmin}_{i\in[n]} \boldsymbol{A}_{i,1}$ and $\boldsymbol{A}_{1,2} = \operatorname{argmax}_{j\in[n]} \boldsymbol{A}_{1,j}$, the new sub-game 2 is created with the corresponding effective strategy set: $\Pi_1 = \{\boldsymbol{a}^1\}$ and $C_1 = \{\boldsymbol{c}^1, \boldsymbol{c}^2\}$. Note that the effective strategy set of the row player remains unchanged in this iteration. The NE in the sub-game 2 is the with respect to entry $\boldsymbol{A}_{1,2}$, thus in the next iteration, the best response targets for the column and row player are with respect to $\boldsymbol{a}^1$ and $\boldsymbol{c}^2$, respectively. Now, in iteration 3, since $\boldsymbol{A}_{2,2} = \operatorname{argmin}_{i\in[n]} \boldsymbol{A}_{i,2}$ and $\boldsymbol{A}_{1,2} = \operatorname{argmax}_{j\in[n]} \boldsymbol{A}_{1,j}$, the new sub-game 3 is created with the corresponding

[12] The same argument holds true in the case $\boldsymbol{c}^n \neq \hat{\boldsymbol{c}}$.

effective strategy set: $\Pi_2 = \{\boldsymbol{a}^1, \boldsymbol{a}^2\}$ and $C_1 = \{\boldsymbol{c}^1, \boldsymbol{c}^2\}$. Note that in this round, the effective strategy set of the column player remains unchanged. Following the same process, the effective strategy set of the row player will add the $\boldsymbol{a}^i$ pure strategy in iteration $2i-1$ while the effective strategy set of the column player will add the pure strategy $\boldsymbol{c}^i$ at iteration $2i-2$. Therefore, the DO method will add the whole pure strategy set until converging to the NE in this example.

For the OSO method, we can follow the same process in the above DO case. That is, for the adversary, we allow it to play the NE of the sub-game in the same order as in DO. Since OSO is a no-regret algorithm and the average loss will remain the same for each time window (since we fix the adversary in this case), the OSO algorithm will converge to the best response with respect to the current average loss. Since we design the game such that in each sub-game, the Nash Equilibrium will be a pure strategy, the best response with respect to the NE of the adversary will also be the NE of the player in the current sub-game. After the player (i.e., following OSO method) converges to the NE of the sub-game, the adversary will move to play the NE of the next sub-game. That way, the OSO algorithm will need to add the whole pure strategy set when playing against this type of adversary.

In a more specific way, the adversary can play the following policy. In the first iteration, the adversary plays $\boldsymbol{c}^1$. Since the best reponse with respect to $\boldsymbol{c}^1$ is $\boldsymbol{a}^1$, $\boldsymbol{a}^1$ will be added to the effective strategy set. Then at iteration 2, the adversary plays $\boldsymbol{c}^2$. Then, $\boldsymbol{a}^2$ is the best response with respect to the current average loss (i.e., $\boldsymbol{A}\boldsymbol{c}^1 + \boldsymbol{A}\boldsymbol{c}^2$). Thus, $\boldsymbol{a}^2$ is added to the effective strategy set. In the next iteration, by the design of the game matrix $\boldsymbol{A}$, we have the following relationship:

$$\boldsymbol{a}^{i+h} = \operatorname*{argmin}_{\boldsymbol{a}\in\Pi} \boldsymbol{a}^\top \boldsymbol{A}(\sum_{j=i}^{i+h} \boldsymbol{c}^j) \ \forall i, h > 0.$$

Thus, by letting the adversary plays $\boldsymbol{c}^1$ to $\boldsymbol{c}^n$ sequentially, the effective strategy set of the agent will need to recover the whole pure strategy set. Note that the policy used by the adversary is also a no-regret algorithm since in the later rounds, the adversary can just follows the pure strategy $\boldsymbol{c}^n$ and achieving the value v of the game, thus deducting any negative payoffs from the first n rounds. □

Theorem 3.22 (Regret Bound of OSO with Less-Frequent Best Response)**.** *Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T$ be a sequence of loss vectors played by an adversary. Then, OSO in Algorithm 14 is a no-regret algorithm with:*

$$\frac{1}{T}\Big(\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{f}\in\Pi}\sum_{t=1}^{T}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle\Big) \leq \frac{\sqrt{k\log(k)}}{\sqrt{2T}} + \frac{\sum_{i=1}^{k}\alpha^i_{|T_i|}}{T},$$

where $k = |\Pi_T|$ is the size of effective strategy set in the final time window.

Proof. W.l.o.g, we assume the player uses the MWU as the no-regret algorithm and starts with only one pure strategy in Π_0 in Algorithm 13. Since in the final time window, the effective strategy set has k elements, there are exactly k time windows. Denote $|T_1|, |T_2|, \ldots, |T_k|$ be the lengths of time windows during each of which the subset of strategies the no-regret algorithm considers does not change. In the case of finite set of strategies, k will be finite and we have

$$\sum_{i=1}^{k} |T_k| = T.$$

In the time window with length $|T_i|$, following the regret bound of MWU we have

$$\sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{f} \in \Pi_{|\bar{T}_i|+1}} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle \leq \sqrt{\frac{|T_i|}{2} \log(i)}, \quad \text{where } |\bar{T}_i| = \sum_{j=1}^{i-1} |T_j|. \tag{3.14}$$

Since in the time window T_i, the size of the effective strategy set does not change, thus we have

$$\min_{\boldsymbol{f} \in \Pi_{|\bar{T}_i|+1}} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{f} \in \Pi} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle \leq \alpha^i_{|T_i|}. \tag{3.15}$$

From Inequalities (3.14) and (3.15) we have

$$\sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{f} \in \Pi} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle \leq \sqrt{\frac{|T_i|}{2} \log(i)} + \alpha^i_{|T_i|}. \tag{3.16}$$

Sum up the inequality (3.16) for $i = 1, \ldots k$ we have

$$\begin{aligned}
&\sum_{i=1}^{k} \left(\sqrt{\frac{|T_i|}{2} \log(i)} + \alpha^i_{|T_i|} \right) \geq \sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \sum_{i=1}^{k} \min_{\boldsymbol{f} \in \Pi} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle \\
&\geq \sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{f} \in \Pi} \sum_{i=1}^{k} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle = \sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{f} \in \Pi} \sum_{i=1}^{T} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle \qquad \text{(3.17a)} \\
&\implies \sqrt{\frac{Tk \log(k)}{2}} + \sum_{i=1}^{k} \alpha^i_{|T_i|} \geq \sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \min_{\boldsymbol{f} \in \Pi} \sum_{i=1}^{T} \langle \boldsymbol{f}, \boldsymbol{x}_t \rangle. \qquad \text{(3.17b)}
\end{aligned}$$

Inequality (3.17a) is due to $\sum \min \leq \min \sum$. Inequality (3.17b) comes from Cauchy-Schwarz inequality and Stirling' approximation. Thus, we have the derived regret bound. □

3.10.1 Proof of Theorem 3.12

Before provide the proof for the Theorem, we need the following lemma:

Lemma 3.23. *DO will converge to ϵ-NE if players can only access to an ϵ-best response in each round.*

Proof. We first prove in the case of single oracle algorithm. The double oracle proof will be similar. Since the number of strategies is finite, by the same argument in the the case of exact best response, the process will converge. Suppose that at time step t, the process stops. Since we use ϵ-best response, we have the following relationship:

$$\boldsymbol{f}_t{}^\top \boldsymbol{A}_t \boldsymbol{y}_t - \min_{\boldsymbol{f}\in\Pi} \boldsymbol{f}^\top \boldsymbol{A}\boldsymbol{y}_t \leq \epsilon$$

If we set the weight of pure strategies does not appear in $\boldsymbol{f}_t$ to be zero to make a $\boldsymbol{f}'_t$, then it is obvious that

$$\boldsymbol{f}_t^\top \boldsymbol{A}_t = \boldsymbol{f}_t'^\top \boldsymbol{A}$$

Thus, we have the following relationship:

$$\boldsymbol{f}_t'^\top \boldsymbol{A}\boldsymbol{y}_t - \min_{\boldsymbol{f}\in\Pi} \boldsymbol{f}^\top \boldsymbol{A}\boldsymbol{y}_t \leq \epsilon \tag{3.18}$$

Further, since $\boldsymbol{y}_t$ is Nash equilibrium of $\boldsymbol{A}_t$, we also have

$$\max_{\boldsymbol{y}\in\Delta_y} \boldsymbol{f}_t'^\top \boldsymbol{A}\boldsymbol{y} - \boldsymbol{f}_t'^\top \boldsymbol{A}\boldsymbol{y}_t = \max_{l\in\Delta_l} \boldsymbol{f}_t{}^\top \boldsymbol{A}_t\boldsymbol{y} - \boldsymbol{f}_t{}^\top \boldsymbol{A}_t\boldsymbol{y}_t = 0 \tag{3.19}$$

From inequalities (3.18) and (3.19), by definition we conclude that $(\boldsymbol{f}'_t, \boldsymbol{y}_t)$ is ϵ-NE of the game $\boldsymbol{A}$. □

Now, we can prove Theorem 3.12:

Theorem 3.24. *Suppose OSO agent can only access the ϵ-best response in each iteration when following Algorithm 13, if the adversary follows a no-regret algorithm, then the average strategy of the agent will converge to an ϵ-NE. Furthermore, the algorithm is ϵ-regret:*

$$\lim_{T\to\infty} \frac{R_T}{T} \leq \epsilon; \quad R_T = \max_{\boldsymbol{f}\in\Delta_\Pi} \sum_{t=1}^{T} \left(\boldsymbol{f}_t^\top \boldsymbol{A}\boldsymbol{y}_t - \boldsymbol{f}^\top \boldsymbol{A}\boldsymbol{y}_t\right).$$

Proof. Suppose that the player uses the Multiplicative Weights Update in Algorithm 13 with ϵ-best response. Denote $|T_1|, |T_2|, \ldots, |T_k|$ be the lengths of time windows during each of which the subset of strategies the no-regret algorithm considers does not change. Furthermore,

$$\sum_{i=1}^{k} |T_k| = T.$$

In a time window T_i, the regret with respect to the best-fixed strategy in the effective strategy set is:

$$\sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{i+1}|}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{f}\in\Pi_{|\bar{T}_i|+1}} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{(i+1)}|}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle \le \sqrt{\frac{|T_i|}{2}\log(i)}, \tag{3.20}$$

where $|\bar{T}_i| = \sum_{j=1}^{i-1}|T_j|$. Since in the time window T_i, the ϵ-best response strategy stays in $\Pi_{\bar{T}_i+1}$ and therefore we have

$$\min_{\boldsymbol{f}\in\Pi_{|\bar{T}_i|+1}} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{(i+1)}|}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{f}\in\Pi} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{(i+1)}|}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle \le \epsilon|T_i|$$

Then, from the inequality (3.20) we have

$$\sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{(i+1)}|}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{f}\in\Pi} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{(i+1)}|}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle \le \sqrt{\frac{|T_i|}{2}\log(i)} + \epsilon|T_i|, \tag{3.21}$$

Sum up the inequality (3.21) for $i = 1, \dots k$ we have

$$\begin{aligned}
&\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \sum_{i=1}^{k}\min_{\boldsymbol{f}\in\Pi} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{(i+1)}|}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle \le \sum_{i=1}^{k}\sqrt{\frac{|T_i|}{2}\log(i)} + \epsilon|T_i|, \\
&\implies \sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{f}\in\Pi}\sum_{i=1}^{k} \sum_{t=|\bar{T}_i|+1}^{|\bar{T}_{(i+1)}|}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle \le \epsilon T + \sum_{i=1}^{k}\sqrt{\frac{|T_i|}{2}\log(i)} \qquad (3.22a) \\
&\implies \sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{f}\in\Pi}\sum_{t=1}^{T}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle \le \epsilon T + \sum_{i=1}^{k}\sqrt{\frac{|T_i|}{2}\log(i)} \\
&\implies \sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{f}\in\Pi}\sum_{t=1}^{T}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle \le \epsilon T + \sqrt{\frac{T}{2}}\sqrt{k\log(k)}. \qquad (3.22b)
\end{aligned}$$

Inequality (3.22a) is due to $\sum\min \le \min\sum$. Inequality (3.22b) comes from Cauchy-Schwarz inequality and Stirling' approximation. Using inequality (3.22b), we have

$$\min_{\boldsymbol{f}\in\Pi}\langle \boldsymbol{f}, \bar{\boldsymbol{x}}\rangle \ge \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \sqrt{\frac{k\log(k)}{2T}} - \epsilon. \tag{3.23}$$

That is, the OSO algorithm is ϵ-regret in the case of ϵ-best response. Since the adversary follows a no-regret algorithm, we have

$$\begin{aligned}
&\max_{\boldsymbol{x}}\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}\rangle - \sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle \le \sqrt{\frac{T}{2}}\sqrt{\log(L)} \\
&\implies \max_{\boldsymbol{x}}\langle \bar{\boldsymbol{f}}, \boldsymbol{x}\rangle \le \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle + \sqrt{\frac{\log(L)}{2T}}
\end{aligned} \tag{3.24}$$

Using the inequalities in (3.23) and (3.24) we have

$$\langle \bar{\boldsymbol{f}}, \bar{\boldsymbol{x}} \rangle \geq \min_{\boldsymbol{f} \in \Pi} \langle \boldsymbol{f}, \bar{\boldsymbol{x}} \rangle \geq \frac{1}{T} \sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle - \sqrt{\frac{k \log(k)}{2T}} - \epsilon$$
$$\geq \max_{\boldsymbol{x}} \langle \bar{\boldsymbol{f}}, \boldsymbol{x} \rangle - \sqrt{\frac{\log(L)}{2T}} - \sqrt{\frac{k \log(k)}{2T}} - \epsilon$$

Similarly, we also have

$$\langle \bar{\boldsymbol{f}}, \bar{\boldsymbol{x}} \rangle \leq \max_{\boldsymbol{x}} \langle \bar{\boldsymbol{f}}, \boldsymbol{x} \rangle \leq \frac{1}{T} \sum_{t=1}^{T} \langle \boldsymbol{f}_t, \boldsymbol{x}_t \rangle + \sqrt{\frac{\log(L)}{2T}}$$
$$\leq \min_{\boldsymbol{f} \in \Pi} \langle \boldsymbol{f}, \bar{\boldsymbol{x}} \rangle + \epsilon + \sqrt{\frac{k \log(k)}{2T}} + \sqrt{\frac{\log(L)}{2T}}$$

Take the limit $T \to \infty$, we then have

$$\max_{\boldsymbol{x}} \langle \bar{\boldsymbol{f}}, \boldsymbol{x} \rangle - \epsilon \leq \langle \bar{\boldsymbol{f}}, \bar{\boldsymbol{x}} \rangle \leq min_{\boldsymbol{f} \in \Pi} \langle \boldsymbol{f}, \bar{\boldsymbol{x}} \rangle + \epsilon$$

Thus $(\bar{\boldsymbol{f}}, \bar{\boldsymbol{x}})$ is the ϵ-Nash equilibrium of the game. □

Lemma 3.25 (Lemma 3.5)**.** *Let $\boldsymbol{g}_t$ be defined as above, then the following relationship holds for any $\boldsymbol{f} \in \mathcal{F}$:*

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \leq \langle \boldsymbol{f}, \sum_{t=1}^{T} \boldsymbol{x}_t \rangle + \frac{R(\boldsymbol{f})}{\eta}.$$

Proof of Lemma 3.5. We prove this by induction. For $t = 1$:

$$\langle \boldsymbol{g}_1, \boldsymbol{x}_1 \rangle \leq \langle \boldsymbol{g}_1, \boldsymbol{x}_1 \rangle + \frac{R(\boldsymbol{g}_1)}{\eta} \leq \langle \boldsymbol{f}, \boldsymbol{x}_1 \rangle + \frac{R(\boldsymbol{f})}{\eta} \; \forall \boldsymbol{f} \in \mathcal{F}.$$

Suppose the statement is true for T such that

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \leq \langle \boldsymbol{f}, \sum_{t=1}^{T} \boldsymbol{x}_t \rangle + \frac{R(\boldsymbol{f})}{\eta} \; \forall \boldsymbol{f} \in \mathcal{F}.$$

Adding $\langle \boldsymbol{g}_{T+1}, \boldsymbol{x}_{T+1} \rangle$ on both sides we have

$$\sum_{t=1}^{T+1} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \leq \langle \boldsymbol{f}, \sum_{t=1}^{T} \boldsymbol{x}_t \rangle + \frac{R(\boldsymbol{f})}{\eta} + \langle \boldsymbol{g}_{T+1}, \boldsymbol{x}_{T+1} \rangle \; \forall \boldsymbol{f} \in \mathcal{F}$$
$$\leq \langle \boldsymbol{g}_{T+1}, \sum_{t=1}^{T} \boldsymbol{x}_t \rangle + \frac{R(\boldsymbol{g}_{T+1})}{\eta} + \langle \boldsymbol{g}_{T+1}, \boldsymbol{x}_{T+1} \rangle$$
$$\leq \langle \boldsymbol{f}, \sum_{t=1}^{T+1} \boldsymbol{x}_t \rangle + \frac{R(\boldsymbol{f})}{\eta} \; \forall \boldsymbol{f} \in \mathcal{F}.$$

Thus the statement is true for $T + 1$.

From the above Inequality, if an algorithm is a no-forward regret, i.e.:

$$\sum_{t=1}^{\top}(\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle) = o(T),$$

then we also have:

$$\begin{aligned}\min_{\boldsymbol{f}\in\mathcal{F}}\sum_{t=1}^{\top}(\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \langle \boldsymbol{f}, \boldsymbol{x}_t\rangle) &\leq \sum_{t=1}^{\top}(\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle) + \frac{R(\boldsymbol{f})}{\eta} \\ &= o(T) + \frac{R(\boldsymbol{f})}{\eta} = o(T).\end{aligned}$$

Thus, the algorithm is also a no-external regret algorithm. □

Lemma 3.26 (Doubling Trick)**.** *The idea of doubling trick is to divide the time interval into different phases and restart the algorithm (i.e., AFTRL) in each phase. We will prove that by considering different phases in the process, the AFTRL will still maintain the regret bound of* $O\left(\sqrt{\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_t^*)^2}\right)$.

Using Lemma 3.5, the regret bound in Equation 3.8 can be derived as:

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \langle \boldsymbol{f}^*, \sum_{t=1}^{T}\boldsymbol{x}_t\rangle \leq \frac{\alpha}{\eta\alpha}R(\boldsymbol{f}^*) + \frac{\eta\alpha}{\beta}\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2 \;\forall \boldsymbol{f}^* \in \mathcal{F}.$$

Now, we break the time interval T *into different phases, in which phase* i *has a constant learning rate* $\eta_i = \eta_0 2^{-i}$*. Define the starting point of phase* $i+1$ *such as*

$$s_{i+1} = \min\{\tau : \frac{\eta_i\alpha}{\beta}\sum_{t=s_i}^{\tau}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_t^*)^2 > \frac{\alpha}{\eta_i\alpha}R(\boldsymbol{f}^*)\}.$$

and $s_1 = 1$*. Let* N *be the last phase of the game and let* $s_{N+1} = T+1$*. We then have*

$$\begin{aligned}\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \langle \boldsymbol{f}^*, \sum_{t=1}^{T}\boldsymbol{x}_t\rangle &\leq \sum_{i=1}^{N}\frac{\alpha}{\eta_i\alpha}R(\boldsymbol{f}^*) + \frac{\eta_i\alpha}{\beta}\sum_{t=s_i}^{s_{i+1}-1}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_t^*)^2 \\ &\leq 2\sum_{i=1}^{N}\frac{\alpha}{\eta_i\alpha}R(\boldsymbol{f}^*) \leq \frac{2^{N+2}}{\eta_0}R(\boldsymbol{f}^*),\end{aligned}$$

where the inequalities come from the definition of s_i*. Note that we have*

$$\begin{aligned}\frac{1}{\eta_0} = \frac{1}{\eta_{N-1}2^{N-1}} &\leq \frac{1}{2^{N-1}}\sqrt{\sum_{t=s_{N-1}}^{s_N}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2}\sqrt{\frac{\alpha}{\beta R(\boldsymbol{f}^*)}} \\ &\leq \frac{1}{2^{N-1}}\sqrt{\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2}\sqrt{\frac{\alpha}{\beta R(\boldsymbol{f}^*)}}.\end{aligned}$$

Thus we have

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \langle \boldsymbol{f}^*, \sum_{t=1}^{T}\boldsymbol{x}_t\rangle \leq \frac{2^{N+2}}{\eta_0}R(\boldsymbol{f}^*)$$
$$\leq 2^{N+2}\frac{1}{2^{N-1}}\sqrt{\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2}\sqrt{\frac{\alpha}{\beta R(\boldsymbol{f}^*)}}R(\boldsymbol{f}^*)$$
$$= 8\sqrt{\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2}\sqrt{\frac{\alpha R(\boldsymbol{f}^*)}{\beta}} = O\left(\sqrt{\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q}\right).$$

Thus, we derive the result.

Lemma 3.27 (Lemma 3.13)**.** *Let $\boldsymbol{f}_t$, $\boldsymbol{f}_{t+1}$ be two consecutive strategies of no-regret algorithms (i.e., FTRL, OMD). Then we have for any norm $\|.\|_q$:*

$$\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_q = O(\frac{1}{\sqrt{T}}).$$

In order to prove Lemma 3.13, we first need to have the following lemmas about the distance between two consecutive strategies of FTRL and OMD:

Lemma 3.28. *Let $\boldsymbol{f}_t$, $\boldsymbol{f}_{t+1}$ be two consecutive strategies of FTRL algorithm. Then we have*

$$\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_p \leq \eta\frac{2n}{\beta},\ \textit{where } \|.\|_p \textit{ denotes } l_p \textit{ norm.}$$

Proof. Following the property of β-strongly convex function we have

$$F_t(\boldsymbol{f}_{t+1}) - F_t(\boldsymbol{f}_t) \geq \frac{\beta}{2\eta}\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_p^2$$
$$\iff \langle \boldsymbol{f}_{t+1}, \sum_{s=1}^{t}\boldsymbol{x}_s\rangle + \frac{R(\boldsymbol{f}_{t+1})}{\eta} - \langle \boldsymbol{f}_{t+1}, \boldsymbol{x}_t\rangle - \langle \boldsymbol{f}_t, \sum_{s=1}^{t-1}\boldsymbol{x}_s\rangle - \frac{R(\boldsymbol{f}_t)}{\eta} \geq \frac{\beta}{2\eta}\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_p^2$$
$$\iff F_{t+1}(\boldsymbol{f}_{t+1}) - \langle \boldsymbol{f}_{t+1}, \boldsymbol{x}_t\rangle - \langle \boldsymbol{f}_t, \sum_{s=1}^{t-1}\boldsymbol{x}_s\rangle - \frac{R(\boldsymbol{f}_t)}{\eta} \geq \frac{\beta}{2\eta}\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_p^2.$$

By definition, we have $F_{t+1}(\boldsymbol{f}_{t+1}) \leq F_{t+1}(\boldsymbol{f}_t)$. Thus, substitute it in the above inequality we have

$$F_{t+1}(\boldsymbol{f}_t) - \langle \boldsymbol{f}_{t+1}, \boldsymbol{x}_t\rangle - \langle \boldsymbol{f}_t, \sum_{s=1}^{t-1}\boldsymbol{x}_s\rangle - \frac{R(\boldsymbol{f}_t)}{\eta} \geq \frac{\beta}{2\eta}\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_p^2$$
$$\iff \langle \boldsymbol{f}_t, \sum_{s=1}^{t}\boldsymbol{x}_s\rangle + \frac{R(\boldsymbol{f}_t)}{\eta} - \langle \boldsymbol{f}_{t+1}, \boldsymbol{x}_t\rangle - \langle \boldsymbol{f}_t, \sum_{s=1}^{t-1}\boldsymbol{x}_s\rangle - \frac{R(\boldsymbol{f}_t)}{\eta} \geq \frac{\beta}{2\eta}\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_p^2$$
$$\iff \langle \boldsymbol{f}_t - \boldsymbol{f}_{t+1}, \boldsymbol{x}_t\rangle \geq \frac{\beta}{2\eta}\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_p^2 \implies \|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_p\|\boldsymbol{x}_t\|_q \geq \frac{\beta}{2\eta}\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_p^2$$
$$\implies \frac{2\eta n}{\beta} \geq \|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_p,$$

since $\boldsymbol{x}_t \in [0,1]^n$ then $\|\boldsymbol{x}_t\|_q \leq n^{1/q} = n^{1-1/p} \leq n$. Thus, we derive the result.

□

A similar property can be found in other no-regret algorithm, such as Online Mirror Descent:

Lemma 3.29. *Let $\boldsymbol{g}_t$, $\boldsymbol{g}_{t+1}$ be two consecutive strategies of OMD algorithm. Then we have*

$$\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_p \leq \frac{\eta}{\beta}$$

Proof. Following the property of β-strongly convex function we have

$$\begin{aligned}
&G_{t+1}(\boldsymbol{g}_t) - G_{t+1}(\boldsymbol{g}_{t+1}) \geq \frac{\beta}{2}\|\boldsymbol{g}_{t+1} - \boldsymbol{g}_t\|_p^2 \\
&\iff \eta\langle \boldsymbol{g}_t - \boldsymbol{g}_{t+1}, \boldsymbol{x}_t\rangle + D_{\mathcal{R}}(\boldsymbol{g}_t, \boldsymbol{g}_t) - D_{\mathcal{R}}(\boldsymbol{g}_{t+1}, \boldsymbol{g}_t) \geq \frac{\beta}{2}\|\boldsymbol{g}_{t+1} - \boldsymbol{g}_t\|_p^2 \\
&\iff \eta\langle \boldsymbol{g}_t - \boldsymbol{g}_{t+1}, \boldsymbol{x}_t\rangle \geq D_{\mathcal{R}}(\boldsymbol{g}_{t+1}, \boldsymbol{g}_t) + \frac{\beta}{2}\|\boldsymbol{g}_{t+1} - \boldsymbol{g}_t\|_p^2 \\
&\implies \eta\langle \boldsymbol{g}_t - \boldsymbol{g}_{t+1}, \boldsymbol{x}_t\rangle \geq \frac{\beta}{2}\|\boldsymbol{g}_{t+1} - \boldsymbol{g}_t\|_p^2 + \frac{\beta}{2}\|\boldsymbol{g}_{t+1} - \boldsymbol{g}_t\|_p^2 \\
&\implies \eta\|\boldsymbol{g}_t - \boldsymbol{g}_{t+1}\|_p\|\boldsymbol{x}_t\|_q \geq \beta\|\boldsymbol{g}_{t+1} - \boldsymbol{g}_t\|_p^2 \\
&\implies \frac{\eta}{\beta}n \geq \|\boldsymbol{g}_{t+1} - \boldsymbol{g}_t\|_p,
\end{aligned}$$

since $D_{\mathcal{R}}(\boldsymbol{g}_t, \boldsymbol{g}_t) = 0$ and $\boldsymbol{x}_t \in [0,1]^n$. □

Now we can prove Lemma 3.13:

Proof of Lemma 3.13. From Lemma 3.28 and Lemma 3.29 along with the property of no-regret algorithm such as $\eta = O(\frac{1}{\sqrt{T}})$, we have

$$\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_p = O(\frac{1}{\sqrt{T}}).$$

Now for $q > p$, it is easy to show that:

$$\begin{aligned}
&\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_q \leq \|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_p \\
&\implies \|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_q = O(\frac{1}{\sqrt{T}}).
\end{aligned}$$

For $q < p$, using the Holder's Inequality, we then have:

$$\begin{aligned}
&\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_q \leq n^{1/q-1/p}\|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_p = n^{1/q-1/p}O(\frac{1}{\sqrt{T}}) \\
&\implies \|\boldsymbol{f}_{t+1} - \boldsymbol{f}_t\|_q = O(\frac{1}{\sqrt{T}}).
\end{aligned}$$

We complete the proof. □

Theorem 3.30 (Theorem 3.14)**.** *Let $\mathcal{F} \subset [0,1]^n$ be a convex compact set and let R be a β-strongly convex function with respect to $\|.\|_p$ norm and $\min_{\boldsymbol{f}\in\mathcal{F}} R(\boldsymbol{f}) = 0$. Denote $\|.\|_q$ the dual norm with $1/p+1/q = 1$. Then the AFTRL achieves the external regret of $O(1)$ or forward regret of $O\Big(\sqrt{\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2}\Big)$ against general adversary. More importantly, against a strategic adversary (i.e., no-external regret algorithms such that FTRL, OMD), AFTRL achieves $O(1)$ external regret or $O(1)$ forward regret.*

Proof of Theorem 3.14. Let us first define $\boldsymbol{h}_{t+1}$ as follow

$$\boldsymbol{h}_{t+1} = \operatorname*{argmin}_{\boldsymbol{f}\in\mathcal{F}} H_{t+1}(\boldsymbol{f}) = \langle \boldsymbol{f}, \sum_{s=1}^{t} \boldsymbol{x}_s + \alpha \boldsymbol{x}_{t+1}\rangle + \frac{R(\boldsymbol{f})}{\eta}.$$

Observe that for any sequence of $\boldsymbol{f}_t \in \mathcal{F}$,

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle = \sum_{t=1}^{T}\langle \boldsymbol{f}_t - \boldsymbol{h}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t-1}\rangle + \sum_{t=1}^{T}\langle \boldsymbol{f}_t - \boldsymbol{h}_t, \boldsymbol{x}_{t-1}\rangle + \sum_{t=1}^{T}\langle \boldsymbol{h}_t, \boldsymbol{x}_t\rangle.$$

We now prove by induction that

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t - \boldsymbol{h}_t, \boldsymbol{x}_{t-1}\rangle + \sum_{t=1}^{T}\langle \boldsymbol{h}_t, \boldsymbol{x}_t\rangle \le \frac{1}{\alpha}\langle \boldsymbol{f}', \sum_{t=1}^{T}\boldsymbol{x}_t\rangle + \frac{\alpha-1}{\alpha}\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle + \frac{1}{\eta\alpha}R(\boldsymbol{f}'),\ \forall \boldsymbol{f}' \in \mathcal{F}. \tag{3.25}$$

For $t = 1$, $\boldsymbol{x}_0 = 0$. Since $\boldsymbol{h}_1 = \operatorname{argmin}_{\boldsymbol{f}\in\mathcal{F}}\langle \boldsymbol{f}, \alpha\boldsymbol{x}_1\rangle + \frac{R(\boldsymbol{f})}{\eta}$, we have $\forall \boldsymbol{f}' \in \mathcal{F}$:

$$\langle \boldsymbol{h}_1, \boldsymbol{x}_1\rangle + \frac{R(\boldsymbol{h}_1)}{\eta\alpha} \le \langle \boldsymbol{f}', \boldsymbol{x}_1\rangle + \frac{R(\boldsymbol{f}')}{\eta\alpha} \implies \frac{1}{\alpha}\langle \boldsymbol{h}_1, \boldsymbol{x}_1\rangle \le \frac{1}{\alpha}\langle \boldsymbol{f}', \boldsymbol{x}_1\rangle + \frac{R(\boldsymbol{f}')}{\eta\alpha}, \tag{3.26}$$

since $\alpha \ge 1$ and $R(\boldsymbol{f}') \ge 0\ \forall \boldsymbol{f}' \in \mathcal{F}$. We also have

$$\begin{aligned}
&\langle \boldsymbol{h}_1, \alpha\boldsymbol{x}_1\rangle + \frac{R(\boldsymbol{h}_1)}{\eta} \le \langle \boldsymbol{g}_1, \alpha\boldsymbol{x}_1\rangle + \frac{R(\boldsymbol{g}_1)}{\eta} \\
&= \langle \boldsymbol{g}_1, \boldsymbol{x}_1\rangle + \frac{R(\boldsymbol{g}_1)}{\eta} + (\alpha-1)\langle \boldsymbol{g}_1, \boldsymbol{x}_1\rangle \text{ (By definition of } \boldsymbol{g}_1) \\
&\le \langle \boldsymbol{h}_1, \boldsymbol{x}_1\rangle + \frac{R(\boldsymbol{h}_1)}{\eta} + (\alpha-1)\langle \boldsymbol{g}_1, \boldsymbol{x}_1\rangle \\
&\implies \langle \boldsymbol{h}_1, \alpha\boldsymbol{x}_1\rangle + \frac{R(\boldsymbol{h}_1)}{\eta} \le \langle \boldsymbol{h}_1, \boldsymbol{x}_1\rangle + \frac{R(\boldsymbol{h}_1)}{\eta} + (\alpha-1)\langle \boldsymbol{g}_1, \boldsymbol{x}_1\rangle \\
&\implies \langle \boldsymbol{h}_1, \boldsymbol{x}_1\rangle \le \langle \boldsymbol{g}_1, \boldsymbol{x}_1\rangle.
\end{aligned}$$

Along with Inequality (3.26) we have

$$\langle \boldsymbol{h}_1, \boldsymbol{x}_1\rangle = \frac{1}{\alpha}\langle \boldsymbol{h}_1, \boldsymbol{x}_1\rangle + \frac{\alpha-1}{\alpha}\langle \boldsymbol{h}_1, \boldsymbol{x}_1\rangle \le \frac{1}{\alpha}\langle \boldsymbol{f}', \boldsymbol{x}_1\rangle + \frac{R(\boldsymbol{f}')}{\eta\alpha} + \frac{\alpha-1}{\alpha}\langle \boldsymbol{g}_1, \boldsymbol{x}_1\rangle.$$

Thus, the first step in the induction for $t = 1$ is correct.

For the purpose of induction, suppose that the above inequality holds for $\tau = T-1$. Using $\boldsymbol{f}' = \boldsymbol{f}_T$ and add $\langle \boldsymbol{f}_T - \boldsymbol{h}_T, \boldsymbol{x}_{T-1}\rangle + \langle \boldsymbol{h}_T, \boldsymbol{x}_{T-1}\rangle$ on both sides we have

$$
\begin{aligned}
&\sum_{t=1}^{T}\langle \boldsymbol{f}_t - \boldsymbol{h}_t, \boldsymbol{x}_{t-1}\rangle + \sum_{t=1}^{T}\langle \boldsymbol{h}_t, \boldsymbol{x}_t\rangle \\
&\leq \frac{1}{\alpha}\langle \boldsymbol{f}_T, \sum_{t=1}^{T-1}\boldsymbol{x}_t\rangle + \frac{\alpha-1}{\alpha}\sum_{t=1}^{T-1}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle + \frac{1}{\eta\alpha}R(\boldsymbol{f}_T) + \langle \boldsymbol{f}_T - \boldsymbol{h}_T, \boldsymbol{x}_{T-1}\rangle + \langle \boldsymbol{h}_T, \boldsymbol{x}_T\rangle \\
&= \frac{1}{\alpha}(\langle \boldsymbol{f}_T, \sum_{t=1}^{T-1}\boldsymbol{x}_t + \alpha\boldsymbol{x}_{T-1}\rangle + \frac{1}{\eta}R(\boldsymbol{f}_T)) + \frac{\alpha-1}{\alpha}\sum_{t=1}^{T-1}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle + \langle \boldsymbol{h}_T, \boldsymbol{x}_T - \boldsymbol{x}_{T-1}\rangle \\
&\leq \frac{1}{\alpha}(\langle \boldsymbol{h}_T, \sum_{t=1}^{T-1}\boldsymbol{x}_t + \alpha\boldsymbol{x}_{T-1}\rangle + \frac{1}{\eta}R(\boldsymbol{h}_T)) + \frac{\alpha-1}{\alpha}\sum_{t=1}^{T-1}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle + \langle \boldsymbol{h}_T, \boldsymbol{x}_T - \boldsymbol{x}_{T-1}\rangle \\
&= \frac{1}{\alpha}(\langle \boldsymbol{h}_T, \sum_{t=1}^{T-1}\boldsymbol{x}_t + \alpha\boldsymbol{x}_T\rangle + \frac{1}{\eta}R(\boldsymbol{h}_T)) + \frac{\alpha-1}{\alpha}\sum_{t=1}^{T-1}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \\
&\leq \frac{1}{\alpha}(\langle \boldsymbol{g}_T, \sum_{t=1}^{T-1}\boldsymbol{x}_t + \alpha\boldsymbol{x}_T\rangle + \frac{1}{\eta}R(\boldsymbol{g}_T)) + \frac{\alpha-1}{\alpha}\sum_{t=1}^{T-1}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \\
&= \frac{1}{\alpha}(\langle \boldsymbol{g}_T, \sum_{t=1}^{T-1}\boldsymbol{x}_t + \boldsymbol{x}_T\rangle + \frac{1}{\eta}R(\boldsymbol{g}_T)) + \frac{\alpha-1}{\alpha}\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \\
&\leq \frac{1}{\alpha}(\langle \boldsymbol{f}', \sum_{t=1}^{T-1}\boldsymbol{x}_t + \boldsymbol{x}_T\rangle + \frac{1}{\eta}R(\boldsymbol{f}')) + \frac{\alpha-1}{\alpha}\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \ \forall \boldsymbol{f}'.
\end{aligned}
$$

The proof is derived from the optimality of $\boldsymbol{f}_t, \boldsymbol{g}_t$ and $\boldsymbol{h}_t$. This concludes the inductive argument.

Now, we only need to bound the sum:

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t - \boldsymbol{h}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t-1}\rangle.$$

Using the property of a strongly convex function we have

$$
\begin{aligned}
&F_t(\boldsymbol{h}_t) - F_t(\boldsymbol{f}_t) \geq \frac{\beta}{2\eta}||\boldsymbol{h}_t - \boldsymbol{f}_t||_p^2 \\
&H_t(\boldsymbol{f}_t) - H_t(\boldsymbol{h}_t) \geq \frac{\beta}{2\eta}||\boldsymbol{h}_t - \boldsymbol{f}_t||_p^2 \\
&\quad\implies F_t(\boldsymbol{h}_t) - F_t(\boldsymbol{f}_t) + H_t(\boldsymbol{f}_t) - H_t(\boldsymbol{h}_t) \geq \frac{\beta}{\eta}||\boldsymbol{h}_t - \boldsymbol{f}_t||_p^2 \\
&\quad\iff \alpha\langle \boldsymbol{h}_t - \boldsymbol{f}_t, \boldsymbol{x}_{t-1} - \boldsymbol{x}_t\rangle \geq \frac{\beta}{\eta}||\boldsymbol{h}_t - \boldsymbol{f}_t||_p^2 \\
&\quad\implies ||\boldsymbol{h}_t - \boldsymbol{f}_t||_p||\boldsymbol{x}_{t-1} - \boldsymbol{x}_t||_q \geq \frac{\beta}{\eta\alpha}||\boldsymbol{h}_t - \boldsymbol{f}_t||_p^2 \\
&\quad\implies ||\boldsymbol{x}_{t-1} - \boldsymbol{x}_t||_q \geq \frac{\beta}{\eta\alpha}||\boldsymbol{h}_t - \boldsymbol{f}_t||_p.
\end{aligned}
$$

Thus, we have

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t - \boldsymbol{h}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t-1}\rangle \leq \sum_{t=1}^{T} ||\boldsymbol{f}_t - \boldsymbol{h}_t||p||\boldsymbol{x}_t - \boldsymbol{x}_{t-1}||_q$$
$$\leq \frac{\eta\alpha}{\beta}\sum_{t=1}^{T}(||\boldsymbol{x}_{t-1} - \boldsymbol{x}_t||_q)^2.$$

Along with the Inequality 3.25 we have

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t - \boldsymbol{h}_t, \boldsymbol{x}_{t-1}\rangle + \sum_{t=1}^{T}\langle \boldsymbol{h}_t, \boldsymbol{x}_t\rangle + \sum_{t=1}^{T}\langle \boldsymbol{f}_t - \boldsymbol{h}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t-1}\rangle$$
$$\leq \frac{1}{\alpha}\langle \boldsymbol{f}', \sum_{t=1}^{T}\boldsymbol{x}_t\rangle + \frac{\alpha-1}{\alpha}\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle + \frac{1}{\eta\alpha}R(\boldsymbol{f}') + \frac{\eta\alpha}{\beta}\sum_{t=1}^{T}(||\boldsymbol{x}_{t-1} - \boldsymbol{x}_t||_q)^2,\ \forall \boldsymbol{f}' \in \mathcal{F}$$
$$\iff \sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle \leq \frac{1}{\alpha}\langle \boldsymbol{f}', \sum_{t=1}^{T}\boldsymbol{x}_t\rangle + \frac{\alpha-1}{\alpha}\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle + \frac{1}{\eta\alpha}R(\boldsymbol{f}') + \frac{\eta\alpha}{\beta}\sum_{t=1}^{T}(||\boldsymbol{x}_{t-1} - \boldsymbol{x}_t||_q)^2,\ \forall \boldsymbol{f}' \in \mathcal{F}.$$

Let $\boldsymbol{f}^* = \text{argmin}_{\boldsymbol{f}\in\mathcal{F}}\langle \boldsymbol{f}', \sum_{t=1}^{T}\boldsymbol{x}_t\rangle$ and $\boldsymbol{R} = \max_{\boldsymbol{f}\in\mathcal{F}} R(f)$. Then from the above inequality we have

$$\frac{1}{\alpha}(\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \langle \boldsymbol{f}^*, \sum_{t=1}^{T}\boldsymbol{x}_t\rangle) + \frac{\alpha-1}{\alpha}(\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle) \leq \frac{1}{\eta\alpha}\boldsymbol{R} + \frac{\eta\alpha}{\beta}\sum_{t=1}^{T}(||\boldsymbol{x}_{t-1} - \boldsymbol{x}_t||_q)^2$$

Now, against a general adversary, if $\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \langle \boldsymbol{f}^*, \sum_{t=1}^{T}\boldsymbol{x}_t\rangle \leq 0$ then by definition, AFTRL has $O(1)$ external regret. In case where $\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \langle \boldsymbol{f}^*, \sum_{t=1}^{T}\boldsymbol{x}_t\rangle \geq 0$, using Inequality (3.8) and setting $\eta\alpha = \sqrt{\beta\boldsymbol{R}/(\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2)}$ we have

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \leq \frac{\alpha}{\alpha-1}\sqrt{\boldsymbol{R}\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2/\beta} = O(\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2).$$

For unknown bound $\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2$, we can use the Doubling Trick as shown in Appendix 3.26 to achieve a similar regret bound.

Against a no-external regret adversary, using Lemma 3.13, we then have:

$$\sum_{t=1}^{T}(\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q)^2 = \sum_{t=1}^{T}(O(\frac{1}{\sqrt{T}}))^2 = O(1).$$

Thus, Inequality (3.8) becomes:

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \frac{1}{\alpha}\langle \boldsymbol{f}', \sum_{t=1}^{T}\boldsymbol{x}_t\rangle - \frac{\alpha-1}{\alpha}\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \leq \frac{1}{\eta\alpha}\boldsymbol{R} + \frac{\eta\alpha}{\beta}O(1) = O(1).$$

Following a similar reasoning for general adversary, AFTRL achieves $O(1)$ external regret or $O(1)$ forward regret against no-external regret adversary. □

Theorem 3.31 (Theorem 3.16)**.** *Let $\mathcal{F}$ be a convex set in a Banach space $\mathcal{B}$. Let $\mathcal{R} : \mathcal{B} \to \mathbb{R}$ be a β-strongly convex function on $\mathcal{F}$ with respect to some norm $\|.\|_p$. For any strategy of the environment and any $\boldsymbol{f}' \in \mathcal{F}$, the Accurate Mirror Descent yields*

$$\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \frac{1}{\alpha}\langle \boldsymbol{f}', \boldsymbol{x}_t\rangle - \frac{\alpha-1}{\alpha}\langle \boldsymbol{g}_{t+1}, \boldsymbol{x}_t\rangle \leq \frac{\eta\alpha}{2\beta}\sum_{t=1}^{T}\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q^2 + \frac{R_{max}^2}{\eta\alpha},$$

where $\mathcal{R}_{max}^2 = \max_{\boldsymbol{f}\in\mathcal{F}}\mathcal{R}(\boldsymbol{f}) - \min_{\boldsymbol{f}\in\mathcal{F}}\mathcal{R}(\boldsymbol{f})$.

Proof of Theorem 3.16. We define $\boldsymbol{h}_{t+1}$ as follow:

$$\boldsymbol{h}_{t+1} = \operatorname*{argmin}_{\boldsymbol{h}\in\mathcal{F}} H_{t+1}(\boldsymbol{h}) = \eta\langle \boldsymbol{h}, \alpha\boldsymbol{x}_t\rangle + D_{\mathcal{R}}(\boldsymbol{h}, \boldsymbol{g}_t).$$

For any $\boldsymbol{f}' \in \mathcal{F}$,

$$\begin{aligned}
&\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \frac{1}{\alpha}\langle \boldsymbol{f}', \boldsymbol{x}_t\rangle - \frac{\alpha-1}{\alpha}\langle \boldsymbol{g}_{t+1}, \boldsymbol{x}_t\rangle \\
&= \langle \boldsymbol{f}_t - \boldsymbol{h}_{t+1}, \boldsymbol{x}_t - \boldsymbol{x}_{t-1}\rangle + \langle \boldsymbol{f}_t - \boldsymbol{h}_{t+1}, \boldsymbol{x}_{t-1}\rangle \\
&+ \langle \boldsymbol{h}_{t+1} - \boldsymbol{g}_{t+1}, \boldsymbol{x}_t\rangle + \frac{1}{\alpha}\langle \boldsymbol{g}_{t+1} - \boldsymbol{f}', \boldsymbol{x}_t\rangle.
\end{aligned} \tag{3.27}$$

Using the property of dual norm, we derive

$$\begin{aligned}
&\langle \boldsymbol{f}_t - \boldsymbol{h}_{t+1}, \boldsymbol{x}_t - \boldsymbol{x}_{t-1}\rangle \leq \|\boldsymbol{f}_t - \boldsymbol{h}_{t+1}\|_p\|\boldsymbol{x}_{t-1} - \boldsymbol{x}_t\|_q \\
&\leq \frac{\beta}{2\eta\alpha}\|\boldsymbol{f}_t - \boldsymbol{h}_{t+1}\|_p^2 + \frac{\eta\alpha}{2\beta}\|\boldsymbol{x}_{t-1} - \boldsymbol{x}_t\|_q^2.
\end{aligned} \tag{3.28}$$

We note that for any $\boldsymbol{g} \in \mathcal{F}$ and $\boldsymbol{f} = \operatorname{argmin}_{\boldsymbol{f}\in\mathcal{F}}\langle \boldsymbol{f}, \boldsymbol{x}\rangle + D_{\mathcal{R}}(\boldsymbol{f}, \boldsymbol{c})$, we have the following inequalities (see e.g. Beck and Teboulle (2003))

$$\langle \boldsymbol{f} - \boldsymbol{g}, \boldsymbol{x}\rangle \leq D_{\mathcal{R}}(\boldsymbol{g}, \boldsymbol{c}) - D_{\mathcal{R}}(\boldsymbol{g}, \boldsymbol{f}) - D_{\mathcal{R}}(\boldsymbol{f}, \boldsymbol{c}).$$

This yields

$$\begin{aligned}
\langle \boldsymbol{f}_t - \boldsymbol{h}_{t+1}, \boldsymbol{x}_{t-1}\rangle &\leq \frac{1}{\eta\alpha}(D_{\mathcal{R}}(\boldsymbol{h}_{t+1}, \boldsymbol{g}_t) - D_{\mathcal{R}}(\boldsymbol{h}_{t+1}, \boldsymbol{f}_t) - D_{\mathcal{R}}(\boldsymbol{f}_t, \boldsymbol{g}_t)), \\
\langle \boldsymbol{h}_{t+1} - \boldsymbol{g}_{t+1}, \boldsymbol{x}_t\rangle &\leq \frac{1}{\eta\alpha}(D_{\mathcal{R}}(\boldsymbol{g}_{t+1}, \boldsymbol{g}_t) - D_{\mathcal{R}}(\boldsymbol{g}_{t+1}, \boldsymbol{h}_{t+1}) - D_{\mathcal{R}}(\boldsymbol{h}_{t+1}, \boldsymbol{g}_t)), \\
\langle \boldsymbol{g}_{t+1} - \boldsymbol{f}', \boldsymbol{x}_t\rangle &\leq \frac{1}{\eta}(D_{\mathcal{R}}(\boldsymbol{f}', \boldsymbol{g}_t) - D_{\mathcal{R}}(\boldsymbol{f}', \boldsymbol{g}_{t+1}) - D_{\mathcal{R}}(\boldsymbol{g}_{t+1}, \boldsymbol{g}_t)).
\end{aligned}$$

Summing up the above inequalities we have

$$\begin{aligned}&\langle \boldsymbol{f}_t - \boldsymbol{h}_{t+1}, \boldsymbol{x}_{t-1}\rangle + \langle \boldsymbol{h}_{t+1} - \boldsymbol{g}_{t+1}, \boldsymbol{x}_t\rangle + \frac{1}{\alpha}\langle \boldsymbol{g}_{t+1} - \boldsymbol{f}', \boldsymbol{x}_t\rangle \\ &\leq \frac{1}{\eta\alpha}(D_{\mathcal{R}}(\boldsymbol{f}', \boldsymbol{g}_t) - D_{\mathcal{R}}(\boldsymbol{f}', \boldsymbol{g}_{t+1}) - D_{\mathcal{R}}\left(\boldsymbol{h}_{t+1}, \boldsymbol{f}_t\right) - D_{\mathcal{R}}(\boldsymbol{f}_t, \boldsymbol{g}_t) - D_{\mathcal{R}}(\boldsymbol{g}_{t+1}, \boldsymbol{h}_{t+1}))\,.\end{aligned} \tag{3.29}$$

Using the property of strongly convex function, we have

$$D_{\mathcal{R}}(\boldsymbol{h}_{t+1}, \boldsymbol{f}_t) \geq \frac{\beta}{2}\|\boldsymbol{h}_{t+1} - \boldsymbol{f}_t\|_p^2; \quad D_{\mathcal{R}}(\boldsymbol{f}_t, \boldsymbol{g}_t) \geq \frac{\beta}{2}\|\boldsymbol{f}_t - \boldsymbol{g}_t\|_p^2. \tag{3.30}$$

Putting Inequalities (3.28),(3.29) and (3.30) in Equality (3.27) we derive that

$$\begin{aligned}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \frac{1}{\alpha}\langle \boldsymbol{f}', \boldsymbol{x}_t\rangle - \frac{\alpha - 1}{\alpha}\langle \boldsymbol{g}_{t+1}, \boldsymbol{x}_t\rangle \leq \frac{\eta\alpha}{2\beta}\|\boldsymbol{x}_{t-1} - \boldsymbol{x}_t\|_q^2 \\ + \frac{1}{\eta\alpha}(D_{\mathcal{R}}(\boldsymbol{f}', \boldsymbol{g}_t) - D_{\mathcal{R}}(\boldsymbol{f}', \boldsymbol{g}_{t+1})) - \frac{\beta}{2\eta\alpha}\|\boldsymbol{f}_t - \boldsymbol{g}_t\|_p^2\end{aligned}$$

Summing over $t = 1, \ldots, T$ yields, for any $\boldsymbol{f}' \in \mathcal{F}$,

$$\begin{aligned}&\sum_{t=1}^{T}\langle \boldsymbol{f}_t, \boldsymbol{x}_t\rangle - \frac{1}{\alpha}\langle \boldsymbol{f}', \boldsymbol{x}_t\rangle - \frac{\alpha - 1}{\alpha}\langle \boldsymbol{g}_{t+1}, \boldsymbol{x}_t\rangle \\ &\leq \frac{\eta\alpha}{2\beta}\sum_{t=1}^{T}\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q^2 + \frac{\mathcal{R}_{max}^2}{\eta\alpha} - \frac{\beta}{2\eta\alpha}\sum_{t=1}^{T}\|\boldsymbol{f}_t - \boldsymbol{g}_t\|_p^2 \\ &\leq \frac{\eta\alpha}{2\beta}\sum_{t=1}^{T}\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q^2 + \frac{\mathcal{R}_{max}^2}{\eta\alpha}.\end{aligned}$$

where $\mathcal{R}_{max}^2 = \max_{\boldsymbol{f}\in\mathcal{F}}\mathcal{R}(\boldsymbol{f}) - \min_{\boldsymbol{f}\in\mathcal{F}}\mathcal{R}(\boldsymbol{f})$.

Using the following inequality with any given $\boldsymbol{f}' \in \mathcal{F}$ and $\boldsymbol{g}_{t+1} = \operatorname{argmin}_{\boldsymbol{g}\in\mathcal{F}}\eta\langle \boldsymbol{g}, \boldsymbol{x}_t\rangle + D_{\mathcal{R}}(\boldsymbol{g}, \boldsymbol{g}_t)$ (e.g., see Beck and Teboulle (2003))

$$\eta\langle \boldsymbol{g}_{t+1} - \boldsymbol{f}', \boldsymbol{x}_t\rangle \leq D_{\mathcal{R}}(\boldsymbol{f}', \boldsymbol{g}_t) - D_{\mathcal{R}}(\boldsymbol{f}', \boldsymbol{g}_{t+1}) - D_{\mathcal{R}}(\boldsymbol{g}_{t+1}, \boldsymbol{g}_t)$$

we can derive that, for any $\boldsymbol{f}' \in \mathcal{F}$,

$$\sum_{i=1}^{T}\langle \boldsymbol{g}_{t+1}, \boldsymbol{x}_t\rangle \leq \sum_{i=1}^{T}\langle \boldsymbol{f}', \boldsymbol{x}_t\rangle + \frac{\mathcal{R}_{max}^2}{\eta} - \frac{\beta}{2\eta}\sum_{t=1}^{T}\|\boldsymbol{g}_{t+1} - \boldsymbol{g}_t\|^2.$$

Thus, the regret with respect to $\sum_{i=1}^{T}\langle \boldsymbol{g}_{t+1}, \boldsymbol{x}_t\rangle$ (i.e., forward regret for AMD) is stronger than the (external) regret with respect to $\sum_{i=1}^{T}\langle \boldsymbol{f}', \boldsymbol{x}_t\rangle$, $\forall \boldsymbol{f}' \in \mathcal{F}$. □

Lemma 3.32 (Lemma 3.17)**.** *Let $\boldsymbol{x}_t$, $\boldsymbol{x}_{t+1}$ be two consecutive strategies of a no-regret algorithm (i.e., FTRL, OMD). Then, we have*

$$\langle \boldsymbol{b}, \boldsymbol{x}_{t+1}\rangle - \langle \boldsymbol{c}, \boldsymbol{x}_{t+1}\rangle = O(\frac{1}{\sqrt{T}}),\ \textit{where}\ \boldsymbol{b} = \operatorname*{argmin}_{\boldsymbol{f}\in\mathcal{F}}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle\ ,\ \boldsymbol{c} = \operatorname*{argmin}_{\boldsymbol{f}\in\mathcal{F}}\langle \boldsymbol{f}, \boldsymbol{x}_{t+1}\rangle.$$

Proof of Lemma 3.17. Since $\boldsymbol{b} = \operatorname{argmin}_{\boldsymbol{f}\in\mathcal{F}}\langle \boldsymbol{f}, \boldsymbol{x}_t\rangle$, we then have: $\langle \boldsymbol{b}, \boldsymbol{x}_t\rangle \leq \langle \boldsymbol{c}, \boldsymbol{x}_t\rangle$. Thus, we can derive that:

$$\begin{aligned}&\langle \boldsymbol{b}, \boldsymbol{x}_{t+1}\rangle - \langle \boldsymbol{c}, \boldsymbol{x}_{t+1}\rangle = \langle \boldsymbol{b}, \boldsymbol{x}_{t+1}\rangle - \langle \boldsymbol{b}, \boldsymbol{x}_t\rangle + \langle \boldsymbol{b}, \boldsymbol{x}_t\rangle - \langle \boldsymbol{c}, \boldsymbol{x}_{t+1}\rangle\\ &\leq \langle \boldsymbol{b}, \boldsymbol{x}_{t+1} - \boldsymbol{x}_t\rangle + \langle \boldsymbol{c}, \boldsymbol{x}_t\rangle - \langle \boldsymbol{c}, \boldsymbol{x}_{t+1}\rangle = \langle \boldsymbol{b}, \boldsymbol{x}_{t+1} - \boldsymbol{x}_t\rangle + \langle \boldsymbol{c}, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle.\end{aligned}$$

Using Lemma 3.13 such that $\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\|_1 = O(\frac{1}{\sqrt{T}})$ and $\boldsymbol{b}, \boldsymbol{c} \in [0,1]^n$ we then have:

$$\begin{aligned}&\langle \boldsymbol{b}, \boldsymbol{x}_{t+1}\rangle - \langle \boldsymbol{c}, \boldsymbol{x}_{t+1}\rangle = \langle \boldsymbol{b}, \boldsymbol{x}_{t+1} - \boldsymbol{x}_t\rangle + \langle \boldsymbol{c}, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle\\ &\leq \|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\|_1 + \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|_1 \leq 2O(\frac{1}{\sqrt{T}}) = O(\frac{1}{\sqrt{T}}).\end{aligned}$$

The proof is complete. □

3.10.2 Proofs of Last Round Convergence of AMWU

In our proof, we use the notation of β-closeness:

Definition 3.33 (β**-closeness** Mehta et al. (2017))**.** Assume $\beta > 0$. A point $(\boldsymbol{f}, \boldsymbol{y}) \in \Delta_n \times \Delta_m$ is β-close if for each $i \in [n]$, it holds $\boldsymbol{f}_i \leq \beta$ or $|\boldsymbol{f}^\top \boldsymbol{A}\boldsymbol{y} - (\boldsymbol{A}\boldsymbol{y})_i| \leq \beta$ and for each $j \in [m]$, it holds $\boldsymbol{y}_i \leq \beta$ or $|\boldsymbol{f}^\top \boldsymbol{A}\boldsymbol{y} - (\boldsymbol{A}^\top \boldsymbol{x})_i| \leq \beta$.

We can now provide the full proof of last round convergence for AMWU.

3.10.2.1 Decreasing K-L distance

In this subsection, part of our analysis bases on the linear variant of AMWU with the following update rule:

$$\boldsymbol{f}_{t+1}(i) = \frac{\boldsymbol{f}_t(i)(1 + \eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_t - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y}_{t-1}))}{\sum_j \boldsymbol{f}_t(j)(1 + \eta((\alpha+1)\boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}_t - \alpha \boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y}_{t-1}))}.$$

Since the variant' update rule does not contain the exponential part, it reduces the complexity in the analysis. We first quantify the distance between two consecutive updates of AMWU by the following lemma:

Lemma 3.34. *Let $\boldsymbol{f} \in \Delta_n$ be the vector of the max player, $\boldsymbol{w}, \boldsymbol{z} \in \Delta_m$ such that $\|\boldsymbol{w} - \boldsymbol{z}\|_1 = O(\eta)$, $\eta\alpha = O(1)$ and suppose $\boldsymbol{f}', \boldsymbol{f}''$ are the next iterates of AMWU and its linear variant with current vector $\boldsymbol{f}$ and vectors $\boldsymbol{w}, \boldsymbol{z}$ of the min player. It holds that*

$$\|\boldsymbol{f}' - \boldsymbol{f}''\|_1 \text{ is } O(\eta^2) \text{ and } \|\boldsymbol{f}' - \boldsymbol{f}\|_1,\ \|\boldsymbol{f}'' - \boldsymbol{f}\|_1 \text{ are } O(\eta).$$

Analogously, it holds for vector $\boldsymbol{y} \in \Delta_m$ of the min player and its next iterates.

Proof of Lemma 3.34. Let η be sufficiently small (smaller than maximum in absolute value entry of $\boldsymbol{A}$). From the assumption that $\|\boldsymbol{w}-\boldsymbol{z}\|_1 = O(\eta)$and $O(\eta\alpha) = O(1)$ we have

$$(\alpha+1)(\boldsymbol{A}\boldsymbol{w})_i - \alpha(\boldsymbol{A}\boldsymbol{z})_i = (\boldsymbol{A}\boldsymbol{w})_i + O(1).$$

Thus, we can derive the following equalities:

$$\begin{aligned}
|\boldsymbol{f}'_i - \boldsymbol{f}''_i| &= \boldsymbol{f}_i \left| \frac{e^{\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_i - \alpha(\boldsymbol{A}\boldsymbol{z})_i)}}{\sum_j \boldsymbol{f}_j e^{\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_j - \alpha(\boldsymbol{A}\boldsymbol{z})_j)}} - \frac{1+\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_i - \alpha(\boldsymbol{A}\boldsymbol{z})_i)}{\sum_j \boldsymbol{f}_j(1+\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_j - \alpha(\boldsymbol{A}\boldsymbol{z})_j))} \right| \\
&= \boldsymbol{f}_i \left| \frac{1+\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_i - \alpha(\boldsymbol{A}\boldsymbol{z})_i) \pm O(\eta^2)}{\sum_j \boldsymbol{f}_j(1+\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_j - \alpha(\boldsymbol{A}\boldsymbol{z})_j)) \pm O(\eta^2)} - \frac{1+\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_i - \alpha(\boldsymbol{A}\boldsymbol{z})_i)}{\sum_j \boldsymbol{f}_j(1+\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_j - \alpha(\boldsymbol{A}\boldsymbol{z})_j))} \right| \\
&= \boldsymbol{f}_i O(\eta^2).
\end{aligned}$$

and hence $\|\boldsymbol{f}' - \boldsymbol{f}''\|_1$ is $O(\eta^2)$. Moreover we have that

$$\begin{aligned}
|\boldsymbol{f}_i - \boldsymbol{f}''_i| &= \boldsymbol{f}_i \left| 1 - \frac{1+\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_i - \alpha(\boldsymbol{A}\boldsymbol{z})_i)}{\sum_j \boldsymbol{f}_j(1+\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_j - \alpha(\boldsymbol{A}\boldsymbol{z})_j))} \right| \\
&= \boldsymbol{f}_i \left| \frac{\sum_j \boldsymbol{f}_j(1+\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_j - \alpha(\boldsymbol{A}\boldsymbol{z})_j)) - (1+\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_i - \alpha(\boldsymbol{A}\boldsymbol{z})_i))}{\sum_j \boldsymbol{f}_j(1+\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_j - \alpha(\boldsymbol{A}\boldsymbol{z})_j))} \right| \\
&= \boldsymbol{f}_i \left| \frac{\sum_j \boldsymbol{f}_j(\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_j - \alpha(\boldsymbol{A}\boldsymbol{z})_j)) - \eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_i - \alpha(\boldsymbol{A}\boldsymbol{z})_i)}{\sum_j \boldsymbol{f}_j(1+\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_j - \alpha(\boldsymbol{A}\boldsymbol{z})_j))} \right| \\
&= \boldsymbol{f}_i \left| \frac{\eta\left(\sum_j \boldsymbol{f}_j((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_j - \alpha(\boldsymbol{A}\boldsymbol{z})_j) - ((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_i - \alpha(\boldsymbol{A}\boldsymbol{z})_i)\right)}{\sum_j \boldsymbol{f}_j(1+\eta((\alpha+1)(\boldsymbol{A}\boldsymbol{w})_j - \alpha(\boldsymbol{A}\boldsymbol{z})_j))} \right| \\
&= \boldsymbol{f}_i O(\eta).
\end{aligned}$$

We can derive the third part of the lemma by using the triangle inequality with the two above proofs. □

We needs the following lemmas in order to prove Lemma 3.37:

Lemma 3.35. *Let $\boldsymbol{f} \in \Delta_n$ be the vector of the max player, $\boldsymbol{w}, \boldsymbol{z} \in \Delta_m$ such that $\|\boldsymbol{w}-\boldsymbol{z}\|_1 = O(\eta)$, $\eta\alpha = O(1)$ and suppose $\boldsymbol{f}', \boldsymbol{f}''$ are the next iterates of AMWU and its linear variant with current vector $\boldsymbol{f}$ and vectors $\boldsymbol{w}, \boldsymbol{z}$ of the min player. It holds that (for η sufficiently small)*

$$\begin{aligned}
&\eta(\boldsymbol{f}' - \boldsymbol{f})^\top \boldsymbol{A}((\alpha+1)\boldsymbol{w} - \alpha\boldsymbol{z}) \\
&= \eta(\boldsymbol{f}'' - \boldsymbol{f})^\top \boldsymbol{A}((\alpha+1)\boldsymbol{w} - \alpha\boldsymbol{z}) - O(\eta^3) \\
&= (1-O(\eta))\eta^2 \sum_i \boldsymbol{f}_i((\boldsymbol{f} - \boldsymbol{e}_i)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{w} - \alpha\boldsymbol{z}))^2 - O(\eta^3) \\
&= (1-O(\eta))\eta^2 \sum_i \boldsymbol{f}'_i((\boldsymbol{f}' - \boldsymbol{e}_i)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{w} - \alpha\boldsymbol{z}))^2 - O(\eta^3).
\end{aligned}$$

Proof. By following Lemma 3.34, we only need to prove the second equality. Set $\boldsymbol{B} = (\mathbb{1}_n \mathbb{1}_m{}^\top + \eta \boldsymbol{A})$. We have that $f_i'' = f_i \frac{(\boldsymbol{B}((\alpha+1)\boldsymbol{w}) - \alpha \boldsymbol{z})_i}{\boldsymbol{f}^\top \boldsymbol{B}((\alpha+1)\boldsymbol{w}) - \alpha \boldsymbol{z})}$ following the definition of linear AMWU. We can derive that

$$
\begin{aligned}
&(\boldsymbol{f}''^\top \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}).(\boldsymbol{f}^\top \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}) \\
&= \sum_{ij} \boldsymbol{B}_{ij} \boldsymbol{f}_i''((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z})_j.(\boldsymbol{f}^\top \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}) \\
&= \sum_{ij} \boldsymbol{B}_{ij} f_i \frac{(\boldsymbol{B}((\alpha+1)\boldsymbol{w}) - \alpha \boldsymbol{z})_i}{\boldsymbol{f}^\top \boldsymbol{B}((\alpha+1)\boldsymbol{w}) - \alpha \boldsymbol{z})}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z})_j.(\boldsymbol{f}^\top \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}) \\
&= \sum_{ij} \boldsymbol{B}_{ij} f_i (\boldsymbol{B}((\alpha+1)\boldsymbol{w}) - \alpha \boldsymbol{z})_i ((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z})_j = \sum_i f_i (\boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z})_i)^2 \\
&= (\boldsymbol{f}^T \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}))^2 + \sum_i \boldsymbol{f}_i (\boldsymbol{f}^T \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}) - (\boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}))_i)^2.
\end{aligned}
$$

Thus we have

$$
\begin{aligned}
&(\boldsymbol{f}''^\top \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}).(\boldsymbol{f}^\top \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}) \\
&= (\boldsymbol{f}^T \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}))^2 + \sum_i \boldsymbol{f}_i (\boldsymbol{f}^T \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}) - (\boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}))_i)^2.
\end{aligned} \tag{3.31}
$$

Since our assumption that $\|\boldsymbol{w} - \boldsymbol{z}\|_1 = O(\eta)$ and $\eta\alpha = O(1)$, we then have:

$$
\|\boldsymbol{A}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z})\| = \|\alpha \boldsymbol{A}(\boldsymbol{w} - \boldsymbol{z}) + \boldsymbol{A}\boldsymbol{w}\| = O(\alpha\eta) + O(1) = O(1).
$$

Thus we also have:

$$
\boldsymbol{f}^T \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}) = 1 \pm O(\eta).
$$

Devide both sides of Equation (3.31) by $\boldsymbol{f}^T \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z})$ we have

$$
\begin{aligned}
&(\boldsymbol{f}''^\top \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}) \\
&= (\boldsymbol{f}^T \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z})) + (1 - O(\eta)) \sum_i \boldsymbol{f}_i (\boldsymbol{f}^T \boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}) - (\boldsymbol{B}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}))_i)^2 \\
&= \eta \boldsymbol{f}^\top \boldsymbol{A}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}) + (1 - O(\eta))\eta^2 \sum_i \boldsymbol{f}_i ((\boldsymbol{f} - \boldsymbol{e}_i)^T \boldsymbol{A}((\alpha+1)\boldsymbol{w} - \alpha \boldsymbol{z}))^2.
\end{aligned}
$$

Thus, the second equality is proven. Other equalities come directly as the result of Lemma 3.34. □

Furthermore, from the above lemma, if we impose the condition:

$$
\eta\alpha = \eta^b,
$$

where b is in $(0, 1]$. Note that this condition does not contradict to $\eta\alpha = O(1)$. Then from the above lemma we have

$$\begin{aligned}
&\eta(\boldsymbol{f}' - \boldsymbol{f})^\top \boldsymbol{A}((\alpha+1)\boldsymbol{w} - \alpha\boldsymbol{z}) = \\
&\eta(\boldsymbol{f}' - \boldsymbol{f})^\top \boldsymbol{A}\boldsymbol{w} + \eta\alpha(\boldsymbol{f}' - \boldsymbol{f})^\top \boldsymbol{A}(\boldsymbol{w} - \boldsymbol{z}) \\
&= \eta(\boldsymbol{f}' - \boldsymbol{f})^\top \boldsymbol{A}\boldsymbol{w} + \eta^b O(\eta^2) \\
&\implies \eta(\boldsymbol{f}' - \boldsymbol{f})^\top \boldsymbol{A}\boldsymbol{w} = (1 - O(\eta))\eta^2 \sum_i \boldsymbol{f}_i((\boldsymbol{f} - \boldsymbol{e}_i)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{w} - \alpha\boldsymbol{z}))^2 - \eta^b O(\eta^2) \\
&= (1 - O(\eta))\eta^2 \sum_i \boldsymbol{f}'_i((\boldsymbol{f}' - \boldsymbol{e}_i)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{w} - \alpha\boldsymbol{z}))^2 - \eta^b O(\eta^2).
\end{aligned}$$

Similarly, we have the following lemma for the min player:

Lemma 3.36. *Let $\boldsymbol{y} \in \Delta_m$, $\boldsymbol{w}, \boldsymbol{z} \in \Delta_n$ and suppose $\boldsymbol{y}'$ is the next iterate of AMWU with current vector $\boldsymbol{y}$ and inputs $\boldsymbol{w}, \boldsymbol{z}$. Furthermore, assume that $\|\boldsymbol{w} - \boldsymbol{z}\|_1 = O(\eta)$ and $\eta\alpha = \eta^b$ for $0 \le b \le 1$. It holds that (for η sufficiently small):*

$$\begin{aligned}
&\eta(\boldsymbol{y}' - \boldsymbol{y})^\top \boldsymbol{A}^\top(-\boldsymbol{w}) \\
&= (1 - O(\eta))\eta^2 \sum_i \boldsymbol{y}'_i((\boldsymbol{y}' - \boldsymbol{e}_i)^\top \boldsymbol{A}^\top((\alpha\boldsymbol{z} - (\alpha+1)\boldsymbol{w}))^2 - \eta^b O(\eta^2).
\end{aligned}$$

In order to prove Theorem 3.39, we need the following lemmas:

Lemma 3.37. *Let $(\boldsymbol{f}_t, \boldsymbol{y}_t)$ be the t-th iteration of AMWU dynamic. For each time step $t \ge 2$ it holds that:*

$$\begin{aligned}
&\eta\boldsymbol{f}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t - \eta\boldsymbol{f}_t^\top \boldsymbol{A}\boldsymbol{y}_{t-1} \\
&= -(1 - O(\eta))\eta^2 \sum_i \boldsymbol{f}_t(i)((\boldsymbol{f}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1}))^2 + \\
&- (1 - O(\eta))\eta^2 \sum_i \boldsymbol{y}_t(i)((\boldsymbol{y}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}^\top((\alpha\boldsymbol{y}_{t-1} - (\alpha+1)\boldsymbol{y}_t))^2 + \eta^b O(\eta^2)
\end{aligned}$$

Proof of Lemma 3.37.

$$\begin{aligned}
&\eta\boldsymbol{f}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t - \eta\boldsymbol{f}_t^\top \boldsymbol{A}\boldsymbol{y}_{t-1} \\
&\le -(1 - O(\eta))\eta^2 \sum_i \boldsymbol{f}_t(i)((\boldsymbol{f}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{y}_{t-1} - \alpha\boldsymbol{y}_{t-2}))^2 + \\
&- (1 - O(\eta))\eta^2 \sum_i \boldsymbol{y}_t(i)((\boldsymbol{y}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}^\top((\alpha\boldsymbol{y}_{t-2} - (\alpha+1)\boldsymbol{y}_{t-1}))^2 + \eta^b O(\eta^2) \\
&= -(1 - O(\eta))\eta^2 \sum_i \boldsymbol{f}_t(i)((\boldsymbol{f}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1}))^2 - (1 - O(\eta))\eta^2\eta^{2b} + \\
&- (1 - O(\eta))\eta^2 \sum_i \boldsymbol{y}_t(i)((\boldsymbol{y}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}^\top((\alpha\boldsymbol{y}_{t-1} - (\alpha+1)\boldsymbol{y}_t))^2 - (1 - O(\eta))\eta^2\eta^{2b} + \eta^b O(\eta^2) \\
&= -(1 - O(\eta))\eta^2 \sum_i \boldsymbol{f}_t(i)((\boldsymbol{f}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1}))^2 + \\
&- (1 - O(\eta))\eta^2 \sum_i \boldsymbol{y}_t(i)((\boldsymbol{y}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}^\top((\alpha\boldsymbol{y}_{t-1} - (\alpha+1)\boldsymbol{y}_t))^2 + \eta^b O(\eta^2)
\end{aligned}$$

□

Lemma 3.38. *Let $(\boldsymbol{f}_t, \boldsymbol{y}_t)$ denote the t-th iterate of AMWU dynamics. It holds for $t \geq 2$ that*

$$\boldsymbol{f}^{*\top}\boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1}) \geq \boldsymbol{f}^{*\top}\boldsymbol{A}\boldsymbol{y}^* \text{ and}$$
$$((\alpha+1)\boldsymbol{f}_t - \alpha\boldsymbol{f}_{t-1})^\top\boldsymbol{A}\boldsymbol{y}^* \leq \boldsymbol{f}^{*\top}\boldsymbol{A}\boldsymbol{y}^*$$

Proof of Lemma 3.38. It is sufficient to show that $((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1}) \in \Delta_m$ and $((\alpha+1)\boldsymbol{f}_t - \alpha\boldsymbol{f}_{t-1}) \in \Delta_n$. From Lemma 3.34 we have $\boldsymbol{f}_t(i) = (1 - O(\eta))\boldsymbol{f}_{t-1}(i)$. Thus, in order to show that $((\alpha+1)\boldsymbol{f}_t(i) - \alpha\boldsymbol{f}_{t-1}(i)) \geq 0$ we need to show that:

$$(1 - O(\eta)) \geq \frac{\alpha}{\alpha+1}$$
$$\iff 1 \geq (\alpha+1)O(\eta),$$

which is true since $\alpha\eta = \eta^b, b \in [0,1]$ and η is small enough. □

Theorem 3.39. *Let $(\boldsymbol{f}^*, \boldsymbol{y}^*)$ be the unique optimal minimax equilibrium and η sufficiently small. Assume that $\alpha\eta = \eta^b$ where $b \in [0,1]$. Then*

$$RE((\boldsymbol{f}^*, \boldsymbol{y}^*)||(\boldsymbol{f}_t, \boldsymbol{y}_t))$$

is decreasing with time t by η^{2+b} unless $(\boldsymbol{f}_t, \boldsymbol{y}_t)$ is $O(\eta^{b/3})-$close.

Proof of Theorem 3.39. We compute the difference in relative entropy distance between two connected strategies:

$$\begin{aligned}
&RE((\boldsymbol{f}^*, \boldsymbol{y}^*)||(\boldsymbol{f}_{t+1}, \boldsymbol{y}_{t+1})) - RE((\boldsymbol{f}^*, \boldsymbol{y}^*)||(\boldsymbol{f}_t, \boldsymbol{y}_t)) \\
&= -\left(\sum_i \boldsymbol{f}^*(i)\log(\frac{\boldsymbol{f}_{t+1}(i)}{\boldsymbol{f}_t(i)}) + \sum_i \boldsymbol{y}^*(i)\log(\frac{\boldsymbol{y}_{t+1}(i)}{\boldsymbol{y}_t(i)})\right) \\
&= -\left(\sum_i \boldsymbol{f}^*(i)\log(e^{\eta((\alpha+1)\boldsymbol{A}\boldsymbol{y}_t - \alpha\boldsymbol{A}\boldsymbol{y}_{t-1})(i)}) + \sum_i \boldsymbol{y}^*(i)\log(e^{\eta(-(\alpha+1)\boldsymbol{A}^\top\boldsymbol{f}_t + \alpha\boldsymbol{A}^\top\boldsymbol{f}_{t-1})(i)})\right) \\
&+ \log\left(\sum_i \boldsymbol{f}_t(i)e^{\eta((\alpha+1)\boldsymbol{A}\boldsymbol{y}_t - \alpha\boldsymbol{A}\boldsymbol{y}_{t-1})(i)}\right) + \log\left(\sum_i \boldsymbol{y}_t(i)e^{\eta(-(\alpha+1)\boldsymbol{A}^\top\boldsymbol{f}_t + \alpha\boldsymbol{A}^\top\boldsymbol{f}_{t-1})(i)}\right) \\
&= -\eta x^{*\top}\boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1}) - \eta y^{*\top}\boldsymbol{A}^\top(-(\alpha+1)\boldsymbol{f}_t + \alpha\boldsymbol{f}_{t-1}) + \\
&\log\left(\sum_i \boldsymbol{f}_t(i)e^{\eta((\alpha+1)\boldsymbol{A}\boldsymbol{y}_t - \alpha\boldsymbol{A}\boldsymbol{y}_{t-1})(i)}\right) + \log\left(\sum_i \boldsymbol{y}_t(i)e^{\eta(-(\alpha+1)\boldsymbol{A}^\top\boldsymbol{f}_t + \alpha\boldsymbol{A}^\top\boldsymbol{f}_{t-1})(i)}\right).
\end{aligned}$$

From Lemma 3.38 we have

$$-\eta\boldsymbol{f}^{*\top}\boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1}) - \eta\boldsymbol{y}^{*\top}\boldsymbol{A}^\top(-(\alpha+1)\boldsymbol{f}_t + \alpha\boldsymbol{f}_{t-1}) \leq 0.$$

Thus we have

$$
\begin{aligned}
&RE((\boldsymbol{f}^*, \boldsymbol{y}^*)||(\boldsymbol{f}_{t+1}, \boldsymbol{y}_{t+1})) - RE((\boldsymbol{f}^*, \boldsymbol{y}^*)||(\boldsymbol{f}_t, \boldsymbol{y}_t)) \\
&\leq \log\left(\sum_i \boldsymbol{f}_t(i) e^{\eta((\alpha+1)\boldsymbol{A}\boldsymbol{y}_t - \alpha\boldsymbol{A}\boldsymbol{y}_{t-1})(i)}\right) + \log\left(\sum_i \boldsymbol{y}_t(i) e^{\eta(-(\alpha+1)\boldsymbol{A}^\top \boldsymbol{f}_t + \alpha\boldsymbol{A}^\top \boldsymbol{f}_{t-1})(i)}\right) \\
&= \log\left(\sum_i \boldsymbol{f}_t(i) e^{\eta((\alpha+1)((\boldsymbol{A}\boldsymbol{y}_t)(i) - \boldsymbol{f}_t^\top \boldsymbol{A}\boldsymbol{y}_t) - \alpha((\boldsymbol{A}\boldsymbol{y}_{t-1})(i) - \boldsymbol{f}_t^\top \boldsymbol{A}\boldsymbol{y}_{t-1}))}\right) \\
&+ \log\left(\sum_i \boldsymbol{y}_t(i) e^{\eta(-(\alpha+1)((\boldsymbol{A}^\top \boldsymbol{f}_t)(i) - \boldsymbol{f}_t^\top \boldsymbol{A}\boldsymbol{y}_t) + \alpha((\boldsymbol{A}^\top \boldsymbol{f}_{t-1})(i) - \boldsymbol{f}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t))}\right) + \alpha\eta(\boldsymbol{f}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t - \boldsymbol{f}_t^\top \boldsymbol{A}\boldsymbol{y}_{t-1}) \\
&= \log\left(\sum_i \boldsymbol{f}_t(i) e^{\eta((\boldsymbol{e}_i - \boldsymbol{f}_t)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1}))}\right) + \log\left(\sum_i \boldsymbol{y}_t(i) e^{\eta((-(\alpha+1)\boldsymbol{f}_t + \alpha\boldsymbol{f}_{t-1})^\top \boldsymbol{A}(\boldsymbol{e}_i - \boldsymbol{y}_t))}\right) \\
&+ \eta^b(\boldsymbol{f}_{t-1}^\top \boldsymbol{A}\boldsymbol{y}_t - \boldsymbol{f}_t^\top \boldsymbol{A}\boldsymbol{y}_{t-1}).
\end{aligned}
$$

Using the Taylor approximation (η is suffciently small) to the function e^x (i.e., $e^x = 1 + x + \frac{1}{2}x^2$) and $\log(1+x) < x$ for $x > 0$, we then have:

$$
\begin{aligned}
&\log\left(\sum_i \boldsymbol{f}_t(i) e^{\eta((\boldsymbol{e}_i - \boldsymbol{f}_t)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1}))}\right) \\
&\leq \log(\sum_i x_t(i)(1 + \eta((\boldsymbol{e}_i - \boldsymbol{f}_t)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1})))+ \\
&\sum_i \boldsymbol{f}_t(i)((\frac{1}{2} + O(\eta^b))\eta^2((\boldsymbol{e}_i - \boldsymbol{f}_t)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1}))^2) \\
&= \log\left(1 + \sum_i \boldsymbol{f}_t(i)((\frac{1}{2} + O(\eta^b))\eta^2((\boldsymbol{e}_i - \boldsymbol{f}_t)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1}))^2\right) \\
&\leq \sum_i \boldsymbol{f}_t(i)((\frac{1}{2} + O(\eta^b))\eta^2((\boldsymbol{e}_i - \boldsymbol{f}_t)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1}))^2.
\end{aligned}
$$

Along with Lemma 3.37, we then have:

$$
\begin{aligned}
&RE((\boldsymbol{f}^*, \boldsymbol{y}^*)||(\boldsymbol{f}_{t+1}, \boldsymbol{y}_{t+1})) - RE((\boldsymbol{f}^*, \boldsymbol{y}^*)||(\boldsymbol{f}_t, \boldsymbol{y}_t)) \\
&\leq \sum_i (\frac{1}{2} + O(\eta^b))\eta^2 \boldsymbol{f}_t(i)((\boldsymbol{e}_i - \boldsymbol{f}_t)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1}))^2 + \\
&\sum_i (\frac{1}{2} + O(\eta^b))\eta^2 \boldsymbol{y}_t(i)((\boldsymbol{y}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}^\top((\alpha\boldsymbol{y}_{t-1} - (\alpha+1)\boldsymbol{y}_t))^2 \\
&- \frac{\eta^b}{\eta}(1 - O(\eta))\eta^2 \sum_i \boldsymbol{f}_t(i)((\boldsymbol{f}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha\boldsymbol{y}_{t-1}))^2 - \\
&\frac{\eta^b}{\eta}(1 - O(\eta))\eta^2 \sum_i \boldsymbol{y}_t(i)((\boldsymbol{y}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}^\top((\alpha\boldsymbol{y}_{t-1} - (\alpha+1)\boldsymbol{y}_t))^2 + \frac{\eta^b}{\eta}\eta^b O(\eta^2)
\end{aligned}
$$

$$\leq -(\frac{1}{2} - O(\eta^b))\eta^2 \sum_i \boldsymbol{f}_t(i)((\boldsymbol{f}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}((\alpha+1)\boldsymbol{y}_t - \alpha \boldsymbol{y}_{t-1}))^2 -$$
$$(\frac{1}{2} - O(\eta^b))\eta^2 \sum_i \boldsymbol{y}_t(i)((\boldsymbol{y}_t - \boldsymbol{e}_i)^\top \boldsymbol{A}^\top((\alpha \boldsymbol{y}_{t-1} - (\alpha+1)\boldsymbol{y}_t))^2 + \eta^b O(\eta^2).$$

Since $\frac{\eta^b}{\eta} > 1$. Now, it is clear that as long as $(\boldsymbol{f}_t, \boldsymbol{y}_t)$ and thus $(\boldsymbol{f}_{t-1}, \boldsymbol{y}_{t-1})$ is not $O(\eta^{b/3})$-close, from the above inequalities we get:

$$RE((\boldsymbol{f}^*, \boldsymbol{y}^*)||(\boldsymbol{f}_{t+1}, \boldsymbol{y}_{t+1})) - RE((\boldsymbol{f}^*, \boldsymbol{y}^*)||(\boldsymbol{f}_t, \boldsymbol{y}_t)) \leq -\Omega(\eta^{b+2}),$$

or the relative entropy distance decreases at least a factor of η^{b+2} and the claim follows. □

3.10.2.2 $\eta^{b/3}$-closeness implies closeness to optimum

We first need the following lemma:

Lemma 3.40. *Let $i \in \text{Supp}(\boldsymbol{f}^*)$ and $j \in \text{Supp}(\boldsymbol{y}^*)$. It holds that $x_T(i) \geq \frac{1}{2}\eta^{b/3}$ and $y_T(i) \geq \frac{1}{2}\eta^{b/3}$ as long as*

$$\eta^{b/3} \leq \min_{s \in \text{Supp}(\boldsymbol{f}^*)} \frac{1}{(nm)^{1/\boldsymbol{f}^*(s)}}, \min_{s \in \text{Supp}(\boldsymbol{y}^*)} \frac{1}{(nm)^{1/\boldsymbol{y}^*(s)}}.$$

Proof. By definition of T, the K-L divergence is decreasing for $2 \leq t \leq T-1$, thus

$$RE((\boldsymbol{f}^*, \boldsymbol{y}^*)||(\boldsymbol{f}_{T-1}, \boldsymbol{y}_{T-1})) < RE((\boldsymbol{f}^*, \boldsymbol{y}^*)||(\boldsymbol{f}_1, \boldsymbol{y}_1)).$$

This implies that:

$$\boldsymbol{f}^*(i) \log\big(\frac{1}{\boldsymbol{f}_{T-1}}(i)\big) \leq \sum_j \boldsymbol{f}^*(j) \log\big(\frac{1}{\boldsymbol{f}_{T-1}}(j)\big)$$
$$\leq \sum_i \boldsymbol{f}^*(i) \log\big(\frac{1}{\boldsymbol{f}_1(i)}\big) + \sum_i \boldsymbol{y}^*(i) \log\big(\frac{1}{\boldsymbol{y}_1(i)}\big) = \log(nm)$$
$$\implies \boldsymbol{f}_T(i) > \frac{1}{(mn)^{1/\boldsymbol{f}^*(i)}} \geq \eta^{b/3}.$$

Since $|\boldsymbol{f}_T(i) - \boldsymbol{f}_{T-1}(i)|$ is $O(\eta)$, the result follows. □

Using the above lemma, we can follow the same argument as in Theorem 3.2 of Daskalakis and Panageas (2019) to prove the following theorem:

Theorem 3.41. *Assume $(\boldsymbol{f}^*, \boldsymbol{y}^*)$ is unique optimal solution of the problem. Let T be the first time KL divergence does not decrease by $\Omega(\eta^{b+2})$. It follows that as $\eta \to 0$, the $\eta^{b/3}$-close point $(\boldsymbol{f}_T, \boldsymbol{y}_T)$ has distance from $(\boldsymbol{f}^*, \boldsymbol{y}^*)$ that goes to zero:*

$$\lim_{\eta \to 0} \|(\boldsymbol{f}^*, \boldsymbol{y}^*) - (\boldsymbol{f}_T, \boldsymbol{y}_T)\|_1 = 0.$$

Proof. From Lemma 3.40 and the definition of T we have $|(\boldsymbol{A}\boldsymbol{y}_T)_i - \boldsymbol{f}_T{}^\top \boldsymbol{A}\boldsymbol{y}_T|$ is $O(\eta^{1/3})$ for i in support of $\boldsymbol{f}^*$ and $|(\boldsymbol{f}_T^\top \boldsymbol{A})_j - \boldsymbol{f}_T^\top \boldsymbol{A}\boldsymbol{y}_T|$ is $O(\eta^{1/3})$ for j in support of $\boldsymbol{y}^*$. We consider $(\boldsymbol{w}_T, \boldsymbol{z}_T)$ the project of $(\boldsymbol{f}_T, \boldsymbol{y}_T)$ by removing all the coordinates with mass less than $\frac{1}{2}\eta^{b/3}$ and rescales it. We have the following relationship:

$$\lim_{\eta \to 0} \|(\boldsymbol{f}_T, \boldsymbol{y}_T) - (\boldsymbol{w}_T, \boldsymbol{z}_T)\| = 0. \tag{3.32}$$

Since for all the cordinates in $\boldsymbol{w}$ and $\boldsymbol{z}$, it holds that $|(\hat{\boldsymbol{A}}\boldsymbol{z}_T)_i - \boldsymbol{w}_T{}^\top \hat{\boldsymbol{A}}\boldsymbol{z}_T|$ and $|(\boldsymbol{w}_T^\top \hat{\boldsymbol{A}})_j - \boldsymbol{w}_T^\top \hat{\boldsymbol{A}}\boldsymbol{z}_T|$ are $O(\eta^{b/3})$, thus $(\boldsymbol{w}, \boldsymbol{z})$ is $O(\eta^{b/3})$-approximate solution of the game $\hat{\boldsymbol{A}}$. Using the following lemma:

Lemma 3.42 (Claim 3.5 in Daskalakis and Panageas (2019))**.** *Let $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ be the unique optimal solution of the game. For every $\epsilon > 0$, there exists an γ so that for every γ-approximate solution $(\boldsymbol{x}, \boldsymbol{y})$ we get that $|x_i - x_1^*| < \epsilon$ for all $i \in [n]$. Analogously holds for player $\boldsymbol{y}$.*

Using the above lemma with $\epsilon = \eta^{b/3}$ and sufficiently small η, we have $|w_i| < \eta^{b/3}$ for every i not in the support of $\boldsymbol{x}^*$. Since the subgame $\hat{\boldsymbol{A}}$ contains all the pure strategy in the NE support of game $\boldsymbol{A}$, subgame $\hat{\boldsymbol{A}}$ will also have a unique NE with the same weight as in the game $\boldsymbol{A}$. Thus we have

$$\lim_{\eta \to 0} \|(\boldsymbol{w}_T, \boldsymbol{z}_T) - (\boldsymbol{f}^*, \boldsymbol{y}^*)\| = 0. \tag{3.33}$$

Combining Equation (3.32) and (3.33) gives us the proof. □

3.10.2.3 Proof of local convergence

We use the following well-known fact in dynamical systems to prove the local convergence:

Proposition 3.43 (see Galor (2007))**.** *If the Jacobian of the continuously differential update rule w at a fixed point $\boldsymbol{z}$ has spectral radius less than one, then there exists a neighborhood U around $\boldsymbol{z}$ such that for all $\boldsymbol{x} \in U$, the dynamic converges to $\boldsymbol{z}$.*

Given this, our local convergence theorem states:

Theorem 3.44. *Let$(\boldsymbol{f}^*, \boldsymbol{y}^*)$ be the unique minimax equilibrium of the game $\boldsymbol{A}$. There exists a neighborhood of $(\boldsymbol{f}^*, \boldsymbol{y}^*)$ such that the E-OMWU dynamics converge.*

Proof of Theorem 3.44. The update rule of AMWU can be described as the following dynamical system:

$$
\begin{aligned}
&g(\boldsymbol{f},\boldsymbol{y},\boldsymbol{z},\boldsymbol{w}) := (g_1(\boldsymbol{f},\boldsymbol{y},\boldsymbol{z},\boldsymbol{w}), g_2(\boldsymbol{f},\boldsymbol{y},\boldsymbol{z},\boldsymbol{w}), g_3(\boldsymbol{f},\boldsymbol{y},\boldsymbol{z},\boldsymbol{w}) g_4(\boldsymbol{f},\boldsymbol{y},\boldsymbol{z},\boldsymbol{w}))\\
&g_{1,i}(\boldsymbol{f},\boldsymbol{y},\boldsymbol{z},\boldsymbol{w}) := (g_1(\boldsymbol{f},\boldsymbol{y},\boldsymbol{z},\boldsymbol{w}))_i := \boldsymbol{f}_i \frac{e^{\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{w})}}{\sum_i \boldsymbol{f}_i e^{\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{w})}} \ \forall i \in [n]\\
&g_{2,i}(\boldsymbol{f},\boldsymbol{y},\boldsymbol{z},\boldsymbol{w}) := (g_2(\boldsymbol{f},\boldsymbol{y},\boldsymbol{z},\boldsymbol{w}))_i := \boldsymbol{y}_i \frac{e^{-\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{x} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{z})}}{\sum_i \boldsymbol{y}_i e^{-\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{x} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{z})}} \ \forall i \in [m]\\
&g_3(\boldsymbol{f},\boldsymbol{y},z,w) := \boldsymbol{I}_{n\times n}\boldsymbol{f}\\
&g_4(\boldsymbol{f},\boldsymbol{y},z,w) := \boldsymbol{I}_{m\times m}\boldsymbol{y}.
\end{aligned}
\tag{3.34}
$$

It is easy to show that $(\boldsymbol{f}^*,\boldsymbol{y}^*,\boldsymbol{f}^*,\boldsymbol{y}^*)$ is the stationary point. Following Proposition 3.43 it is sufficient to prove that that the eigenvalue of the Jacobian matrix of g at $(\boldsymbol{f}^*,\boldsymbol{y}^*,\boldsymbol{f}^*,\boldsymbol{y}^*)$ is less than 1.

We now calculate the Jacobian matrix of g at the point $(\boldsymbol{f}^*,\boldsymbol{y}^*,\boldsymbol{f}^*,\boldsymbol{y}^*)$ and show that the spectral radius less than one. We study the Jacobian computed at the stationary point $(\boldsymbol{f}^*,\boldsymbol{y}^*,\boldsymbol{f}^*,\boldsymbol{y}^*)$.

Let v be the value of the game and $\boldsymbol{f}^*,\boldsymbol{y}^*$ is the unique minimax equilibrium (i.e $\boldsymbol{f}^{*\top}\boldsymbol{A}\boldsymbol{y}^* = v$). For $i \notin Supp(\boldsymbol{f}^*)$(e.g. $\boldsymbol{f}_i^* = 0$), we have

$$
\frac{\partial g_{1,i}}{\partial f_i} = \frac{e^{\eta(\boldsymbol{A}\boldsymbol{y}^*)_i}}{\sum_t \boldsymbol{f}_t^* e^{\eta(\boldsymbol{A}\boldsymbol{y}^*)_t}} = \frac{e^{\eta(\boldsymbol{A}\boldsymbol{y}^*)(i)}}{e^{\eta v}}
$$

and other partial derivatives equal to zero. Therefore, $\frac{e^{\eta(\boldsymbol{A}\boldsymbol{y}_i^*)}}{e^{\eta v}} < 1$ is an eigenvalue of the Jacobian computed at the optimal solution(e.g. Due to the uniqueness, $\boldsymbol{A}\boldsymbol{y}_i^* < v$). Similarly, we have for $j \notin Supp(\boldsymbol{y}^*)$, $\frac{\partial \boldsymbol{g}_{2,j}}{\partial \boldsymbol{y}_j} = \frac{e^{-\eta(\boldsymbol{A}^\top \boldsymbol{x}^*)_j}}{e^{-\eta v}} < 1$ is an eigenvalue of the Jacobian matrix. By removing the row and columns corresponding to above eigenvalue, we create a matrix J containing only the elements in the support of $\boldsymbol{f}^*$ and $\boldsymbol{y}^*$. From above, it is clear that the spectral radius of the Jacobian matrix less than 1 iff the spectral of the new matrix J less than 1. Denote D_x, D_y be the diagonal matrix containing non-zero element of $\boldsymbol{f}^*$ and $\boldsymbol{y}^*$ respectively. Let $\boldsymbol{B}$ be the submatrix of of payoff $\boldsymbol{A}$ corresponding to non-zero element of $\boldsymbol{f}^*,\boldsymbol{y}^*$. We then have the matrix J as follow:

$$
A = \begin{bmatrix}
\boldsymbol{I}_{k_1\times k_1} - D_x 1_k 1_k^T & \eta(\alpha+1) D_x(\boldsymbol{B} - v 1_{k_1} 1_{k_2}^\top) & 0_{k1\times k1} & -\eta\alpha D_x(\boldsymbol{B} - v 1_{k_1} 1_{k_2}^T)\\
(\alpha+1)\eta D_y(v 1_{k_2} 1_{k_1}^\top - \boldsymbol{B}^\top) & \boldsymbol{I}_{k_2\times k_2} - D_y 1_{k_2} 1_{k_2}^\top & -\eta\alpha D_y(v 1_{k_2} 1_{k_1}^\top - \boldsymbol{B}^\top) & 0_{k_2\times k_2}\\
\boldsymbol{I}_{k_1\times k_1} & 0_{k_1\times k_2} & 0_{k_1\times k_1} & 0_{k1\times k2}\\
0_{k_2\times k_1} & \boldsymbol{I}_{k_2\times k_2} & 0_{k_2\times k_1} & 0_{k_2\times k_2}
\end{bmatrix}
$$

It is clear that $(1_{k_1}, 0_{k_2}, 0_{k_1}, 0_{k_2}), (0_{k_1}, 1_{k_2}, 0_{k_1}, 0_{k_2})$ are left eigenvectors with eigenvalues zero and thus any right eigenvector $(\boldsymbol{f},\boldsymbol{y},\boldsymbol{z},\boldsymbol{w})$ with nonzero eigenvalue has the property that $\boldsymbol{f}^\top 1_{k_1} = 0$ and $\boldsymbol{y}^\top 1_{k_2} = 0$. Thus, every nonzero eigenvalue of the matrix above is

an eigenvalue of the following matrix:

$$J_{new} = \begin{bmatrix} \boldsymbol{I}_{k_1\times k_1} & \eta(\alpha+1)D_x\boldsymbol{B} & 0_{k1\times k1} & -\eta\alpha D_x\boldsymbol{B} \\ -(\alpha+1)\eta D_y\boldsymbol{B}^\top & \boldsymbol{I}_{k_2\times k_2} & \eta\alpha D_y\boldsymbol{B}^\top & 0_{k_2\times k_2} \\ \boldsymbol{I}_{k_1\times k_1} & 0_{k_1\times k_2} & 0_{k_1\times k_1} & 0_{k1\times k_2} \\ 0_{k_2\times k_1} & \boldsymbol{I}_{k_2\times k_2} & 0_{k_2\times k_1} & 0_{k_2\times k_2} \end{bmatrix}$$

Using the determinant of block matrix we have the characteristic polynomial of the matrix:

$$J_{new} = (-1)^k \det\left(\begin{bmatrix} \lambda(1-\lambda)\boldsymbol{I}_{k_1\times k_1} & \eta(\lambda(\alpha+1)-\alpha)D_x\boldsymbol{B} \\ -\eta(\lambda(\alpha+1)-\alpha)D_y\boldsymbol{B}^\top & \lambda(1-\lambda)\boldsymbol{I}_{k_2\times k_2} \end{bmatrix}\right)$$

This equivalent to

$$(\alpha-(\alpha+1)\lambda)^k q\left(\frac{\lambda(\lambda-1)}{(\alpha+1)\lambda-\alpha}\right),$$

where $q(\lambda)$ is the characteristic polynomial of

$$J_{small} = \left(\begin{bmatrix} 0_{k_1\times k_1} & \eta D_x\boldsymbol{B} \\ -\eta D_y\boldsymbol{B}^\top & 0_{k_2\times k_2} \end{bmatrix}\right)$$

Following Lemma B.6 in Daskalakis and Panageas (2019), we then have J_{small} has eigenvalues of the form $\pm i\eta\tau$ with $\tau \in \mathcal{R}$. Denote $\sigma := \eta\tau$ and thus σ and $\sigma\alpha$ can be sufficiently small in absolute value. We derive that any nonzero eigenvalue λ of the matrix J will satisfy:

$$\begin{aligned} &\frac{\lambda(\lambda-1)}{(\alpha+1)\lambda-\alpha} = i\sigma \\ &\iff \lambda^2-\lambda(1+i\sigma(\alpha+1))+i\sigma\alpha = 0 \\ &\lambda = \frac{1+i\sigma(\alpha+1)\pm\sqrt{1-\sigma^2(\alpha+1)^2-i2\sigma(\alpha-1)}}{2}. \end{aligned}$$

Suppose that $\sqrt{1-\sigma^2(\alpha+1)^2-i2\sigma(\alpha-1)} = x+iy$, then we can derive that in order to maximize the magnitude of λ when σ is relatively small, we have

$$x = \sqrt{\frac{1-\sigma^2(\alpha+1)^2+\sqrt{(1-\sigma^2(\alpha+1)^2)^2+4\sigma^2(\alpha-1)^2}}{2}},\ y = \frac{-\sigma(\alpha-1)}{x}.$$

Thus, the square of magnitude of λ will be:

$$\frac{(1+x)^2+(\sigma(\alpha+1)+y)^2}{4}$$

We note that for sufficiently small σ:

$$\begin{aligned} x &= \sqrt{\frac{1-\sigma^2(\alpha+1)^2+\sqrt{(1+\sigma^2(\alpha+1)^2)^2-16\sigma^2\alpha}}{2}} \\ &\leq \sqrt{\frac{1-\sigma^2(\alpha+1)^2+(1+\sigma^2(\alpha+1)^2)-2\sigma^2\alpha}{2}} \\ &= \sqrt{1-\sigma^2\alpha} \end{aligned}$$

Furthermore, we have

$$\begin{aligned} x &= \sqrt{\frac{1-\sigma^2(\alpha+1)^2+\sqrt{(1+\sigma^2(\alpha+1)^2)^2-16\sigma^2\alpha}}{2}} \\ &\geq \sqrt{\frac{1-\sigma^2(\alpha+1)^2+(1+\sigma^2(\alpha+1)^2)-8\sigma^2\alpha}{2}} \\ &= \sqrt{1-4\sigma^2\alpha}. \end{aligned}$$

Since $\sqrt{1-4\sigma^2\alpha} \leq x \leq 1$ we have

$$\frac{-\sigma(\alpha-1)}{\sqrt{1-4\sigma^2\alpha}} \leq y = \frac{-\sigma(\alpha-1)}{x} \leq -\sigma(\alpha-1).$$

We will prove that:

$$\begin{aligned} &\sigma(\alpha+1)+\frac{-\sigma(\alpha-1)}{\sqrt{1-4\sigma^2\alpha}} \geq 0 \\ &\iff (\alpha+1) \geq \frac{(\alpha-1)}{\sqrt{1-4\sigma^2\alpha}} \\ &\iff (\alpha^2+2\alpha+1)(1-4\sigma^2\alpha) \geq (\alpha-1), \end{aligned}$$

which is true since σ and $\sigma\alpha$ can set sufficiently small. Thus we have

$$0 \leq \sigma(\alpha+1)+y \leq 2\sigma$$

We then have:

$$\begin{aligned} &\frac{(1+x)^2+(\sigma(\alpha+1)+y)^2}{4} \leq \frac{(1+\sqrt{1-4\sigma^2\alpha})^2+(2\sigma)^2}{4} \\ &\leq \frac{2+2\sqrt{1-4\sigma^2\alpha}-4\sigma^2\alpha+4\sigma^2}{4} \leq 1, \end{aligned}$$

Since $\alpha \geq 1$ and the equality happens only when $\sigma = 0$. For $\sigma = 0$, it means that J_{new} has an eigenvalue which is equal to one. Suppose $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}, \hat{\boldsymbol{w}})$ is the corresponding eigenvector. We then have $\boldsymbol{I}\hat{\boldsymbol{x}} - \boldsymbol{I}\hat{\boldsymbol{z}} = 0$ and $\boldsymbol{I}\hat{\boldsymbol{y}} - \boldsymbol{I}\hat{\boldsymbol{w}} = 0$, thus we derive that: $\hat{\boldsymbol{x}} = \hat{\boldsymbol{z}}$ and $\hat{\boldsymbol{y}} = \hat{\boldsymbol{w}}$. Furthermore, we also have: $D_x\boldsymbol{B}\hat{\boldsymbol{x}} = 0$ and $D_y\boldsymbol{B}^\top\hat{\boldsymbol{y}} = 0$, thus we have $\boldsymbol{B}\hat{\boldsymbol{x}} = 0$ and $\boldsymbol{B}^\top\hat{\boldsymbol{y}} = 0$. From previous argument, we also have: $\hat{\boldsymbol{x}}^\top 1_{k_1} = 0$ and $\hat{\boldsymbol{y}}^\top 1_{k_2} = 0$. Thus, the strategy $(\boldsymbol{x}^*, \boldsymbol{y}^*) + t(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ also an optimal strategy for small enough t to make every

element non-negative. Since the assumption of uniqueness, we then have $\hat{\boldsymbol{x}} = 0, \hat{\boldsymbol{y}} = 0$, contradiction. Thus, every eigenvalue of matrix J has magnitude of less than 1. The proof is complete. □

Derivatives calculation

Set $S_{\boldsymbol{x}} = \sum_i f_i e^{\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}w)}$ and $S_{\boldsymbol{y}} = \sum_i \boldsymbol{y}_i e^{-\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{x} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{z})}$. The derivative at $(\boldsymbol{f}^*, \boldsymbol{y}^*, \boldsymbol{f}^*, \boldsymbol{y}^*)$ is as follow:

$$\frac{\partial g_{1,i}}{\partial f_i} = \frac{e^{\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}w)}}{S_{\boldsymbol{x}}} - f_i \frac{e^{2\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}w)}}{S_{\boldsymbol{x}}{}^2} \forall i \in [n],$$

$$\frac{\partial g_{1,i}}{\partial \boldsymbol{x}_j} = f_i e^{\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}w)} \frac{-e^{\eta((\alpha+1)\boldsymbol{e}_j{}^\top \boldsymbol{A}\boldsymbol{y} - \alpha \boldsymbol{e}_j{}^\top \boldsymbol{A}w)}}{S_{\boldsymbol{x}}{}^2} \forall i \in [n], j \in [m], j \neq i,$$

$$\frac{\partial g_{1,i}}{\partial \boldsymbol{y}_j} = f_i e^{\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}w)} \frac{\eta(\alpha+1)\boldsymbol{A}_{i,j} S_{\boldsymbol{x}} - \eta(\alpha+1)\sum_t \boldsymbol{A}_{tj}\boldsymbol{x}_t e^{\eta((\alpha+1)e_t{}^\top \boldsymbol{A}\boldsymbol{y} - \alpha e_t{}^\top \boldsymbol{A}w)}}{S_{\boldsymbol{x}}^2} \forall i \in [n], j = i,$$

$$\frac{\partial g_{1,i}}{\partial \boldsymbol{z}_j} = 0 \ \ \forall i, j \in [n],$$

$$\frac{\partial g_{1,i}}{\partial \boldsymbol{w}_j} = f_i e^{\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{y} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}w)} \frac{-\alpha\eta \boldsymbol{A}_{ij} S_{\boldsymbol{x}} + \eta\alpha \sum_t \boldsymbol{A}_{tj}\boldsymbol{x}_t e^{\eta((\alpha+1)e_t{}^\top \boldsymbol{A}\boldsymbol{y} - \alpha e_t{}^\top \boldsymbol{A}w)}}{S_{\boldsymbol{x}}^2} \forall i \in [n], j \in [m].$$

$$\frac{\partial g_{2,i}}{\partial \boldsymbol{y}_i} = \frac{e^{-\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{x} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{z})}}{S_{\boldsymbol{y}}} - \boldsymbol{y}_i \frac{e^{-2\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{x} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{z})}}{S_{\boldsymbol{y}}{}^2} \forall i \in [m],$$

$$\frac{\partial g_{2,i}}{\partial \boldsymbol{y}_j} = \boldsymbol{y}_i e^{-\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{x} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{z})} \frac{-e^{-\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{x} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{z})}}{S_{\boldsymbol{y}}{}^2} \forall i \in [n], j \in [m], j \neq i,$$

$$\frac{\partial g_{2,i}}{\partial \boldsymbol{x}_j} = \boldsymbol{y}_i e^{-\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{x} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{z})} \frac{-\eta(\alpha+1)\boldsymbol{A}_{i,j} S_{\boldsymbol{y}} + \eta(\alpha+1)\sum_t \boldsymbol{A}_{tj}\boldsymbol{y}_t e^{-\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{x} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{z})}}{S_{\boldsymbol{y}}^2} \forall i \in [m], j \in [n],$$

$$\frac{\partial g_{2,i}}{\partial \boldsymbol{z}_j} = \boldsymbol{y}_i e^{-\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{x} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{z})} \frac{\eta\alpha \boldsymbol{A}_{i,j} S_{\boldsymbol{y}} - \eta\alpha \sum_t \boldsymbol{A}_{tj}\boldsymbol{x}_t e^{-\eta((\alpha+1)\boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{x} - \alpha \boldsymbol{e}_i{}^\top \boldsymbol{A}\boldsymbol{z})}}{S_{\boldsymbol{y}}^2} \forall i \in [m], j \in [n],$$

$$\frac{\partial g_{2,i}}{\partial \boldsymbol{w}_j} = 0 \ \ \forall i, j \in [m],$$

$$\frac{\partial g_{3,i}}{\partial f_i} = 1 \ \ \forall i \in [n], \ 0 \text{ otherwise},$$

$$\frac{\partial g_{4,i}}{\partial \boldsymbol{y}_i} = 1 \ \ \forall i \in [m], \ 0 \text{ otherwise},$$

3.11 Appendix B: Additional Experimental Results

3.11.1 Oblivious adversary

We specify our experiment setting as follow. In a chosen random matrix game, we first let the agent follows a fixed MWU against the adversary follows MWU with a chosen learning rate in the set: $[0.5, 0.45, 0.4, \ldots, 0.05]$ [13]. Then, we record the strategies of the adversary in each round and consider it as the oblivious adversary. To highlight the difference between AMWU and OMWU, we also test the performance of OWMU with learning rate $\eta = 1$. For the random games, we test it on 5 random seeds for each matrix size. For the meta games, we run our algorithms against 5 different oblivious

[13] Each learning rate will create different oblivious adversary.

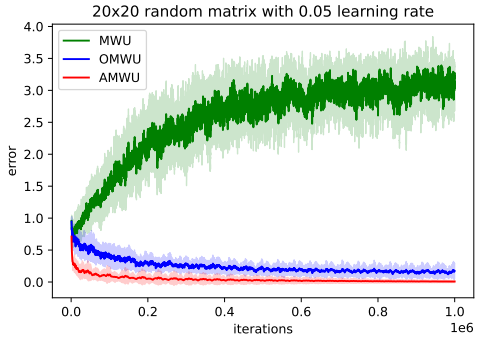
adversary (i.e., MWU with the learning rate in $[0.5, 0.4, 0.3, 0.2, 0.1]$) and report the average performance as well as the standard deviation.

Average performance against oblivious adversary: we report performance of AMWU and other baselines against different oblivious adversaries, i.e., the MWU adversary with different learning rate $[0.5, 0.45, 0.4. \ldots, 0.05]$. As we can see in Figure 3.9 and Figure 3.10, AMWU outperfoms other baselines by a large margin across all the adversary setting in random matrix games. A similar trend can be observed in the Connect Four and Disc experiments in Figure 3.8.

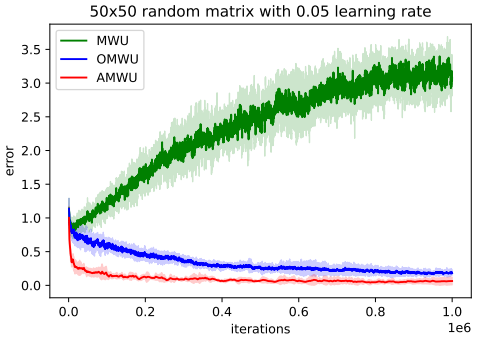
3.11.2 Last round convergence of AMWU

For a fair comparison, we set up a common learning rate for our algorithm AMWU and the baselines MWU and OMWU. In the experiments of average performance, we first set the common learning rate $\eta = 0.01$ and the exploiting rate $\alpha = 100$. In order to highlight the difference between AMWU and OMWU, we also test the performance of OMWU with learning rate $\eta = 1$. That is, the OWMU with the same relative weight between the predictable sequence $\boldsymbol{x}_{t-1}$ and the regularizer $R(\boldsymbol{f})$ as AMWU (i.e., $\eta_{OMWU} = \eta_{AMWU} \times \alpha_{AMWU}$). In the experiments of last round convergence, we vary the common learning rate η (i.e., $\eta = [0.01, 0.025, 0.05]$) to see whether the convergence trend we see is robust against the learning rate. In here we focus on the random matrix games (20×20 and 50×50 dimensions) due to its nice property of unique Nash Equilibrium, which AMWU and OMWU require to convergence. Since there is no guarantee of convergence of OMWU with a large learning rate (e.g., $\eta = 1$), we do not consider $OMWU_1$ as a baseline in this experiment.

Last round convergence in self-play: we report the performance of AMWU and other baselines in self-play setting. As we can see in Figure 3.11, Figure 3.12 and Figure 3.6, AMWU outperforms OWMU and MWU by a large margin across all the 3 different learning rate setting. The MWU shows divergence in last round convergence in as expected in Bailey and Piliouras (2018). A similar trend can be observed in the Connect Four and Disc experiments in Figure 3.7.

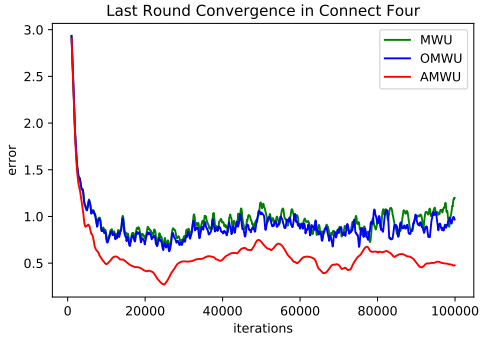


(A) 20×20 random games

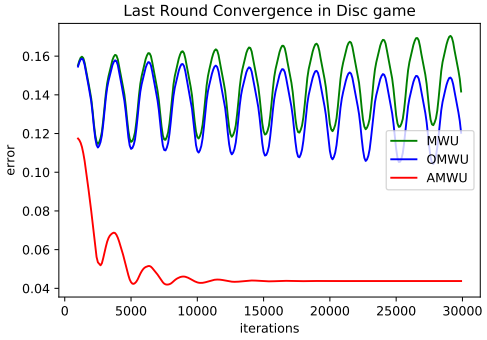


(B) 50×50 random games

FIGURE 3.6: Last round convergence in random games with 0.05 learning rate

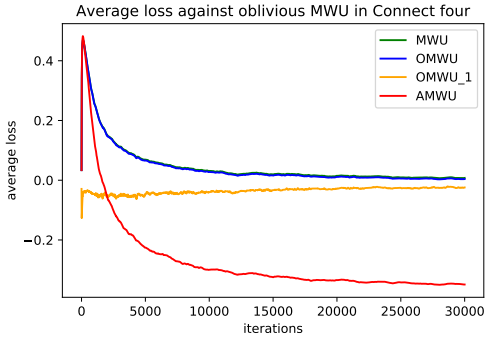


(A) Connect Four meta game

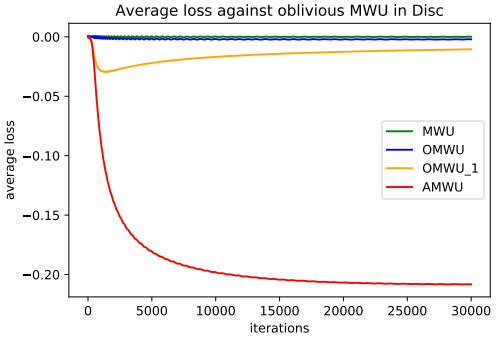


(B) Disc meta game

FIGURE 3.7: Last round convergence in meta games

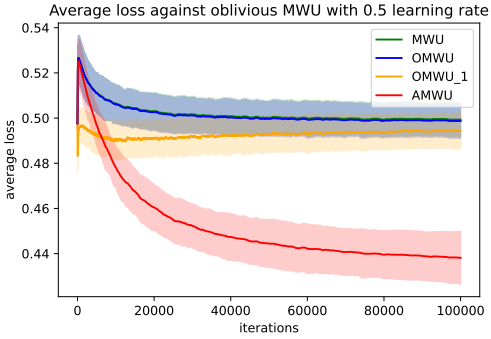


(A) Against MWU adversary in Connect Four

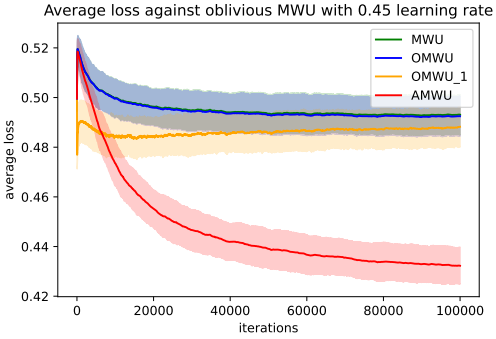


(B) Against MWU adversary in Disc

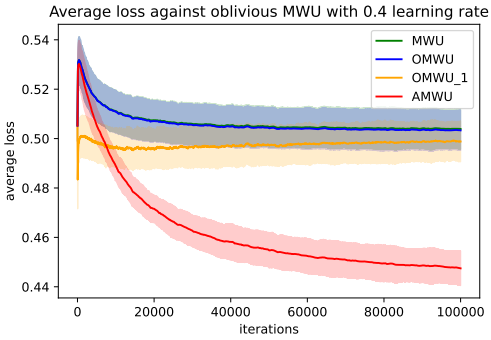
FIGURE 3.8: Against Oblivious MWU adversary in meta games



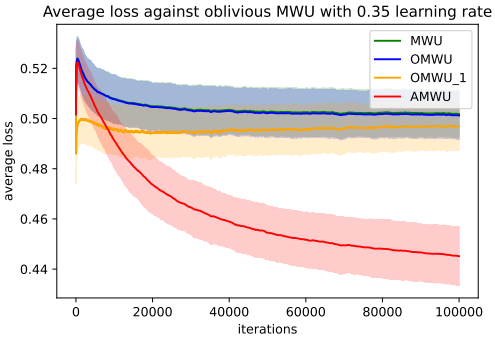
(A) 0.5 learning rate MWU adversary



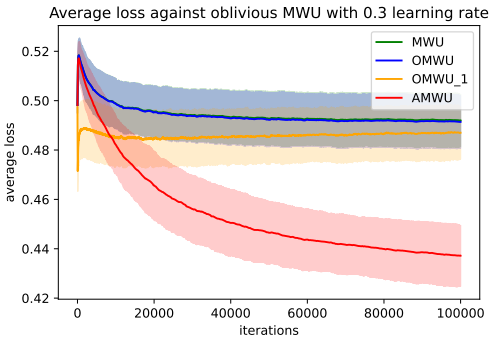
(B) 0.45 learning rate MWU adversary



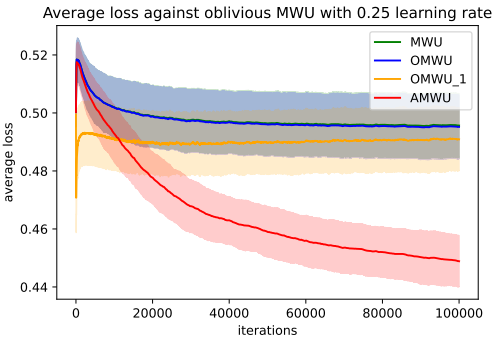
(C) 0.4 learning rate MWU adversary



(D) 0.35 learning rate MWU adversary

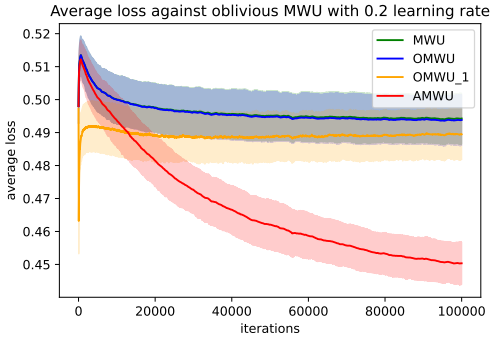


(E) 0.3 learning rate MWU adversary

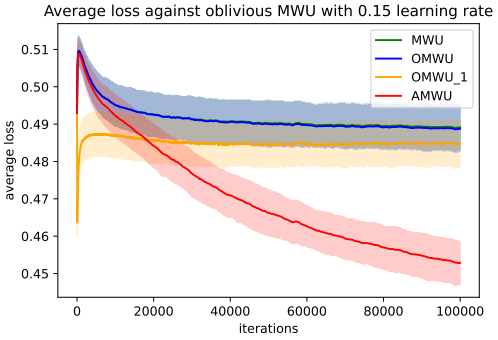


(F) 0.25 learning rate MWU adversary

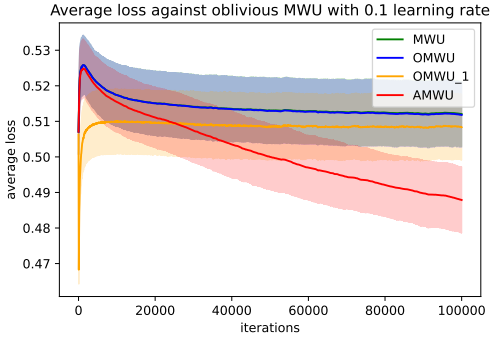
FIGURE 3.9: Against different Oblivious MWU adversary in random games



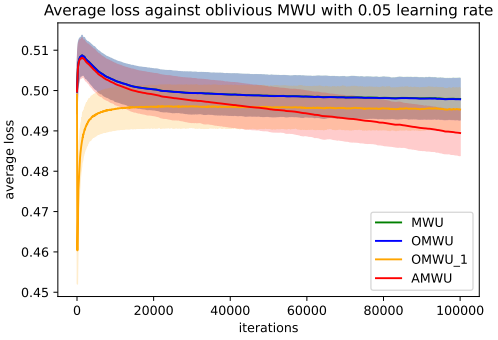
(A) 0.2 learning rate MWU adversary



(B) 0.15 learning rate MWU adversary

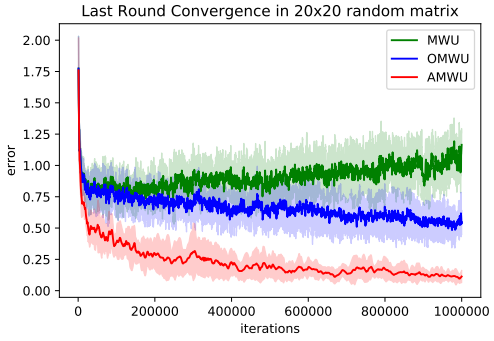


(C) 0.1 learning rate MWU adversary

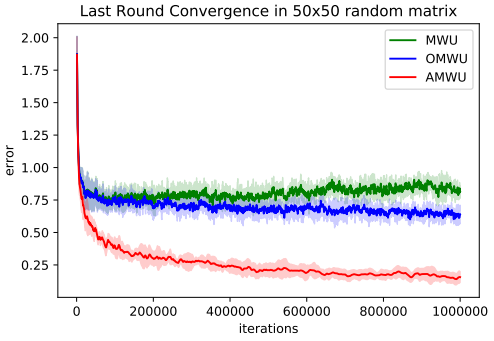


(D) 0.05 learning rate MWU adversary

FIGURE 3.10: Against different Oblivious MWU adversary in random games

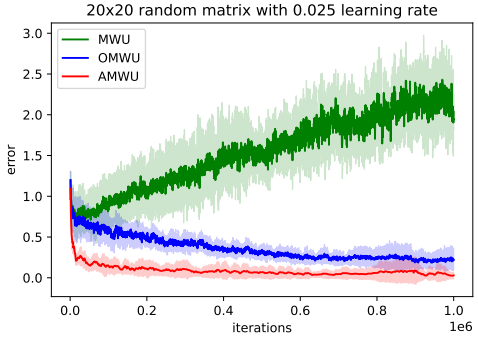


(A) 20×20 random games

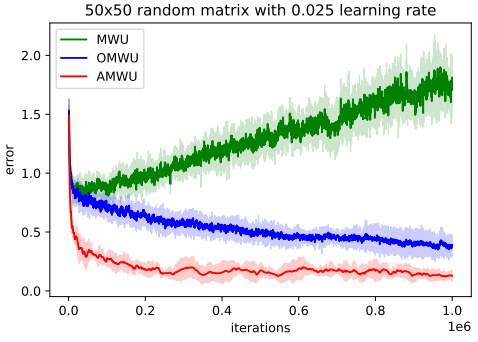


(B) 50×50 random games

FIGURE 3.11: Last round convergence in random games with 0.01 learning rate



(A) 20×20 random games



(B) 50×50 random games

FIGURE 3.12: Last round convergence in random games with 0.025 learning rate

Chapter 4

Online Markov Decision Processes Against Strategic Adversary

In this chapter, we address the challenge of online learning in the presence of a strategic adversary under the Online Markov Decision Processes (OMDPs) framework. Our study begins by demonstrating that the existing algorithm, MDP-Expert (Even-Dar et al., 2009), which works well with oblivious adversaries, can still apply to our strategic adversary setting and achieve a policy regret bound of $\mathcal{O}(\sqrt{T\log(L)} + \tau^2\sqrt{T\log(|A|)})$, where L is the size of the adversary's pure strategy set and $|A|$ denotes the size of the agent's action space. To address real-world games where the support size of a Nash equilibrium (NE) is small, we propose a novel algorithm, *MDP-Online Oracle Expert* (MDP-OOE), that achieves a policy regret bound of $\mathcal{O}(\sqrt{T\log(L)} + \tau^2\sqrt{Tk\log(k)})$, where k depends only on the support size of the NE. MDP-OOE leverages the key benefit of Double Oracle in game theory and can effectively solve games with prohibitively large action space. Lastly, to gain a better understanding of the learning dynamics of no-regret methods, we introduce Last Round Convergence in OMDPs algorithm (LRC-OMDP) that achieves last round convergence to a NE against the strategic adversary. To our best knowledge, this is the first work leading to the last iteration result in OMDPs.

4.1 Introduction

Reinforcement Learning (RL) (Sutton and Barto, 2018) provides a general solution framework for optimal decision making under uncertainty, where the agent aims to minimise its cumulative loss while interacting with the environment. While RL algorithms have shown empirical and theoretical successes in stationary environments, it is an open challenge to deal with non-stationary environments in which the loss function and/or the transition dynamics change over time (Laurent et al., 2011). In tackling non-stationary environments, we are interested in designing learning algorithms that can achieve a no-regret guarantee (Even-Dar et al., 2009; Dick et al., 2014), where the regret is defined as the difference between the accumulated total loss and the total loss of the best fixed stationary policy in hindsight.

There are online learning algorithms that can achieve no-external regret property with changing loss function (but not changing transition dynamics), either in the full information (Even-Dar et al., 2009; Dick et al., 2014) or the bandit (Neu et al., 2010; Neu and Olkhovskaya, 2021) settings. However, most existing solutions are established based on the key assumption that the adversary is *oblivious*, meaning the changes in loss functions do not depend on the historical trajectories of the agent. This crucial assumption limits the applicability of no-regret algorithms to many RL fields, particularly multi-agent reinforcement learning (MARL) (Yang and Wang, 2020). In a multi-agent system, since all agents are learning simultaneously, one agent's adaption on its strategy will make the environment *non-oblivious* from other agents' perspective. Therefore, to find the optimal strategy for each player, one must consider the strategic reactions of others rather than regard them as purely oblivious. As such, studying no-regret algorithms against a non-oblivious adversary is a pivotal step in adapting existing online learning techniques into MARL settings.

Another challenge in online learning is the non-convergence dynamics in a system. When agents apply no-regret algorithms such as Multiplicative Weights Update (MWU) (Freund and Schapire, 1999) or Follow the Regularized Leader (FTRL) (Shalev-Shwartz, 2012) to play against each other, the system demonstrates behaviours that are Poincaré recurrent (Mertikopoulos et al., 2018), meaning the last round convergence can never be achieved (Bailey and Piliouras, 2018). Recent works (Dinh et al., 2021; Mertikopoulos et al., 2019) have focused on different learning dynamics in normal-form games that can lead to last round convergence to a Nash equilibrium (NE) while maintaining the no-regret property. Yet, when it comes to OMDPs, it still remains an open challenge on how the no-regret property and the last round convergence can be both achieved, especially against the strategic adversary. The focus of OMDPs is often on regret bound analysis against oblivious adversary (Even-Dar et al., 2009; Neu et al., 2010; Dick et al., 2014), in which last round convergence property is impossible to achieve due to the adversary's fixed behaviour. When a non-oblivious adversary is considered, the focus is

on finding stationary points of the system (Leslie et al., 2020; Guan et al., 2016) rather than analysing the dynamic leading to the last round convergence to a NE.

Markov decision processes (MDPs) provide a popular tool to formulate stochastic optimization problems (Sutton and Barto, 2018), yet it is often that only a relaxation of real models can satisfy the Markovian assumption. In situations where the reward function can change over time and thus the Markovian assumption is not satisfied, OMDPs offer a general solution by applying existing experts' algorithms to more adversarial MDPs (Even-Dar et al., 2009). OMDPs algorithms provide the agent with a performance guarantee under the assumption that the adversary is oblivious (Neu et al., 2013; Dick et al., 2014), thus limiting its application in settings where the adversary is also a learning agent.

In this chapter, we relax the assumption of the oblivious adversary in OMDPs and study a new setting where the loss function is chosen by a strategic agent that follows a no-external regret algorithm. This setting can be used in applications within economics to model systems and firms (Filar and Vrieze, 1997), for example, an oligopoly with a dominant player, or ongoing interactions between industry players and authority (e.g., a government that acts as an order-setting body). Another motivating example is the stochastic inventory control problem (Puterman, 1990). In each period, based on the current inventory, the store manager needs to decide the number of items to order from the supplier. The manager faces the dilemma: having too many items will increase the inventory cost while running out of items will lead to revenue loss. Since both the item price and the inventory cost can change over time, the problem can be considered as an OMDP. Furthermore, the supplier can decide the item price based on the total demand of the item as well as its capacity to maximise its profit, thus making it a non-oblivious strategic adversary.

Under this setting, we study how the agent can achieve different goals such as no-policy regret and last round convergence.

Our contributions in this chapter are at three folds:

- We prove that the well-known MDP-Expert (MDP-E) algorithm (Even-Dar et al., 2009) can apply to achieve a policy regret bound of $\mathcal{O}(\sqrt{T\log(L)}+\tau^2\sqrt{T\log(|A|)})$, and the average strategies of the agents will converge to a NE of the game.
- For many real-world applications where the support size of NE is small (McMahan et al., 2003; Dinh et al., 2022), we introduce an efficient no-regret algorithm, *MDP-Online Oracle Expert (MDP-OOE)*, which achieves the policy regret bound of $\mathcal{O}(\tau^2\sqrt{Tk\log(k)}+\sqrt{T\log(L)})$ against the non-oblivious strategic adversary, where k depends on the support size of the NE. MDP-OOE inherits the key benefits of both Double Oracle (McMahan et al., 2003) and MDP-E (Even-Dar et al., 2009); it can solve games with large action space while maintaining the no-regret property.

- To achieve last round convergence guarantee for no-external regret algorithms, we introduce the algorithm of *Last Round Convergence in OMDPs (LRC-OMDP)* such that in cases where the adversary follows a no-external regret algorithm, the dynamics will lead to the last round convergence to a NE. To the best of our knowledge, this is the first last-iteration convergence result in OMDPs.

4.2 Related Work

The setting of OMDPs with the strategic adversary, though novel, shares certain aspects in common with existing literature in online learning and stochastic game domains. Here we review each of these research branches.

Many researchers have considered OMDPs with an oblivious environment, where the loss function can be set arbitrarily. The performance of the algorithm is measured by external regret: the difference between the total loss and the best stationary policy in hindsight. In this setting with stationary transition dynamics, MDP-E (Even-Dar et al., 2009) proved that if the agent bounds the "local" regret in each state, then the "global" regret will be bounded. Neu et al. (2010, 2013) considered the same problem with the bandit reward feedback and provided no-external regret algorithms in this setting. Dick et al. (2014) studied a new approach for OMDPs where the problem can be transformed into an online linear optimization form, from which no-external regret algorithms can be derived. Cheung et al. (2019) proposed a no-external regret algorithm in the case of non-stationary transition distribution, given that the variation of the loss and transition distributions do not exceed certain variation budgets.

In a non-oblivious environment, Yu et al. (2009) provided an example demonstrating that no algorithms can guarantee sublinear external regret against a non-oblivious adversary. Thus, in OMDPs with non-oblivious opponents (e.g., agents using adaptive algorithms), the focus is often on finding stationary points of the system rather than finding a no-external regret algorithm (Leslie et al., 2020). In this chapter, we study cases where the adversary follows an adaptive no-regret algorithm, and tackle the hardness result of non-oblivious environments in OMDPs.

The problem of the non-oblivious adversary has also been studied in the multi-armed bandit setting, a special case of OMDPs. In this setting, Arora et al. (2012a) considered m-memory bounded adversary and provided an algorithm with a policy regret bound that depends linearly on m, where the policy regret includes the adversary's adaptive behaviour (i.e., see Equation (4.1)). Compared to their work, our study considers strategic adversary which turns out to be ∞-memory bounded adversary. Thus the algorithm suggested in Arora et al. (2012a) can not be applied. Recently, Dinh et al. (2021) studied the same strategic adversary in full information normal-form setting and provided an algorithm that leads to last round convergence. However, both of the above works

only studied the simplified version of OMDPs, thus they do not capture the complexity of the problem. We argue that since strategic adversary setting has many applications due to the popularity of no-regret algorithms (Cesa-Bianchi and Lugosi, 2006; Zinkevich et al., 2007; Daskalakis et al., 2018), it is important to study no-regret methods in more practical settings such as OMDPs.

Stochastic games (SGs) (Shapley, 1953; Deng et al., 2021) offer a multi-player game framework where agents jointly decide the loss and the state transition. Compared to OMDPs, the main difference is that SGs allow each player to have a representation of states, actions and rewards, thus players can learn the representations over time and find the NE of the stochastic games (Wei et al., 2017; Tian et al., 2021). The performance in SGs is often measured by the difference between the average loss and the value of the game (i.e., the value when both players play a NE), which is a weaker notion of regret compared to the best fixed policy in hindsight in OMDPs. Intuitively, the player can learn the structure of the game (i.e., transition model, reward function) over time, thus on average, the player can calculate and compete with the value of the game. In non-episodic settings, the Upper Confidence Stochastic Game algorithm (UCSG) (Wei et al., 2017) guarantees the regret of $\text{Reg}_T = \tilde{\mathcal{O}}(D^3|S|^5|A|+D|S|\sqrt{|A|T})$ with high probability, given that the opponent's action is observable. However, to compete with the best stationary policy, knowing the game structure does not guarantee a good performance (i.e., the performance will heavily depend on the strategic behaviour of opponents). Tian et al. (2021) proved that in the SG setting, achieving no regret with respect to the best stationary policy in hindsight is statistically hard. Our settings can be considered as a sub-class of SGs where only the agent controls the transition model (i.e., single controller SGs), based on this, we try to overcome the above challenge.

We summarise the difference between our setting and OMDPs and SGs in Figure 4.1. Compared to OMDPs, we relax the assumption about the oblivious environment and study a non-oblivious counterpart with a strategic adversary. Compared to SGs, we relax the assumption of knowing opponent's action in a non-episodic setting and our results only require observation of the loss functions. Furthermore, the performance measurement is with respect to the best stationary policy in hindsight, which is proven

FIGURE 4.1: The scope of our contribution in this chapter.

	Non-oblivious adversary within a two-player game framework	Oblivious adversary in Markov Decision Processes
Regret *w.r.t* best policy in hindsight	MDP-OOE (our contribution) $\mathcal{O}(\tau^2\sqrt{Tk\log(k)}+\sqrt{T\log(L)})$	OMDPs: (MDP-E) (Even-Dar et al., 2009) $\text{Reg}_T = \mathcal{O}(\tau^2\sqrt{T\log(\|A\|)})$
Regret *w.r.t.* value of the game	SGs:(UCSG) (Wei et al., 2017) $\text{Reg}_T = \tilde{\mathcal{O}}(D^3\|S\|^5\|A\| + D\|S\|\sqrt{\|A\|T})$	OMDPs

to be statistically hard in SGs (Tian et al., 2021). Intuitively, since we consider the problem of single controller SGs, it can overcome the hardness result. Guan et al. (2016) studied a similar setting to our work, where only one player affects the transition kernel of the game. By viewing the game as an online linear optimisation, it can derive the minimax equilibrium of the game. There are two main challenges of the algorithm. Firstly, it requires both players to pre-calculate the minimax equilibrium of the game and fixes to this strategy during the repeated game. Thus, in the situation where the adversary is an independent agent (i.e., it follows a different learning dynamic), the proposed algorithm can not be applied. Secondly, and most importantly, the no regret analysis is not provided for the algorithm in Guan et al. (2016), thus the algorithm can not be applied in an adversary environment. We fully address both challenges in this work.

4.3 Problem Formulations & Preliminaries

We consider OMDPs where at each round $t \in \mathbb{N}$, an adversary can choose the loss function $\boldsymbol{l}_t$ based on the agent's policy history $\{\pi_1, \pi_2, \ldots, \pi_{t-1}\}$. Formally, we have OMDPs with finite state space S; finite action set for the agent at each state A; and a fixed transition model P. The agent's starting state, x_1, is distributed according to some distribution μ_0 over S. At time t, given state $x_t \in S$, the agent chooses an action $a_t \in A$, then the agent moves to a new random state x_{t+1} which is determined by the fixed transition model $P(x_{t+1}|x_t, a_t)$. Simultaneously, the agent receives an immediate loss $\boldsymbol{l}_t(x_t, a_t)$, in which the loss function $\boldsymbol{l}_t : S \times A \to R$ is bounded in $[0,1]^{|A|\times|S|}$ and is chosen by the adversary from a simplex $\Delta_L := \{\boldsymbol{l} \in \mathbb{R}^{|A|\times|S|} | \boldsymbol{l} = \sum_{i=1}^{L} x_i \boldsymbol{l}_i,\ \sum_{i=1}^{L} x_i = 1,\ x_i \geq 0\ \forall i\}$ where $\{\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_L\}$ are the loss vectors of the adversary. We assume a zero-sum game setting where the adversary receives the loss of $-\boldsymbol{l}_t(x_t, a_t)$ at round t and consider popular full information feedback (Even-Dar et al., 2009; Dick et al., 2014), meaning the agent can observe the loss function $\boldsymbol{l}_t$ after each round t.

Against a strategic adversary, the formal definition of no-external regret becomes inadequate since the adversary is allowed to adapt to the agent's action. In this chapter, we adopt the same approach in Arora et al. (2012a) and consider policy regret. Formally, the goal of the agent is to have minimum policy regret with respect to the best fixed policy in hindsight:

$$R_T(\pi) = \mathbb{E}_{X,A}\left[\sum_{t=1}^{T} \boldsymbol{l}_t^{\pi_t}(X_t, A_t)\right] - \mathbb{E}_{X,A}\left[\sum_{t=1}^{T} \boldsymbol{l}_t^{\pi}(X_t^{\pi}, A_t^{\pi})\right], \tag{4.1}$$

where $\boldsymbol{l}_t^{\pi_t}$ denotes the loss function at time t while the agent follows $\pi_1, \ldots, \pi_T$ and $\boldsymbol{l}_t^{\pi}$ is the adaptive loss function against the fixed policy π of the agent. We say that the agent achieves sublinear policy regret (i.e., no-policy regret property) with respect to the best

fixed strategy in hindsight if $R_T(\pi)$ satisfies:

$$\lim_{T\to\infty} \max_{\pi} \frac{R_T(\pi)}{T} = 0.$$

In a general non-oblivious adversary, we prove by a counter-example that it is impossible to achieve an algorithm with a sublinear policy regret [1]. Suppose the agent faces an adversary such that it gives a very low loss for the agent if the action in the first round of the agent is a specific action (i.e., by fixing the loss function to $\mathbf{0}$), otherwise the adversary will give a high loss (i.e., by fixing the loss function to $\mathbf{1}$). Against this type of adversary, without knowing the specific action, the agent's policy regret in Equation (4.1) will be $\mathcal{O}(T)$. Thus, in the general non-oblivious adversary case, we will have a hardness result in policy regret. To resolve the hardness result, we study the strategic adversary in OMDPs.

Assumption 2 (Strategic Adversary). The adversary flows a no-external regret algorithm such as for any sequence of π_t:

$$\lim_{T\to\infty} \max_{\boldsymbol{l}} \frac{R_T(\boldsymbol{l})}{T} = 0, \text{ where } R_T(\boldsymbol{l}) = \mathbb{E}_{X,A}\left[\sum_{t=1}^{T} \boldsymbol{l}(X_t, A_t)\right] - \mathbb{E}_{X,A}\left[\sum_{t=1}^{T} \boldsymbol{l}_t^{\pi_t}(X_t, A_t)\right].$$

The rationale of Assumption 2 comes from the vanilla property of no-external algorithms: without prior information, the adversary will not do worse than the best-fixed strategy in hindsight (Dinh et al., 2021). Thus, without the priority knowledge about the agent, the adversary will have the incentive to follow a no-external regret algorithm. In the same way as the full information feedback assumption for the agent, we assume that after each round t, the adversary observes the agent' stationary policy distribution $\boldsymbol{d}_{\pi_t}$.

For every policy π, we define $P(\pi)$ the state transition matrix induced by π such that $P(\pi)_{s,s'} = \sum_{a\in A} \pi(a|s) P^a_{s,s'}$. We assume through the chapter that we have the mixing time assumption, which is a common assumption in OMDPs (Even-Dar et al., 2009; Dick et al., 2014; Neu et al., 2013):

Assumption 3 (Mixing time). There exists a constant $\tau > 0$ such that for all distributions $\boldsymbol{d}$ and $\boldsymbol{d}'$ over the state space and for any policy π,

$$\left\|\boldsymbol{d}P(\pi) - \boldsymbol{d}'P(\pi)\right\|_1 \leq e^{-1/\tau}\left\|\boldsymbol{d} - \boldsymbol{d}'\right\|_1,$$

where $\|\boldsymbol{x}\|_1$ denotes the l_1 norm of a vector $\boldsymbol{x}$.

Denote $\boldsymbol{v}_t^{\pi}(x, a)$ the probability of (state, action) pair (x, a) at time step t by following policy π with initial state x_1. Following Assumption 3, for any initial states, $\boldsymbol{v}_t^{\pi}$ will converge to a stationary distribution $\boldsymbol{d}_{\pi}$ as t goes to infinity. Denote $\boldsymbol{d}_{\Pi}$ the stationary

[1] In the multi-armed bandit setting, it is also impossible to achieve sublinear policy regret against all adaptive adversaries (see Theorem 1 in Arora et al. (2012a)).

distribution set from all agent's deterministic policies. With a slight abuse of notation, when an agent follows an algorithm A with policies $\pi_1, \pi_2, \ldots$ at each time step, we denote $\boldsymbol{v}_t(x,a) = \mathbb{P}\left[X_t = x, A_t = a\right]$, $\boldsymbol{d}_t = \boldsymbol{d}_{\pi_t}$. Thus, the regret in Equation (4.1) can be expressed as

$$R_T(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} \left\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{v}_t \right\rangle\right] - \mathbb{E}\left[\sum_{t=1}^{T} \left\langle \boldsymbol{l}_t^{\pi}, \boldsymbol{v}_t^{\pi} \right\rangle\right].$$

Assumption 3 allows us to define the average loss of policy π in an online MDP with loss $\boldsymbol{l}$ as $\eta_{\boldsymbol{l}}(\pi) = \langle \boldsymbol{l}, \boldsymbol{d}_\pi \rangle$ and the accumulated loss $Q_{\pi,\boldsymbol{l}}(s,a)$ is defined as

$$Q_{\pi,\boldsymbol{l}}(s,a) = \mathbb{E}\left[\sum_{t=1}^{\infty} \left(\boldsymbol{l}(s_t,a_t) - \eta_{\boldsymbol{l}}(\pi)\right) \middle| s_1 = s, a_1 = a, \pi\right].$$

As the dynamic between the agent and adversary is zero-sum, we can apply the minimax theorem (Neumann, 1928):

$$\min_{\boldsymbol{d}_\pi \in \Delta_{\boldsymbol{d}_\Pi}} \max_{\boldsymbol{l} \in \Delta_L} \langle \boldsymbol{l}, \boldsymbol{d}_\pi \rangle = \max_{\boldsymbol{l} \in \Delta_L} \min_{\boldsymbol{d}_\pi \in \Delta_{\boldsymbol{d}_\Pi}} \langle \boldsymbol{l}, \boldsymbol{d}_\pi \rangle = v. \tag{4.2}$$

The saddle point $(\boldsymbol{l}, \boldsymbol{d}_\pi)$ that satisfies Equation (4.2) is the NE of the game (Nash Jr, 1950) and v is the called the value of the game. Our work is based on no-external regret algorithms in the normal-form game such as Multiplicative Weights Update (Freund and Schapire, 1999), which is described as

Definition 4.1 (Multiplicative Weights Update)**.** Let $\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots$ be a sequence of feedback received by the agent. The agent is said to follow the MWU if strategy $\tilde{\boldsymbol{\pi}}_{t+1}$ is updated as follows

$$\tilde{\boldsymbol{\pi}}_{t+1}(i) = \tilde{\boldsymbol{\pi}}_t(i) \frac{\exp(-\mu_t \boldsymbol{k}_t(\boldsymbol{a}^i))}{\sum_{i=1}^{n} \tilde{\boldsymbol{\pi}}_t(i) \exp(-\mu_t \boldsymbol{k}_t(\boldsymbol{a}^i))}, \forall i \in [n], \tag{4.3}$$

where $\mu_t > 0$ is a parameter, n is the number of pure strategies (i.e., experts) and $\tilde{\boldsymbol{\pi}}_0 = [1/n, \ldots, 1/n]$.

We also consider ϵ-Nash equilibrium of the game:

Definition 4.2 (ϵ-Nash equilibrium)**.** Assume $\epsilon > 0$. We call a point $(\boldsymbol{l}, \boldsymbol{d}_\pi) \in \Delta_L \times \Delta_{\boldsymbol{d}_\Pi}$ ϵ-NE if:

$$\max_{\boldsymbol{l} \in \Delta_L} \langle \boldsymbol{l}, \boldsymbol{d}_\pi \rangle - \epsilon \leq \langle \boldsymbol{l}, \boldsymbol{d}_\pi \rangle \leq \min_{\boldsymbol{d}_\pi \in \Delta_{\boldsymbol{d}_\Pi}} \langle \boldsymbol{l}, \boldsymbol{d}_\pi \rangle + \epsilon.$$

Under the setting of OMDPs against the strategic adversary who aims to minimise the external regret (i.e., Assumption 2), we study several properties that the agent can achieve such as no-policy regret and last round convergence.

4.4 MDP-Expert against Strategic Adversary

Algorithm 19 MDP-Expert (MDP-E)

1: **Input:** Expert algorithm B_s (i.e., MWU) for each state
2: **for** $t = 1$ to ∞ **do**
3: Use algorithm B_s with expert set A and the feedback $Q_{\pi_t, \boldsymbol{l}_t}(s, .)$ for each state s
4: Output π_{t+1} and observe $\boldsymbol{l}_{t+1}$
5: **end for**

When the agent plays against a non-oblivious opponent, one challenge is that the best fixed policy π is not based on the current loss sequence $[\boldsymbol{l}_1, \boldsymbol{l}_2, \dots]$ of the agent but a different loss sequence $[\boldsymbol{l}_1^\pi, \boldsymbol{l}_2^\pi \dots]$ induced by the policy π. Thus, to measure the regret in the case of a non-oblivious opponent, we need information on how the opponent will play against a fixed policy π. Under Assumption 2, we prove that the existing MDP-E method (Even-Dar et al., 2009), which is designed for the oblivious adversary, will have no- policy regret property against the non-oblivious strategic adversary in our setting. Intuitively, MDP-E maintains a no-external regret algorithm (i.e., MWU) in each state to bound the local regret, thus the global regret can be bounded accordingly. The pseudocode of MDP-E is given in Algorithm 19. The following lemma links the relationship between the external-regret of the adversary and the regret with respect to the policy stationary distribution:

Lemma 4.3. *Under MDP-E played by the agent, the external-regret of the adversary in Assumption 2 can be expressed as:*

$$\begin{aligned} R_T(\boldsymbol{l}) &= \mathbb{E}_{X,A}\left[\sum_{t=1}^{T} \boldsymbol{l}(X_t, A_t)\right] - \mathbb{E}_{X,A}\left[\sum_{t=1}^{T} \boldsymbol{l}_t^{\pi_t}(X_t, A_t)\right] \\ &= \sum_{t=1}^{T} \langle \boldsymbol{l}, \boldsymbol{d}_{\pi_t} \rangle - \sum_{t=1}^{T} \langle \boldsymbol{l}_t, \boldsymbol{d}_{\pi_t} \rangle + \mathcal{O}\big(\tau^2 \sqrt{T \log(|A|)}\big). \end{aligned}$$

Proof. It is sufficient to show that for any sequence of $\boldsymbol{l}_t$

$$\mathbb{E}_{X,A}\left[\sum_{t=1}^{T} \boldsymbol{l}_t(X_t, A_t)\right] - \sum_{t=1}^{T} \langle \boldsymbol{l}_t, \boldsymbol{d}_{\pi_t} \rangle = \mathcal{O}(\tau^2 \sqrt{T \log(|A|)}),$$

where $\boldsymbol{l}_t$ denotes the loss vector of the adversary when the agent follows $\pi_1, \pi_2, \dots$ (i.e., the same as $\boldsymbol{l}_t^{\pi_t}$). Using the consequence of Lemma 5.2 in Even-Dar et al. (2009) [2], for

[2] For the completeness of the work, we provide the lemma in Appendix A.

any sequence of $\boldsymbol{l}_t$ we have

$$\begin{aligned}
&\mathbb{E}_{X,A}\left[\sum_{t=1}^{T}\boldsymbol{l}_t(X_t,A_t)\right]-\sum_{t=1}^{T}\langle\boldsymbol{l}_t,\boldsymbol{d}_{\pi_t}\rangle\\
&=\sum_{t=1}^{T}\langle\boldsymbol{l}_t,\boldsymbol{v}_t-\boldsymbol{d}_{\pi_t}\rangle\leq\sum_{t=1}^{T}|\langle\boldsymbol{l}_t,\boldsymbol{v}_t-\boldsymbol{d}_{\pi_t}\rangle|\leq\sum_{t=1}^{T}\|\boldsymbol{v}_t-\boldsymbol{d}_{\pi_t}\|_1\\
&\leq\sum_{t=1}^{T}2\tau^2\sqrt{\frac{\log(|A|)}{t}}+2e^{-t/\tau}\leq 4\tau^2\sqrt{T\log(|A|)}+2(1+\tau)=\mathcal{O}\big(\tau^2\sqrt{T\log(|A|)}\big).
\end{aligned}$$

The proof is complete. □

Based on Lemma 4.3, we can tell that the sublinear regret will hold if and only if the adversary maintains a sublinear regret with respect to the agent's policy stationary distribution. As we assume that after each time t, the adversary can observe the stationary distribution $\boldsymbol{d}_{\pi_t}$, then by applying standard no-external regret algorithm for online linear optimization against the feedback $\boldsymbol{d}_{\pi_t}$ (i.e., MWU), the adversary can guarantee a good performance for himself. Thus, the Assumption 2 for the adversary is justifiable.

In the rest of the chapter, without loss of generality, we will study the case where the external-regret of the adversary with respect to the agent's policy stationary distribution has the following bound (i.e., the adversary follows optimal no-external regret algorithms such as MWU, FTRL with respect to policy stationary distribution of the agent [3]):

$$\max_{\boldsymbol{l}\in\Delta_L}\left(\sum_{t=1}^{T}\langle\boldsymbol{l},\boldsymbol{d}_{\pi_t}\rangle-\sum_{t=1}^{T}\langle\boldsymbol{l}_t,\boldsymbol{d}_{\pi_t}\rangle\right)=\sqrt{\frac{T\log(L)}{2}}.$$

The next lemma provides a lower bound for the performance of a fixed policy of the agent against a strategic adversary.

Lemma 4.4. *Suppose the agent follows a fixed stationary strategy π, then the adversary will converge to the best response to the fixed stationary strategy and*

$$\sum_{t=1}^{T}\langle\boldsymbol{l}_t^{\pi},\boldsymbol{d}_{\pi}\rangle\geq Tv-\sqrt{\frac{T\log(L)}{2}}.$$

Proof. From Lemma 4.3, if the adversary follows a no-regret algorithm to achieve good performance in Assumption 2, then the adversary must follow a no-external regret algorithm with respect to the policy's stationary distribution. Without loss of generality, we can assume that the adversary follows the Multiplicative Weight Update with respect to the policy's stationary distribution $\boldsymbol{d}_{\pi}$. Then following the property of Multiplicative

[3] If the adversary does not follow the optimal bound (i.e., irrational), then regret bound of the agent will change accordingly.

Weight Update in online linear problem, we have

$$\max_{\boldsymbol{l}\in L}\langle \boldsymbol{l}, \boldsymbol{d}_\pi\rangle - \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{l}_t^\pi, \boldsymbol{d}_\pi\rangle \leq \sqrt{\frac{\log(L)}{2T}}.$$

From the famous minimax theorem (Neumann, 1928) we also have:

$$\max_{\boldsymbol{l}\in L}\langle \boldsymbol{l}, \boldsymbol{d}_\pi\rangle \geq \min_{\boldsymbol{d}_\pi\in\boldsymbol{d}_\Pi}\max_{\boldsymbol{l}\in L}\langle \boldsymbol{l}, \boldsymbol{d}_\pi\rangle = v.$$

Thus we have

$$\sum_{t=1}^{T}\langle \boldsymbol{l}_t^\pi, \boldsymbol{d}_\pi\rangle \geq T\max_{\boldsymbol{l}\in L}\langle \boldsymbol{l}, \boldsymbol{d}_\pi\rangle - \sqrt{\frac{T\log(L)}{2}} \geq Tv - \sqrt{\frac{T\log(L)}{2}}.$$

□

From Lemma 4.4, we can prove the following theorem:

Theorem 4.5. *Suppose the agent follows MDP-E Algorithm 19 against a strategic adversary, then the regret with respect to the stationary distribution will be bounded by*

$$\sum_{t=1}^{T}\left\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\right\rangle - \sum_{t=1}^{T}\left\langle \boldsymbol{l}_t^{\pi}, \boldsymbol{d}_{\pi}\right\rangle \leq \sqrt{\frac{T\log(L)}{2}} + 3\tau\sqrt{\frac{T\log(|A|)}{2}}.$$

Proof. From Lemma 4.4, it is sufficient to show that

$$\sum_{t=1}^{T}\left\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\right\rangle \leq Tv + 3\tau\sqrt{\frac{T\log(|A|)}{2}}.$$

Since the agent uses a no-regret algorithm with respect to the stationary distribution (i.e., MDP-E), following the same argument in Theorem 5.3 in Even-Dar et al. (2009) we have

$$\sum_{t=1}^{T}\left\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\right\rangle \leq T\min_{\boldsymbol{d}_\pi}\left\langle \hat{\boldsymbol{l}}, \boldsymbol{d}_\pi\right\rangle + 3\tau\sqrt{\frac{T\log(|A|)}{2}},$$

where $\hat{\boldsymbol{l}} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{l}_t^{\pi_t}$. From the minimax equilibrium, we also have

$$\min_{\boldsymbol{d}_\pi}\langle \hat{\boldsymbol{l}}, \boldsymbol{d}_\pi\rangle \leq \max_{\boldsymbol{l}\in\Delta_L}\min_{\boldsymbol{d}_\pi\in\boldsymbol{d}_\Pi}\langle \boldsymbol{l}, \boldsymbol{d}_\pi\rangle = v.$$

Thus, the proof is complete. □

Now, we can make the link between the stationary regret and the regret of the agent in Equation (4.1).

Theorem 4.6. *Suppose the agent follows MDP-E Algorithm 19 against a strategic adversary, then the agent's regret in Equation (4.1) will be bounded by*

$$R_T(\pi) = \mathcal{O}(\sqrt{T\log(L)} + \tau^2\sqrt{T\log(|A|)}).$$

Proof. Using the consequence of Lemma 5.2 in Even-Dar et al. (2009), for any sequence of $\boldsymbol{l}_t$ we have

$$\begin{aligned}
&\sum_{t=1}^{T}\langle \boldsymbol{l}_t, \boldsymbol{v}_t - \boldsymbol{d}_{\pi_t}\rangle \\
&\leq \sum_{t=1}^{T} |\langle \boldsymbol{l}_t, \boldsymbol{v}_t - \boldsymbol{d}_{\pi_t}\rangle| \\
&\leq \sum_{t=1}^{T} \|\boldsymbol{v}_t - \boldsymbol{d}_{\pi_t}\|_1 \leq \sum_{t=1}^{T} 2\tau^2\sqrt{\frac{\log(|A|)}{t}} + 2e^{-t/\tau} \\
&\leq 4\tau^2\sqrt{T\log(|A|)} + 2(1+\tau) = \mathcal{O}\big(\tau^2\sqrt{T\log(|A|)}\big).
\end{aligned}$$

Thus we have

$$\sum_{t=1}^{T} |\langle \boldsymbol{l}_t, \boldsymbol{v}_t - \boldsymbol{d}_{\pi_t}\rangle| \leq 2(1+\tau) + 4\tau^2\sqrt{T\log(|A|)}.$$

Furthermore, if the agent uses a fixed policy π then by Lemma 4.4, we have

$$|\sum_{t=1}^{T}\langle \boldsymbol{l}_t, \boldsymbol{d}_\pi - \boldsymbol{v}_t^\pi\rangle| \leq 2\tau + 2.$$

Since the agent uses MDP-E, a no-external regret algorithm, following the same argument in Theorem 4.1 in Even-Dar et al. (2009) we have

$$\begin{aligned}
\sum_{t=1}^{T}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle &\leq T\min_{\boldsymbol{d}_\pi}\langle \hat{\boldsymbol{l}}, \boldsymbol{d}_\pi\rangle + 3\tau\sqrt{\frac{T\log(|A|)}{2}} \\
&\leq Tv + 3\tau\sqrt{\frac{T\log(|A|)}{2}}.
\end{aligned}$$

Along with Lemma 4.4, we have

$$\begin{aligned}
&\sum_{t=1}^{T}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \sum_{t=1}^{T}\langle \boldsymbol{l}_t^{\pi}, \boldsymbol{d}_\pi\rangle \\
&\leq \left(Tv + 3\tau\sqrt{\frac{T\log(|A|)}{2}}\right) - \left(Tv - \sqrt{\frac{T\log(L)}{2}}\right) \\
&= 3\tau\sqrt{\frac{T\log(|A|)}{2}} + \sqrt{\frac{T\log(L)}{2}}.
\end{aligned}$$

Using the above two inequalities, we can bound the regret of the agent with respect to the regret of the policy's stationary distribution:

$$\begin{aligned}
R_T(\pi) &= \mathbb{E}_{x,a}\left[\sum_{t=1}^{T} \boldsymbol{l}_t^{\pi_t}(x_t, a_t)\right] - \mathbb{E}_{x,a}\left[\sum_{t=1}^{T} \boldsymbol{l}_t^{\pi}(x_t^{\pi}, a_t^{\pi})\right] \\
&= \sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{v}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi}, \boldsymbol{v}_t^{\pi} \rangle \\
&\leq \sum_{t=1}^{T} \left(\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t} \rangle + |\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{v}_t - \boldsymbol{d}_{\pi_t} \rangle| \right) - \sum_{t=1}^{T} \left(\langle \boldsymbol{l}_t^{\pi}, \boldsymbol{d}_{\pi} \rangle - |\langle \boldsymbol{l}_t^{\pi}, \boldsymbol{v}_t^{\pi} - \boldsymbol{d}_{\pi} \rangle| \right) \\
&\leq \sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t} \rangle - \sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi}, \boldsymbol{d}_{\pi} \rangle + 2(1+\tau) + 4\tau^2 \sqrt{T \log(|A|)} + 2 + 2\tau \\
&\leq \sqrt{\frac{T \log(L)}{2}} + 3\tau \sqrt{\frac{T \log(|A|)}{2}} + 4(1+\tau) + 4\tau^2 \sqrt{T \log(|A|)} \\
&= \mathcal{O}(\sqrt{T \log(L)} + \tau^2 \sqrt{T \log(|A|)}).
\end{aligned}$$

The proof is complete. □

We note that Theorem 4.6 will hold true for a larger set of adversary outside Assumption 2 (e.g., FP (Brown, 1951)) satisfying the following property: for every fixed policy of the agent, the adversary's policy converges to the best response with respect to this fixed policy. With this property, we can bound the performance of the agent's fixed policy in Lemma 4.4 and thus derive the regret bound of the algorithm. Note that the regret bound in Theorem 4.6 will depend on the rate of convergence to best response against the agent's fixed policy.

As we have shown in previous theorems, the dynamic of playing a no-regret algorithm in OMDPs against a strategic adversary can be interpreted as a two-player zero-sum game setting with the corresponding stationary distribution. From the classical saddle point theorem (Freund and Schapire, 1999), if both players follow a no-regret algorithm then the average strategies will converge to the saddle point (i.e., a NE).

Theorem 4.7. *Suppose the agent follows MDP-E against the strategic adversary, then the average strategies of both the agent and the adversary will converge to the ϵ_t-Nash equilibrium of the game with:*

$$\epsilon_T = \sqrt{\frac{\log(L)}{2T}} + 3\tau \sqrt{\frac{\log(|A|)}{2T}}.$$

Proof. Since the agent and the adversary use no-regret algorithms with respect to the policy's stationary distribution, we can use the property of regret bound in a normal-form game to apply. Thus we have

$$\max_{\boldsymbol{l}\in L}\langle \boldsymbol{l}, \hat{\boldsymbol{d}}_\pi\rangle - \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle \le \sqrt{\frac{\log(L)}{2T}},$$
$$\frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \min_{\boldsymbol{d}_\pi}\langle \hat{\boldsymbol{l}}, \boldsymbol{d}_\pi\rangle \le 3\tau\sqrt{\frac{\log(|A|)}{2T}},$$

where $\hat{\boldsymbol{d}_\pi} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{d}_{\pi_t}$ and $\hat{\boldsymbol{l}} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{l}_t^{\pi_t}$. From this, we can prove that

$$\begin{aligned}\langle \hat{\boldsymbol{l}}, \hat{\boldsymbol{d}}_\pi\rangle &\ge \min_{\boldsymbol{d}_\pi}\langle \hat{\boldsymbol{l}}, \boldsymbol{d}_\pi\rangle \ge \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - 3\tau\sqrt{\frac{\log(|A|)}{2T}} \\ &\ge \max_{\boldsymbol{l}\in L}\langle \boldsymbol{l}, \hat{\boldsymbol{d}}_\pi\rangle - \sqrt{\frac{\log(L)}{2T}} - 3\tau\sqrt{\frac{\log(|A|)}{2T}},\end{aligned}$$

and,

$$\begin{aligned}\langle \hat{\boldsymbol{l}}, \hat{\boldsymbol{d}}_\pi\rangle &\le \max_{\boldsymbol{l}\in L}\langle \boldsymbol{l}, \hat{\boldsymbol{d}}_\pi\rangle \le \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle + \sqrt{\frac{\log(L)}{2T}} \\ &\le \min_{\boldsymbol{d}_\pi}\langle \hat{\boldsymbol{l}}, \boldsymbol{d}_\pi\rangle + 3\tau\sqrt{\frac{\log(|A|)}{2T}} + \sqrt{\frac{\log(L)}{2T}}.\end{aligned}$$

Thus, with $\epsilon_t = \sqrt{\frac{\log(L)}{2T}} + 3\tau\sqrt{\frac{\log(|A|)}{2T}}$, we derive

$$\max_{\boldsymbol{l}\in L}\langle \boldsymbol{l}, \hat{\boldsymbol{d}}_\pi\rangle - \epsilon_t \le \langle \hat{\boldsymbol{l}}, \hat{\boldsymbol{d}}_\pi\rangle \le \min_{\boldsymbol{d}_\pi}\langle \hat{\boldsymbol{l}}, \boldsymbol{d}_\pi\rangle + \epsilon_t.$$

By definition, $(\hat{\boldsymbol{l}}, \hat{\boldsymbol{d}}_\pi)$ is ϵ_t-Nash equilibrium. □

With the sublinear convergence rate to an NE, the dynamic between MDP-E and no-regret adversary (i.e., MWU) provides an efficient method to solve the single-controller SGs.

4.5 MDP-Online Oracle Expert Algorithm

As shown in the previous section, we can bound the regret in Equation (4.1) by bounding the regret with respect to the stationary distribution. In MDP-E, the regret bound (i.e., $\mathcal{O}\big(\sqrt{T\log(L)} + \tau^2\sqrt{T\log(|A|)}\big)$) depends on the size of pure strategy set (i.e., $|A|$) thus it becomes less efficient when the agent has a prohibitively large pure strategy set.

Interestingly, a recent paper by Dinh et al. (2022) suggested that on normal-form games, it is possible to achieve a better regret bound where it only depends on the support size

of NE rather than $|A|$. Unfortunately, extending this finding for OMDPs is highly non-trivial. The method in Dinh et al. (2022) is designed for normal-form games only; in the worst scenario, its regret bound will depend on the size of pure strategy set, which is huge under our settings (i.e., $|A|^{|S|}$).

In this section, we provide a no-policy regret algorithm: MDP-Online Oracle Expert (MDP-OOE). It achieves the regret bound that only depends on the size of NE support rather than the size of the game. We start by presenting the small NE support size assumption.

Assumption 4 (Small Support Size of NE). Let $(\boldsymbol{d}_{\boldsymbol{\pi}^*}, \boldsymbol{l}^*)$ be a Nash equilibrium of the game of size $|A|^{|S|} \times L$. We assume the support size of $(\boldsymbol{d}_{\boldsymbol{\pi}^*}, \boldsymbol{l}^*)$ is smaller than the game size: $\max\left(|\operatorname{supp}(\boldsymbol{d}_{\boldsymbol{\pi}^*})|, |\operatorname{supp}(\boldsymbol{l}^*)|\right) < \min(|A|^{|S|}, L)$.

Note that the assumption of small support size of NE holds in many real-world games (Czarnecki et al., 2020; Dinh et al., 2022; Perez-Nieves et al., 2021; Liu et al., 2021; Yang et al., 2021). In addition, we prove that such an assumption also holds in cases where the loss vectors $[\boldsymbol{l}_1, ..., \boldsymbol{l}_L]$ are sampled from a continuous distribution and the size of the loss vector set L is small compared to the agent's pure strategy set, that is, $|A|^{|S|} \gg L$, thus further justifying the generality of this assumption.

Lemma 4.8. *Suppose that all loss functions are sampled from a continuous distribution and the size of the loss function set is small compared to the agent's pure strategy set (i.e., $|A|^{|S|} \gg L$). Let $(\boldsymbol{d}_{\boldsymbol{\pi}^*}, \boldsymbol{l}^*)$ be a Nash equilibrium of the game of size $|A|^{|S|} \times L$. Then we have*

$$\max\left(|\operatorname{supp}(\boldsymbol{d}_{\boldsymbol{\pi}^*})|, |\operatorname{supp}(\boldsymbol{l}^*)|\right) \leq L.$$

Proof. Within the set of all zero-sum games, the set of zero-sum games with non-unique equilibrium has Lebesgue measure zero (Bailey and Piliouras, 2018). Thus, if the loss function's entries are sampled from a continuous distribution, then with probability one, the game has a unique NE. Following the Theorem 1 in Bohnenblust et al. (1950) for game with unique NE, we have

$$|\operatorname{supp}(\boldsymbol{d}_{\boldsymbol{\pi}^*})| = |\operatorname{supp}(\boldsymbol{l}^*)|.$$

We also note that the support size of the NE can not exceed the size of the game:

$$|\operatorname{supp}(\boldsymbol{d}_{\boldsymbol{\pi}^*})| \leq |A|^{|S|}; \quad |\operatorname{supp}(\boldsymbol{l}^*)| \leq L.$$

Thus we have

$$\max\left(|\operatorname{supp}(\boldsymbol{d}_{\boldsymbol{\pi}^*})|, |\operatorname{supp}(\boldsymbol{l}^*)|\right) = |\operatorname{supp}(\boldsymbol{l}^*)| \leq L.$$

□

Since the pure strategy set of the adversary L is much smaller compared to the pure strategy set of the agent $|A|^{|S|}$, the support size of NE will highly likely be smaller compared to the size of agent's strategy set. Thus the agent can exploit this extra information to achieve better performance.

We now present the MDP-Online Oracle Expert (MDP-OOE) algorithm as follows. MDP-OOE maintains a set of effective strategy A_t^s in each state. In each iteration, the best response with respect to the average loss function will be calculated. If all the actions in the best response are included in the current effective strategy set A_t^s for each state, then the algorithm continues with the current set A_t^s in each state. Otherwise, the algorithm updates the set of effective strategy in steps 8 and 9 of Algorithm 20. We define the period of consecutive iterations as one *time window* T_i in which the set of effective strategy A_t^s stays fixed, i.e., $T_i := \left\{t \;\middle|\; |A_t^s| = i\right\}$. Intuitively, since both the agent and the adversary use a no-regret algorithm to play, the average strategy of both players will converge to the NE of the game. Under the small NE support size assumption, the size of the agent's effective strategy set is also small compared to the whole pure strategy set (i.e., $|A|^{|S|}$). MDP-OOE ignores the pure strategies with poor average performance and only considers ones with high average performance. The regret bound with respect to the agent's stationary distribution is given as follows:

Algorithm 20 MDP-Online Oracle Expert

1: **Initialise:** Sets $A_0^1, \dots A_0^S$ of effective strategy set in each state
2: **for** $t = 1$ to ∞ **do**
3: $\quad \pi_t = BR(\bar{\boldsymbol{l}})$
4: $\quad$ **if** $\pi_t(s, .) \in A_{t-1}^s$ for all s **then**
5: $\quad\quad A_t^s = A_{t-1}^s$ for all s
6: $\quad\quad$ Using the expert algorithm B_s with effective strategy set A_t^s and the feedback $Q_{\pi_t, \boldsymbol{l}_t}(s, .)$
7: $\quad$ **else if** there exists $\pi_t(s, .) \notin A_{t-1}^s$ **then**
8: $\quad\quad A_t^s = A_{t-1}^s \cup \pi_t(s, .) \quad$ if $\pi_t(s, .) \notin A_{t-1}^s$
9: $\quad\quad A_t^s = A_{t-1}^s \cup a \quad$ if $\pi_t(s, .) \in A_{t-1}^s$ where a is randomly selected from the set A / A_{t-1}^s.
10: $\quad\quad$ Reset the expert algorithm B_s with effective strategy set A_t^s and the feedback $Q_{\pi_t, \boldsymbol{l}_t}(s, .)$
11: $\quad$ **end if**
12: $\quad \bar{\boldsymbol{l}} = \sum_{i=\bar{T}_i}^{T} \boldsymbol{l}_t$
13: **end for**

Theorem 4.9. *Suppose the learning agent uses MDP-OOE Algorithm 20, then the regret with respect to the stationary distribution will be bounded by:*

$$\sum_{t=1}^{T} \left\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t} \right\rangle - \left\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi} \right\rangle \leq 3\tau \left(\sqrt{2Tk \log(k)} + \frac{k \log(k)}{8} \right),$$

where k is the number of time windows.

Proof. We first have:

$$\begin{aligned}
\mathrm{E}_{s\sim \boldsymbol{d}_\pi}[Q_{\pi_t,\boldsymbol{l}_t}(s,\pi)] &= \mathrm{E}_{s\sim \boldsymbol{d}_\pi, a\sim\pi}[Q_{\pi_t,\boldsymbol{l}_t}(s,a)] \\
&= \mathrm{E}_{s\sim \boldsymbol{d}_\pi, a\sim\pi}[\boldsymbol{l}_t(s,a) - \eta_{\boldsymbol{l}_t}(\pi_t) + \mathrm{E}_{s'\sim P_{s,a}}[Q_{\pi_t,\boldsymbol{l}_t}(s',\pi_t)]] \\
&= \mathrm{E}_{s\sim \boldsymbol{d}_\pi, a\sim\pi}[\boldsymbol{l}_t(s,a)] - \eta_{\boldsymbol{l}_t}(\pi_t) + \mathrm{E}_{s\sim \boldsymbol{d}_\pi}[Q_{\pi_t,\boldsymbol{l}_t}(s,\pi_t)] \\
&= \eta_{\boldsymbol{l}_t}(\pi) - \eta_{\boldsymbol{l}_t}(\pi_t) + \mathrm{E}_{s\sim \boldsymbol{d}_\pi}[Q_{\pi_t,\boldsymbol{l}_t}(s,\pi_t)].
\end{aligned}$$

Thus we have

$$\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_\pi\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle = \sum_{s\in S} \boldsymbol{d}_\pi(s)\left(Q_{\pi_t,\boldsymbol{l}_t}(s,\pi) - Q_{\pi_t,\boldsymbol{l}_t}(s,\pi_t)\right). \tag{4.4}$$

Let $T_1, T_2, ..., T_k$ be the time window that the $\mathrm{BR}(\bar{\boldsymbol{l}})$ does not change. Then in that time window, the best response to the current $\bar{\boldsymbol{l}}$ is inside the current pure strategies set in each state. In each time window, following Equation (4.4) we have

$$\sum_{t=|\bar{T}_i|}^{\bar{T}_{i+1}} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_\pi\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle = \sum_{s\in S} \boldsymbol{d}_\pi(s) \sum_{t=|\bar{T}_i|}^{\bar{T}_{i+1}} \left(Q_{\pi_t,\boldsymbol{l}_t}(s,\pi) - Q_{\pi_t,\boldsymbol{l}_t}(s,\pi_t)\right).$$

Since during each time window, the pure strategy set A_t^s does not change, thus we have

$$\min_{\pi\in\Pi} \sum_{t=|\bar{T}_i|}^{\bar{T}_{i+1}} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_\pi\rangle = \min_{\pi\in A^s_{|\bar{T}_i|}} \sum_{t=|\bar{T}_i|}^{\bar{T}_{i+1}} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_\pi\rangle.$$

Thus, in each state s of a time window, the agent only needs to minimize the loss with respect to the action in $A^s_{|\bar{T}_i|}$. Put it differently, the expert algorithm in each state does not need to consider all pure action in each state, but just the current effective strategy set. For a time window T_i, if the agent uses a no-regret algorithm with the current effective action set and the learning rate $\mu_t = \sqrt{8\log(i)/t}$, then the regret in each state will be bounded by (Cesa-Bianchi and Lugosi, 2006):

$$3\tau\left(\sqrt{2|T_i|\log(A_t^s)} + \frac{\log(A_t^s)}{8}\right) \le 3\tau\left(\sqrt{2|T_i|\log(i)} + \frac{\log(i)}{8}\right).$$

Thus, the regret in this time interval will also be bounded by:

$$\sum_{t=|\bar{T}_i|}^{\bar{T}_{i+1}} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_\pi\rangle \le 3\tau\left(\sqrt{2|T_i|\log(i)} + \frac{\log(i)}{8}\right). \tag{4.5}$$

Sum up from $i = 1$ to k in Inequality (4.5) we have

$$\begin{aligned}
&\sum_{t=1}^{T}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi}\rangle \\
&= \sum_{i=1}^{k}\sum_{t=|\bar{T}_i|}^{\bar{T}_{i+1}} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi}\rangle \\
&\leq \sum_{i=1}^{k} 3\tau\left(\sqrt{2|T_i|\log(i)} + \frac{\log(i)}{8}\right) \\
&\leq 3\tau\left(\sqrt{2Tk\log(k)} + \frac{k\log(k)}{8}\right).
\end{aligned}$$

The proof is complete. □

In Algorithm 20, each time the agent updates the effective strategy set A_t^s at state s, exactly one new pure strategy is added into the effective strategy set for each state, thus the number k will be at most $|A|$. Therefore, we have the regret w.r.t the stationary distribution in the worst case will be:

$$3\tau\left(\sqrt{2T|A|\log(|A|)} + \frac{|A|\log(|A|)}{8}\right).$$

However, as shown in (Dinh et al., 2022, Figure 1), the number of iteration in DO method (respectively the number of time window in our setting) is linearly dependent in the support size of the NE, thus with Assumption 4, Algorithm 20 will be highly efficient.

Remark 4.10. The regret bound in Theorem 4.9 will still hold in the case we consider the total average lost instead of average lost in each time window when calculating the best response in Algorithm 20.

Proof. We prove by induction that

$$\min_{\pi\in\Pi}\sum_{t=1}^{\bar{T}_k}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi}\rangle \leq \sum_{j=1}^{k}\left[\sum_{t=\bar{T}_{j-1}+1}^{\bar{T}_j} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_j}\rangle\right],$$

where $\boldsymbol{d}_{\pi_j}$ denotes the best response in the interval $[1, \bar{T}_j]$.

For $k = 1$, the claim is obvious. Suppose the claim is true k. We then have:

$$\begin{aligned}
&\min_{\pi\in\Pi}\sum_{t=1}^{\bar{T}_{k+1}}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi}\rangle = \sum_{t=1}^{\bar{T}_{k+1}}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_{k+1}}\rangle \\
&= \sum_{t=1}^{\bar{T}_k}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_{k+1}}\rangle + \sum_{t=\bar{T}_k+1}^{\bar{T}_{k+1}}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_{k+1}}\rangle \\
&\leq \min_{\pi\in\Pi}\sum_{t=1}^{\bar{T}_k}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi}\rangle + \sum_{t=\bar{T}_k+1}^{\bar{T}_{k+1}}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_{k+1}}\rangle \\
&\leq \sum_{j=1}^{k}\left[\sum_{t=\bar{T}_{j-1}+1}^{\bar{T}_j}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_j}\rangle\right] + \sum_{t=\bar{T}_k+1}^{\bar{T}_{k+1}}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_{k+1}}\rangle \qquad (4.6a) \\
&= \sum_{j=1}^{k+1}\left[\sum_{t=\bar{T}_{j-1}+1}^{\bar{T}_j}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_j}\rangle\right],
\end{aligned}$$

where the inequality (4.6a) dues to the induction assumption. Thus, for all k we have

$$\min_{\pi\in\Pi}\sum_{t=1}^{\bar{T}_k}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi}\rangle \leq \sum_{j=1}^{k}\left[\sum_{t=\bar{T}_{j-1}+1}^{\bar{T}_j}\langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t}\rangle - \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_j}\rangle\right].$$

In other words, the Algorithm 20 will have a similar regret bound when using the best response with respect to the total average strategy of the adversary. □

Given the regret with respect to policy's stationary distribution in Theorem 4.9, we can now derive the regret bound of Algorithm 20 with respect to the true performance:

Theorem 4.11. *Suppose the agent uses MDP-OOE Algorithm 20 against a strategic adversary in our OMDPs setting, then the regret in Equation (4.1) can be bounded by:*

$$R_T(\pi) = \mathcal{O}(\tau^2\sqrt{Tk\log(k)} + \sqrt{T\log(L)}).$$

The full proof is given in Appendix 4.9. Notably, MDP-OOE will not only reduce the regret bound in the case the number of strategy set k is small, it also reduces the computational hardness of computing expert algorithm when the number of experts is prohibitively large.

MDP-Online Oracle Algorithm with ϵ-best response. In Algorithm 20, in each iteration the agent needs to calculate the exact best response to the average loss function $\bar{\boldsymbol{l}}$. Since calculating the exact best response is computationally hard and maybe infeasible in many situations (Vinyals et al., 2019), an alternative way is to consider ϵ-best response. That is, in each iteration in Algorithm 20, the agent can only access to a ϵ-best response

to the average loss function, where ϵ is a predefined parameter. In this situation, we provide the regret analysis for Algorithm 20 as follow.

Theorem 4.12. *Suppose the agent only accesses to ϵ-best response in each iteration when following Algorithm 20. If the adversary follows a no-external regret algorithm then the average strategy of the agent and the adversary will converge to ϵ-Nash equilibrium. Furthermore, the algorithm has ϵ-regret.*

The full proof is given in Appendix 4.9. Theorem 4.12 implies that by following MDP-OOE, the agent can optimise the accuracy level (in terms of ϵ) based on the data that it receives to obtain the convergence rate and regret bound accordingly.

4.6 Last Round Convergence to NE in OMDPs

In this section, we investigate OMDPs where the agent not only aims to minimize the regret but also stabilize the strategies. This is motivated by the fact that changing strategies through repeated games may be undesirable (e.g., see Dinh et al. (2021); Daskalakis and Panageas (2019)). In online learning literature, minimizing regret and achieving the system's stability are often two conflicting goals. That is, if all player in a system follows a no-regret algorithm (e.g., MWU, FTRL) to minimise the regret, then the dynamic of the system will become chaotic and the strategies of players will not converge in the last round (Dinh et al., 2021; Mertikopoulos et al., 2018).

To achieve the goal, we start by studying the scenarios where the agent knows its NE of the game π^*. We then propose an algorithm: Last Round Convergence in OMDPs (LRC-OMDP) that leads to last round convergence to NE of the game in our setting. This is the first algorithm to our knowledge that achieves last round convergence in OMDPs where only the learning agent knows the NE of the game. Notably, this goal is non-trivial to achieve. For example, if the agent keeps following the same strategy (i.e., the NE), then while the system might be stabilised (i.e., the adversary converges to the best response), yet this is still not a no-regret algorithm. Moreover, we notice that understanding the learning dynamics even when the NE is known is still challenging in the multi-agent learning domain. The AWESOME (Conitzer and Sandholm, 2007) and CMLeS (Chakraborty and Stone, 2014) algorithms make significant efforts to achieve convergence to NE under the assumption that each agent has access to a precomputed NE strategy. Compared to these algorithms, LRC-OMDP enjoys the key benefit that it does not require the adversary to know its NE. Importantly, the adversary in our setting can be any type of strategic agent who observes the history and applies a no-regret algorithm to play, rather than being a restricted opponent such as a stationary opponent in AWESOME or a memory-bounded opponent in CMLeS.

Algorithm 21 Last Round Convergence in OMDPs

Input: Current iteration t
Output: Strategy π_t for the agent
for $t = 1, 2, \ldots, T$ **do**
  if $t = 2k - 1, k \in \mathbb{N}$ **then**
    $\pi_t = \pi^*$
  else if $t = 2k, k \in \mathbb{N}$ **then**
    $\hat{\pi}_t(s) = \text{argmin}_{a \in A} Q_{\pi^*, \boldsymbol{l}_t}(s, a)\ \forall s \in S$
    $\alpha_t = \frac{v - \eta_{\boldsymbol{l}_{t-1}}(\hat{\pi}_t)}{\beta}$; $\boldsymbol{d}_{\pi_t} = (1 - \alpha_t)\boldsymbol{d}_{\pi^*} + \alpha_t \boldsymbol{d}_{\hat{\pi}_t}$
    Output π_t via $\boldsymbol{d}_{\pi_t}$
  end if
end for

The LRC-OMDP algorithm can be described as follow. At each odd round, the agent follows the NE strategy π^* so that in the next round, the strategy of the adversary will not deviate from the current strategy. Then, at the following even round, the agent chooses a strategy such that $\boldsymbol{d}_{\pi_t}$ is a direction towards the NE strategy of the adversary. Depending on the distance between the current strategy of the adversary and its NE (which is measured by $v - \eta_{\boldsymbol{l}_{t-1}}(\hat{\pi}_t)$), the agent will choose a step size α_t such that the strategy of the adversary will approach the NE. Note here that β is a constant parameter and depends on the specific no-regret algorithm adversary follows, there is a different optimal value for β. In the case where the adversary follows the MWU algorithm, we can set $\beta = 1$.

We first introduce the condition in which the system achieves stability through the following lemma:

Lemma 4.13. *Let π^* be the NE strategy of the agent. Then, $\boldsymbol{l}$ is the Nash Equilibrium of the adversary if the two following conditions hold:*

$$Q_{\pi^*, \boldsymbol{l}}(s, \pi^*) = \underset{\pi \in \Pi}{\text{argmin}}\, Q_{\pi^*, \boldsymbol{l}}(s, \pi)\ \ \forall s \in S\ \ \textit{and}\ \ \eta_l(\pi^*) = v.$$

Proof. Using the definition of accumulated loss function Q we have

$$\begin{aligned}
&\mathbb{E}_{s \in \boldsymbol{d}_\pi}[Q_{\pi^*, \boldsymbol{l}}(s, \pi)] = \mathbb{E}_{s \in \boldsymbol{d}_\pi, a \in \pi}[Q_{\pi^*, \boldsymbol{l}}(s, a)] \\
&= \mathbb{E}_{s \in \boldsymbol{d}_\pi, a \in \pi}[\boldsymbol{l}(s, a) - \eta_l(\pi^*) + \mathbb{E}_{s' \sim P_{sa}}[Q_{\pi^*, \boldsymbol{l}}(s', \pi^*)]] \\
&= \mathbb{E}_{s \in \boldsymbol{d}_\pi, a \in \pi}[\boldsymbol{l}(s, a) - \eta_l(\pi^*)] + \mathbb{E}_{s \in \boldsymbol{d}_\pi}[Q_{\pi^*, \boldsymbol{l}}(s, \pi^*)] \\
&= \eta_l(\pi) - \eta_l(\pi^*) + \mathbb{E}_{s \in \boldsymbol{d}_\pi}[Q_{\pi^*, \boldsymbol{l}}(s, \pi^*)].
\end{aligned}$$

Thus we have

$$\eta_l(\pi) - \eta_l(\pi^*) = \mathbb{E}_{s \in \boldsymbol{d}_\pi}[Q_{\pi^*, \boldsymbol{l}}(s, \pi) - Q_{\pi^*, \boldsymbol{l}}(s, \pi^*).] \tag{4.7}$$

Since we assume that

$$\mathbb{Q}_{\pi^*, \boldsymbol{l}}(s, \pi^*) = \underset{\pi \in \Pi}{\text{argmin}}\, \mathbb{Q}_{\pi^*, \boldsymbol{l}}(s, \pi)\ \ \forall s \in S,$$

we have

$$\mathbb{Q}_{\pi^*,\boldsymbol{l}}(s,\pi) \geq \mathbb{Q}_{\pi^*,\boldsymbol{l}}(s,\pi^*) \ \ \forall s \in S, \pi \in \Pi.$$

It implies that

$$\mathbb{E}_{s\in\boldsymbol{d}_\pi}[Q_{\pi^*,\boldsymbol{l}}(s,\pi) - Q_{\pi^*,\boldsymbol{l}}(s,\pi^*)] \geq 0 \ \ \forall \pi \in \Pi.$$

Therefore we have

$$\eta_l(\pi) \geq \eta_l(\pi^*) \ \ \forall \pi \in \Pi.$$

Along with the assumption $\eta_l(\pi^*) = v$, we have the following relationship:

$$\operatorname*{argmin}_{\pi\in\Pi} \eta_l(\pi) = \eta_l(\pi^*) = v. \tag{4.8}$$

Now we prove that for the loss function $\boldsymbol{l}$ that satisfies Equation (4.8), then $\boldsymbol{l}$ is NE for the adversary. Let $(\pi^*, \boldsymbol{l}^*)$ be one of the NE of the game. Since the game we are considering is zero-sum game, $(\pi^*, \boldsymbol{l}^*)$ satisfies the famous minimax theorem:

$$\min_{\pi\in\Pi}\max_{\boldsymbol{l}_1\in L}\langle \boldsymbol{l}_1, \boldsymbol{d}_\pi\rangle = \max_{\boldsymbol{l}_1\in L}\min_{\pi\in\Pi}\langle \boldsymbol{l}_1, \boldsymbol{d}_\pi\rangle = v \ \text{ where } \langle \boldsymbol{l}, \boldsymbol{d}_\pi\rangle = \eta_l(\pi).$$

From Equation (4.8) we have

$$v = \min_{\pi\in\Pi}\langle \boldsymbol{l}, \boldsymbol{d}_\pi\rangle \leq \langle \boldsymbol{l}, \boldsymbol{d}_{\pi^*}\rangle. \tag{4.9}$$

Further, since $\boldsymbol{l}^*$ is the NE of the game, then we have

$$v = \langle \boldsymbol{l}^*, \boldsymbol{d}_{\pi^*}\rangle = \max_{\boldsymbol{l}_1\in L}\langle \boldsymbol{l}_1, \boldsymbol{d}_{\pi^*}\rangle \geq \langle \boldsymbol{l}, \boldsymbol{d}_{\pi^*}\rangle. \tag{4.10}$$

From Inequalities (4.9) and (4.10) we have

$$v = \langle \boldsymbol{l}, \boldsymbol{d}_{\pi^*}\rangle = \min_{\pi\in\Pi}\langle \boldsymbol{l}, \boldsymbol{d}_\pi\rangle = \max_{\boldsymbol{l}_1\in L}\langle \boldsymbol{l}_1, \boldsymbol{d}_{\pi^*}\rangle.$$

Thus, by definition $(\boldsymbol{l}, \pi^*)$ is the Nash equilibrium of the game. In other words, the loss function $\boldsymbol{l}$ satisfies the above assumption is the NE of the adversary. □

The above lemma implies that if there is no improvement in the Q-value function for every state and the value of the current loss function equals to the value of the game, then there is last round convergence to the NE. In situations where there is an improvement in one state, the following lemma bounds the value of a new strategy:

Lemma 4.14. *Assume that* $\forall \pi \in \Pi$, $\boldsymbol{d}_\pi(s) > 0$. *Then if there exists* $s \in S$ *such that*

$$Q_{\pi^*,\boldsymbol{l}_t}(s,\pi^*) > \operatorname*{argmin}_{\pi\in\Pi} Q_{\pi^*,\boldsymbol{l}_t}(s,\pi),$$

then for $\pi_{t+1}(s) = \operatorname{argmin}_{a\in A} Q_{\pi^*,\boldsymbol{l}_t}(s,a)\ \forall s \in S$*:*

$$\eta_{\boldsymbol{l}_t}(\pi_{t+1}) < v.$$

Proof. From the minimax theorem, we have

$$\eta_{\boldsymbol{l}_t}(\pi^*) \le \eta_{\boldsymbol{l}^*}(\pi^*) = v \quad \forall \boldsymbol{l} \in L.$$

From the proof of Lemma 4.13 we have

$$\eta_{\boldsymbol{l}_t}(\pi) - \eta_{\boldsymbol{l}_t}(\pi^*) = \mathbb{E}_{s\in \boldsymbol{d}_\pi}[Q_{\pi^*,\boldsymbol{l}_t}(s,\pi) - Q_{\pi^*,\boldsymbol{l}_t}(s,\pi^*)]\ \forall \pi \in \Pi.$$

Since the construction of the new strategy π_{t+1} we have

$$\mathbb{E}_{s\in \boldsymbol{d}_{\pi_{t+1}}}[Q_{\pi^*,\boldsymbol{l}_t}(s,\pi_{t+1}) - Q_{\pi^*,\boldsymbol{l}_t}(s,\pi^*)] < 0,$$

thus we have

$$\eta_{\boldsymbol{l}_t}(\pi) < \eta_{\boldsymbol{l}_t}(\pi^*) \le 0.$$

The proof is complete. □

Based on the above lemmas, we can bound the relative entropy distance between the current strategy of the adversary and a Nash equilibrium:

Lemma 4.15. *Assume that the adversary follows the MWU algorithm with non-increasing step size* μ_t *such that* $\lim_{T\to\infty}\sum_{t=1}^{T}\mu_t = \infty$ *and there exists* $t' \in \mathbb{N}$ *with* $\mu_{t'} \le \frac{1}{3}$*. Then we have*

$$RE(\boldsymbol{l}^*\|\boldsymbol{l}_{2k-1}) - RE(\boldsymbol{l}^*\|\boldsymbol{l}_{2k+1}) \ge \frac{1}{2}\mu_{2k}\alpha_{2k}(v - \eta_{\boldsymbol{l}_{2k-1}}(\hat{\pi}_{2k}))\ \ \forall k \in \mathbb{N}:\ \ 2k \ge t'.$$

Proof. Using the definition of relative entropy we have

$$\begin{aligned}
&\mathrm{RE}\,(\boldsymbol{l}^*\|\boldsymbol{l}_{2k-1}) - \mathrm{RE}\,(\boldsymbol{l}^*\|\boldsymbol{l}_{2k+1}) \\
&= (\mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{2k+1}) - \mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{2k})) + (\mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{2k}) - \mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{2k-1})) \\
&= \left(\sum_{i=1}^{n}\boldsymbol{l}^*(i)\log\left(\frac{\boldsymbol{l}^*(i)}{\boldsymbol{l}_{2k+1}(i)}\right) - \sum_{i=1}^{n}\boldsymbol{l}^*(i)\log\left(\frac{\boldsymbol{l}^*(i)}{\boldsymbol{l}_{2k}(i)}\right)\right) + \\
&\quad \left(\sum_{i=1}^{n}\boldsymbol{l}^*(i)\log\left(\frac{\boldsymbol{l}^*(i)}{\boldsymbol{l}_{2k}(i)}\right) - \sum_{i=1}^{n}\boldsymbol{l}^*(i)\log\left(\frac{\boldsymbol{l}^*(i)}{\boldsymbol{l}_{2k-1}(i)}\right)\right) \\
&= \left(\sum_{i=1}^{n}\boldsymbol{l}^*(i)\log\left(\frac{\boldsymbol{l}_{2k}(i)}{\boldsymbol{l}_{2k+1}(i)}\right)\right) + \left(\sum_{i=1}^{n}\boldsymbol{l}^*(i)\log\left(\frac{\boldsymbol{l}_{2k-1}(i)}{\boldsymbol{l}_{2k}(i)}\right)\right).
\end{aligned}$$

Following the update rule of the Multiplicative Weights Update algorithm we have

$$\begin{aligned}
&\mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{2k+1}) - \mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{2k-1}) \\
&= (-\mu_{2k}\langle \boldsymbol{l}^*, \boldsymbol{d}_{\pi_{2k}}\rangle + \log(Z_{2k})) + (-\mu_{2k-1}\langle \boldsymbol{l}^*, \boldsymbol{d}_{\pi_{2k}}\rangle + \log(Z_{2k-1})) \\
&\leq \left(-\mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{l}_{2k}(i)e^{\mu_{2k}\langle \boldsymbol{e}_i, \boldsymbol{d}_{\pi_{2k}}\rangle}\right)\right) + (-\mu_{2k-1}v + \log(Z_{2k-1})) \qquad (4.11a) \\
&= \left(-\mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{l}_{2k-1}(i)e^{\mu_{2k-1}\langle \boldsymbol{e}_i, \boldsymbol{d}_{\pi_{2k-1}}\rangle}e^{\mu_{2k}\langle \boldsymbol{e}_i, \boldsymbol{d}_{\pi_{2k}}\rangle}\right) - \log(Z_{2k-1})\right) \\
&+ (-\mu_{2k-1}v + \log(Z_{2k-1}))\,,
\end{aligned}$$

where Inequality (4.11a) is due to the fact that $\langle \boldsymbol{l}^*, \boldsymbol{d}_\pi\rangle \geq v\ \forall \pi$. Thus,

$$\begin{aligned}
&\mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{2k+1}) - \mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{2k-1}) \\
&\leq \left(-\mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{l}_{2k-1}(i)e^{\mu_{2k-1}\langle \boldsymbol{e}_i, \boldsymbol{d}_{\pi_{2k-1}}\rangle}e^{\mu_{2k}\langle \boldsymbol{e}_i, \boldsymbol{d}_{\pi_{2k}}\rangle}\right)\right) - \mu_{2k-1}v \\
&\leq \left(-\mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{l}_{2k-1}(i)e^{\mu_{2k-1}v}e^{\mu_{2k}\langle \boldsymbol{e}_i, \boldsymbol{d}_{\pi_{2k}}\rangle}\right)\right) - \mu_{2k-1}v \qquad (4.12a) \\
&= -\mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{l}_{2k-1}(i)e^{\mu_{2k}\langle \boldsymbol{e}_i, \boldsymbol{d}_{\pi_{2k}}\rangle}\right),
\end{aligned}$$

where Inequality (4.12a) is the result of the inequality:

$$\langle \boldsymbol{l}, \boldsymbol{d}_{\pi^*}\rangle \leq v \ \ \forall \boldsymbol{l}.$$

Now, using the update rule of Algorithm 21

$$\boldsymbol{d}_{\pi_{2k}} = (1-\alpha_{2k})\boldsymbol{d}_{\pi^*} + \alpha_{2k}\boldsymbol{d}_{\hat{\pi}_{2k}},$$

we have

$$\begin{aligned}
&\mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{2k+1}) - \mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{2k-1}) \\
&\leq -\mu_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{l}_{2k-1}(i)e^{\mu_{2k}((1-\alpha_{2k})\langle \boldsymbol{e}_i, \boldsymbol{d}_{\pi^*}\rangle + \alpha_{2k}\langle \boldsymbol{e}_i, \boldsymbol{d}_{\hat{\pi}_{2k}}\rangle)}\right) \\
&\leq -\mu_{2k}\alpha_{2k}v + \log\left(\sum_{i=1}^{n} \boldsymbol{l}_{2k-1}(i)e^{\mu_{2k}\alpha_{2k}\langle \boldsymbol{e}_i, \boldsymbol{d}_{\hat{\pi}_{2k}}\rangle}\right).
\end{aligned}$$

Denote $f(\boldsymbol{l}_{2k-1}) = \langle \boldsymbol{l}_{2k-1}, \boldsymbol{d}_{\hat{\pi}_{2k}} \rangle$, we then have

$$
\begin{aligned}
&\mathrm{RE}(\boldsymbol{l}^* \| \boldsymbol{l}_{2k+1}) - \mathrm{RE}(\boldsymbol{l}^* \| \boldsymbol{l}_{2k-1}) \\
&\leq -\mu_{2k}\alpha_{2k} v + \log\left(\sum_{i=1}^{n} \boldsymbol{l}_{2k-1}(i) e^{\mu_{2k}\alpha_{2k}\langle \boldsymbol{e}_i, \boldsymbol{d}_{\hat{\pi}_{2k}}\rangle}\right) \\
&= \mu_{2k}\alpha_{2k}(1-v) + \log\left(\sum_{i=1}^{n} \boldsymbol{l}_{2k-1}(i) e^{-\mu_{2k}\alpha_{2k}(1-\langle \boldsymbol{e}_i, \boldsymbol{d}_{\hat{\pi}_{2k}}\rangle)}\right) && \text{(4.14a)} \\
&\leq \mu_{2k}\alpha_{2k}(1-v) + \log\left(\sum_{i=1}^{n} \boldsymbol{l}_{2k-1}(i)(1-(1-e^{-\mu_{2k}\alpha_{2k}})(1-\langle \boldsymbol{e}_i, \boldsymbol{d}_{\hat{\pi}_{2k}}\rangle))\right) && \text{(4.14b)} \\
&= \mu_{2k}\alpha_{2k}(1-v) + \log\left(1-(1-e^{-\mu_{2k}\alpha_{2k}})(1-\langle \boldsymbol{l}_{2k-1}, \boldsymbol{d}_{\hat{\pi}_{2k}}\rangle)\right) \\
&\leq \mu_{2k}\alpha_{2k}(1-v) - (1-e^{-\mu_{2k}\alpha_{2k}})(1-\langle \boldsymbol{l}_{2k-1}, \boldsymbol{d}_{\hat{\pi}_{2k}}\rangle) && \text{(4.14c)} \\
&= \mu_{2k}\alpha_{2k}(1-v) - (1-e^{-\mu_{2k}\alpha_{2k}})(1-f(\boldsymbol{l}_{2k-1})),
\end{aligned}
$$

Equation (4.14a) is created by adding and subtracting $\mu_{2k}\alpha_{2k}$ on the first and second terms.

Inequalities (4.14*b*, 4.14*c*) are due to

$$\beta^x \leq 1-(1-\beta)x \quad \forall \beta \geq 0\ \boldsymbol{l} \in [0,1] \text{ and } \log(1-x) \leq -x \ \ \forall x < 1.$$

We can develop Inequality (4.14c) further as

$$
\begin{aligned}
&\mathrm{RE}(\boldsymbol{l}^* \| \boldsymbol{l}_{2k+1}) - \mathrm{RE}(\boldsymbol{l}^* \| \boldsymbol{l}_{2k-1}) \\
&\leq \mu_{2k}\alpha_{2k}(1-v) - \left(1-e^{-\mu_{2k}\alpha_{2k}}\right)(1-f(\boldsymbol{l}_{2k-1})) \\
&\leq \mu_{2k}\alpha_{2k}(1-v) - \left(1-\left(1-\mu_{2k}\alpha_{2k}+\frac{1}{2}(\mu_{2k}\alpha_{2k})^2\right)\right)(1-f(\boldsymbol{l}_{2k-1})) && \text{(4.15a)} \\
&= \mu_{2k}\alpha_{2k}(f(\boldsymbol{l}_{2k-1})-v) + \frac{1}{2}(\mu_{2k}\alpha_{2k})^2(1-f(\boldsymbol{l}_{2k-1})) \\
&\leq \mu_{2k}\alpha_{2k}(f(\boldsymbol{l}_{2k-1})-v) + \frac{1}{2}\mu_{2k}\alpha_{2k}\mu_{2k}\frac{v-f(\boldsymbol{l}_{2k-1})}{\beta}(1-f(\boldsymbol{l}_{2k-1})) && \text{(4.15b)} \\
&\leq \mu_{2k}\alpha_{2k}(f(\boldsymbol{l}_{2k-1})-v) + \frac{1}{2}\mu_{2k}\alpha_{2k}\ (v-f(\boldsymbol{l}_{2k-1})) && \text{(4.15c)} \\
&= -\frac{1}{2}\mu_{2k}\alpha_{2k}(v-f(\boldsymbol{l}_{2k-1})) \leq 0.
\end{aligned}
$$

Here, Inequality (4.15*a*) is due to $e^x \leq 1+x+\frac{1}{2}x^2 \ \ \forall \boldsymbol{l} \in [-\infty, 0]$, Inequality (4.15b) comes from the definition of α_t:

$$\alpha_t = \frac{v-f(\boldsymbol{l}_{2k-1})}{\beta},\ \beta \geq 1-f(\boldsymbol{l}),\ f(\boldsymbol{l}_{2k-1}) \leq 1.$$

Finally, Inequality (4.15*c*) comes from the choice of k at the beginning of the proof, i.e., $\mu_{2k} \leq 1$. □

we finally reach the last round convergence of LRC-MDP in Algorithm 21.

Theorem 4.16. *Assume that the adversary follows the MWU algorithm with non-increasing step size μ_t such that $\lim_{T\to\infty}\sum_{t=1}^{T}\mu_t = \infty$ and there exists $t' \in \mathbb{N}$ with $\mu_{t'} \leq \frac{1}{3}$. If the agent follows Algorithm 21 then there exists a Nash equilibrium $\boldsymbol{l}^*$ for the adversary such that $lim_{t\to\infty}\boldsymbol{l}_t = \boldsymbol{l}^*$ almost everywhere and $lim_{t\to\infty}\pi_t = \pi^*$.*

Proof. We focus on the regret analysis with respect to the stationary distribution $\boldsymbol{d}_{\pi_t}$. Let $\boldsymbol{l}^*$ be a minimax equilibrium strategy of the adversary ($\boldsymbol{l}^*$ may not be unique). Following the above Lemma, for all $k \in \mathbb{N}$ such that $2k \geq t'$, we have

$$\mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{2k+1}) - \mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{2k-1}) \leq -\frac{1}{2}\mu_{2k}\alpha_{2k}(v - f(\boldsymbol{l}_{2k-1})), \tag{4.16}$$

where we denote $f(\boldsymbol{l}_{2k-1}) = \langle \boldsymbol{l}_{2k-1}, \boldsymbol{d}_{\hat{\pi}_{2k}}\rangle$. Thus, the sequence of relative entropy $\mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{2k-1})$ is non-increasing for all $k \geq \frac{t'}{2}$. As the sequence is bounded below by 0, it has a limit for any minimax equilibrium strategy $\boldsymbol{l}^*$. Since t' is a finite number and $\sum_{t=1}^{\infty}\mu_t = \infty$, we have $\sum_{t=t'}^{\infty}\mu_t = \infty$. Thus,

$$\lim_{T\to\infty}\sum_{k=\left\lceil\frac{t'}{2}\right\rceil}^{T}\mu_{2k} = \infty.$$

We will prove that $\forall\epsilon > 0$, $\exists h \in \mathbb{N}$ such that when the agent follows Algorithm 21 and the adversary follows MWU algorithm, the adversary will play strategy $\boldsymbol{l}_h$ at round h and $v - f(\boldsymbol{l}_h) \leq \epsilon$. In particular, we prove this by contradiction. That is, suppose that $\exists\epsilon > 0$ such that $\forall h \in \mathbb{N}$, $v - f(\boldsymbol{l}_h) > \epsilon$. Then $\forall k \in \mathbb{N}$,

$$\alpha_{2k}(v - f(\boldsymbol{l}_{2k-1})) = \frac{(v - f(\boldsymbol{l}_{2k-1}))^2}{\beta} > \frac{\epsilon^2}{\beta}.$$

Let k vary from $\left\lceil\frac{t'}{2}\right\rceil$ to T in Equation (4.16). By summing over k, we obtain:

$$\begin{aligned}\mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{2T+1}) &\leq \mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{t'}) - \frac{1}{2}\sum_{k=\left\lceil\frac{t'}{2}\right\rceil}^{T}\mu_{2k}\alpha_{2k}(v - f(\boldsymbol{l}_{2k-1}))\\ &\leq \mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{t'}) - \frac{1}{2}\frac{e^2}{\beta}\sum_{k=\left\lceil\frac{t'}{2}\right\rceil}^{T}\mu_{2k}.\end{aligned}$$

Since $\lim_{T\to\infty}\sum_{k=\left\lceil\frac{t'}{2}\right\rceil}^{T}\mu_{2k} = \infty$ and $\mathrm{RE}(\boldsymbol{l}^*\|\boldsymbol{l}_{T+1}) \geq 0$, it contradicts our assumption about $\forall h \in \mathbb{N}$, $v - f(\boldsymbol{l}_h) > \epsilon$.

Now, we take a sequence of $\epsilon_k > 0$ such that $\lim_{k\to\infty}\epsilon_k = 0$. Then for each k, there exists $\boldsymbol{l}_{t_k} \in \Delta_n$ such that $v - \epsilon_k \leq f(\boldsymbol{l}_{t_k}) \leq v$. As Δ_n is a compact set and $\boldsymbol{l}_{t_k}$ is

bounded then following the Bolzano-Weierstrass theorem, there is a convergence subsequence $\boldsymbol{l}_{\bar{t}_k}$. The limit of that sequence, $\bar{\boldsymbol{l}}^*$, is a minimax equilibrium strategy of the row player (since $f(\bar{\boldsymbol{l}}^*) = f(\lim_{k\to\infty} \boldsymbol{l}_{\bar{t}_k}) = \lim_{k\to\infty} f(\boldsymbol{l}_{\bar{t}_k}) = v$). Combining with the fact that $\mathrm{RE}(\bar{\boldsymbol{l}}^* \| \boldsymbol{l}_{2k-1})$ is non-increasing for $k \geq \left\lceil \frac{t'}{2} \right\rceil$ and $\mathrm{RE}(\bar{\boldsymbol{l}}^* \| \bar{\boldsymbol{l}}^*) = 0$, we have $\lim_{k\to\infty} \mathrm{RE}(\bar{\boldsymbol{l}}^* \| \boldsymbol{l}_{2k-1}) = 0$. We also note that

$$\begin{aligned}\mathrm{RE}(\bar{\boldsymbol{l}}^* \| \boldsymbol{l}_{2k}) - \mathrm{RE}(\bar{\boldsymbol{l}}^* \| \boldsymbol{l}_{2k-1}) &= -\mu_{2k-1} \langle \bar{\boldsymbol{l}}^*, \boldsymbol{d}_{\pi_{2k-1}} \rangle + \log \left(\sum_{i=1}^{n} \boldsymbol{l}_{2k-1}(i) e^{\mu_{2k-1} \langle \boldsymbol{e}_i, \boldsymbol{d}_{\pi^*} \rangle} \right) \\ &\leq -\mu_{2k-1} v + \log \left(\sum_{i=1}^{n} \boldsymbol{l}_{2k-1}(i) e^{\mu_{2k-1} v} \right) = 0,\end{aligned}$$

following the fact that $\langle \bar{\boldsymbol{l}}^*, \boldsymbol{d}_\pi \rangle \geq v$ for all $\pi \in \Pi$ and $\langle \boldsymbol{l}, \boldsymbol{d}_{\pi^*} \rangle \leq v$ for all $\boldsymbol{l}$. Thus, we have $\lim_{k\to\infty} \mathrm{RE}(\bar{\boldsymbol{l}}^* \| \boldsymbol{l}_{2k}) = 0$ as well. Subsequently, $\lim_{t\to\infty} \mathrm{RE}(\bar{\boldsymbol{l}}^* \| \boldsymbol{l}_t) = 0$, which concludes the proof. □

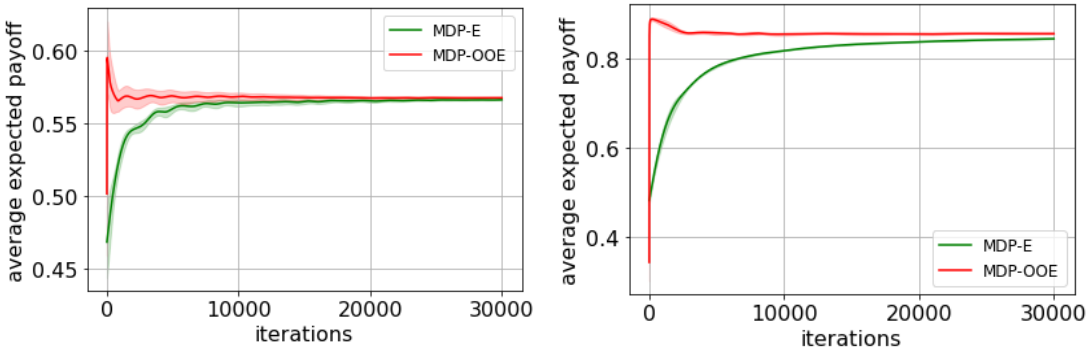
The Algorithm 21 also applies in situations where the adversary follows different learning dynamics such as Follow the Regularized Leader or linear MWU (Dinh et al., 2021). In these situations, Algorithm 21 requires adapting the constant parameter β so that the convergence result still holds. Since both the agent and the adversary converge to a NE, the NE is also the best fixed strategy in hindsight. Consequently, LRC-OMDP is also a no-regret algorithm where the regret bound depends on the convergence rate to the NE.

4.7 Experiment

In this section, we aim to demonstrate the effectiveness of our practical use algorithm MDP-OOE compared to the well-known MDP-E algorithm (Even-Dar et al., 2009).

We consider random games in which the entries of the transition matrix are first sampled from a uniform distribution $\mathbf{U}(0, 1)$, then follow the normalization. Similarly, the entries of the loss vectors from the adversary $\boldsymbol{l}_t$ are also sampled from a uniform distribution $\mathbf{U}(0, 1)$. Following Lemma 4.8, by fixing a small number of loss vectors L, we can bound the size of the Nash support of our games. Thus, we fix the number of loss vectors $L = 3$ and consider different games with the number of actions in each state in the set $[3, 100, 500]$. We then run MDP-E and MDP-OOE against the same opponent following a no-regret MWU algorithm and measure the average payoff of the two algorithms [4]. For each setting, we run 5 seeds where each seed considers an MWU adversary with a different starting strategy.

[4] W.l.o.g, we consider the payoff (i.e., -the loss) for the agent in our experiments so that the agent aims to maximize the payoff.



(A) 3 actions in each state

(B) 100 actions in each state

FIGURE 4.2: Performance comparisons in average payoff in random games

As we can see in Figure 4.2, MDP-OOE outperforms MDP-E in all games we consider. The difference in performance between the MDP-OOE and MDP-E becomes more significant when a larger action set is considered (See Figure 4.4 in the Appendix 4.10). Intuitively, since the performance of MDP-OOE only depends on the support size of the NE, a large size of the action set will not affect its performance. In contrast, a large action set will significantly affect the performance of MDP-E as it considers the whole action set in the strategy update. We observe a similar performance in other settings with a different number of loss vectors as shown in Figure 4.3 in the Appendix 4.10. The advantage of MDP-OOE in term of average payoff over MDP-E match our expectation as the support size of the NEs in these games are much smaller than the action set by design. Interestingly, even when the action set is small (i.e., $|A| = 3$), MDP-OOE still outperforms MDP-E in our experiments.

Note here that since we consider two-player zero-sum games and both the agent and the opponent follow no-regret algorithms, the average payoff of MDP-OOE and MDP-E will eventually converge to the value of the game, as shown in Figure 4.2.

4.8 Conclusion

In this chapter, we study a novel setting in Online Markov Decision Processes where the loss function is chosen by a non-oblivious strategic adversary who follows a no-external regret algorithm. In this new setting, we then revisit the MDP-E algorithm and provide a sublinear regret bound $\mathcal{O}(\sqrt{T\log(L)}+\tau^2\sqrt{T\log(|A|)})$. We suggest a new algorithm of MDP-OOE that achieves the policy regret of $\mathcal{O}(\sqrt{T\log(L)}+\tau^2\sqrt{Tk\log(k)})$ where the regret does not depend on the size of the strategy set $|A|$ but the effective strategy set k. Finally, in tackling the non-convergence property of no-regret algorithms, we provide the LRC-OMDP algorithm for the agent that leads to the first-known result of the last round convergence to a NE against the strategic adversary in OMDPs.

4.9 Appendix A: Detail Proofs

We provide the following lemmas and proposition:

Lemma 4.17 (Lemma 3.3 in Even-Dar et al. (2009))**.** *For all loss function $\boldsymbol{l}$ in $[0,1]$ and policies π, $Q_{\boldsymbol{l},\pi}(s,a) \leq 3\tau$.*

Lemma 4.18 (Lemma 1 from Neu et al. (2013))**.** *Consider uniformly ergodic OMDPs with mixing time τ with losses $\boldsymbol{l}_t \in [0,1]^{\boldsymbol{d}}$. Then, for any $T > 1$ and policy π with stationary distribution $\boldsymbol{d}_\pi$, it holds that*

$$\sum_{t=1}^{T} |\langle \boldsymbol{l}_t, \boldsymbol{d}_\pi - \boldsymbol{v}_t^\pi \rangle| \leq 2\tau + 2.$$

This lemma guarantees that the performance of a policy's stationary distribution is similar to the actual performance of the policy in the case of a fixed policy.

In the other case of non-fixed policy, the following lemma bound the performance of the policy's stationary distribution of algorithm A with the actual performance:

Lemma 4.19 (Lemma 5.2 in Even-Dar et al. (2009))**.** *Let $\pi_1, \pi_2, \ldots$ be the policies played by MDP-E algorithm $\mathcal{A}$ and let $\tilde{\boldsymbol{d}}_{\mathcal{A},t}$, $\tilde{\boldsymbol{d}}_{\pi_t} \in [0,1]^{|S|}$ be the stationary state distribution. Then,*

$$\|\tilde{\boldsymbol{d}}_{\mathcal{A},t} - \tilde{\boldsymbol{d}}_{\pi_t}\|_1 \leq 2\tau^2 \sqrt{\frac{\log(|A|)}{t}} + 2e^{-t/\tau}.$$

From the above lemma, since the policy's stationary distribution is a combination of stationary state distribution and the policy's action in each state, it is easy to show that:

$$\|\boldsymbol{v}_t - \boldsymbol{d}_{\pi_t}\|_1 \leq \|\tilde{\boldsymbol{d}}_{\mathcal{A},t} - \tilde{\boldsymbol{d}}_{\pi_t}\|_1 \leq 2\tau^2 \sqrt{\frac{\log(|A|)}{t}} + 2e^{-t/\tau}.$$

Proposition 4.20. *For the MWU algorithm (Freund and Schapire, 1999) with appropriate μ_t, we have*

$$R_T(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{l}_t(\pi_t)\right] - \mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{l}_t(\pi)\right] \leq M\sqrt{\frac{T\log(n)}{2}},$$

where $\|\boldsymbol{l}_t(.)\| \leq M$. Furthermore, the strategy $\boldsymbol{\pi}_t$ does not change quickly: $\|\boldsymbol{\pi}_t - \boldsymbol{\pi}_{t+1}\| \leq \sqrt{\frac{\log(n)}{t}}$.

Proof. For a fixed T, if the loss function satisfies $\boldsymbol{l}_t(.)\| \leq 1$ then by setting $\mu_t = \sqrt{\frac{8\log(n)}{T}}$, following Theorem 2.2 in Cesa-Bianchi and Lugosi (2006) we have

$$R_T(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{l}_t(\pi_t)\right] - \mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{l}_t(\pi)\right] \leq 1\sqrt{\frac{T\log(n)}{2}}. \tag{4.17}$$

Thus, in the case where $\boldsymbol{l}_t(.)\| \leq M$, by scaling up both sides by M in Equation (4.17) we have the first result of the Proposition 4.20. For the second part, follow the updating rule of MWU we have

$$\begin{aligned}\pi_{t+1}(i) - \pi_t(i) &= \pi_t(i)\left(\frac{\exp(-\mu_t \boldsymbol{l}_t(\boldsymbol{a}^i))}{\sum_{i=1}^{n} \boldsymbol{\pi}_t(i)\exp(-\mu_t \boldsymbol{l}_t(\boldsymbol{a}^i))} - 1\right) \\ &\approx \pi_t(i)\left(\frac{1-\mu_t \boldsymbol{l}_t(\boldsymbol{a}^i)}{1-\mu_t \boldsymbol{l}_t(\pi_t)} - 1\right) \qquad (4.18a) \\ &= \mu_t \pi_t(i)\frac{\boldsymbol{l}_t(\pi_t) - \boldsymbol{l}_t(\boldsymbol{a}^i)}{1-\mu_t \boldsymbol{l}_t(\pi_t)} = \mathcal{O}(\mu_t),\end{aligned}$$

where we use the approximation $e^x \approx 1 + x$ for small x in Equation (4.18a). Thus, the difference in two consecutive strategies π_t will be proportional to the learning rate μ_t, which is set to be $\mathcal{O}(\sqrt{\frac{\log(n)}{t}})$. A similar result can be found in Proposition 1 in Even-Dar et al. (2009). □

Theorem (Theorem 4.11). Suppose the agent uses MDP-OOE Algorithm 20 against a strategic adversary in our OMDPs setting, then the regret in Equation (4.1) can be bounded by:

$$R_T(\pi) = \mathcal{O}(\tau^2\sqrt{Tk\log(k)} + \sqrt{T\log(L)}).$$

Proof. First, we bound the difference between the true loss and the loss with respect to the policy's stationary distribution. Following the Algorithm 20, at the start of each time interval T_i (i.e., the time interval in which the effective strategy set does not change), the learning rate needs to restart to $\mathcal{O}(\sqrt{\log(i)/t_i})$, where i denotes the number of pure strategies in the effective strategy set in the time interval T_i and t_i is the relative position of the current round in that interval. Thus, following Lemma 5.2 in Even-Dar et al. (2009), in each time interval T_i, the difference between the true loss and the loss with respect to the policy's stationary distribution will be:

$$\begin{aligned}&\sum_{t=t_{i-1}+1}^{t_i} |\langle \boldsymbol{l}_t, \boldsymbol{v}_t - \boldsymbol{d}_{\pi_t}\rangle| \leq \sum_{t=t_{i-1}+1}^{t_i} \|\boldsymbol{v}_t - \boldsymbol{d}_{\pi_t}\|_1 \\ &\leq \sum_{t=1}^{T_i} 2\tau^2\sqrt{\frac{\log(i)}{t}} + 2e^{-t/\tau} \\ &\leq 4\tau^2\sqrt{T_i\log(i)} + 2(1+\tau).\end{aligned}$$

From this we have

$$\begin{aligned}&\sum_{t=1}^{T} |\langle \boldsymbol{l}_t, \boldsymbol{v}_t - \boldsymbol{d}_{\pi_t}\rangle| = \sum_{i=1}^{k}\sum_{t=t_{i-1}+1}^{t_i} |\langle \boldsymbol{l}_t, \boldsymbol{v}_t - \boldsymbol{d}_{\pi_t}\rangle| \\ &\leq \sum_{i=1}^{k}\left(4\tau^2\sqrt{T_i\log(i)} + 2(1+\tau)\right) \\ &\leq 4\tau^2\sqrt{Tk\log(k)} + 2k(1+\tau).\end{aligned}$$

Following Lemma 1 from (Neu et al., 2013), we also have:

$$\sum_{t=1}^{T} |\langle \boldsymbol{l}_t, \boldsymbol{d}_\pi - \boldsymbol{v}_t^\pi \rangle| \leq 2\tau + 2.$$

Thus the regret in Equation (4.1) can be bounded by:

$$\begin{aligned}
R_T(\pi) &\leq \left(\sum_{t=1}^{T} \langle \boldsymbol{d}_{\pi_t}, \boldsymbol{l}_t \rangle + \sum_{t=1}^{T} |\langle \boldsymbol{l}_t, \boldsymbol{v}_t - \boldsymbol{d}_{\pi_t} \rangle| \right) - \left(\sum_{t=1}^{T} \langle \boldsymbol{l}_t^\pi, \boldsymbol{d}_\pi \rangle - \sum_{t=1}^{T} |\langle \boldsymbol{l}_t, \boldsymbol{d}_\pi - \boldsymbol{v}_t^\pi \rangle| \right) \\
&= \left(\sum_{t=1}^{T} \langle \boldsymbol{d}_{\pi_t}, \boldsymbol{l}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{l}_t^\pi, \boldsymbol{d}_\pi \rangle \right) + \sum_{t=1}^{T} |\langle \boldsymbol{l}_t, \boldsymbol{v}_t - \boldsymbol{d}_{\pi_t} \rangle| + \sum_{t=1}^{T} |\langle \boldsymbol{l}_t, \boldsymbol{d}_\pi - \boldsymbol{v}_t^\pi \rangle| \\
&\leq 3\tau \left(\sqrt{2Tk \log(k)} + \frac{k \log(k)}{8} \right) + \frac{\sqrt{T \log(L)}}{\sqrt{2}} + 4\tau^2 \sqrt{Tk \log(k)} + 2k(1+\tau) + 2\tau + 2 \\
&= \mathcal{O}(\tau^2 \sqrt{Tk \log(k)} + \sqrt{T \log(L)}).
\end{aligned}$$

The proof is complete. □

Theorem (Theorem 4.12). Suppose the agent only accesses ϵ-best response in each iteration when following Algorithm 20. If the adversary follows a no-external regret algorithm then the average strategy of the agent and the adversary will converge to ϵ-Nash equilibrium. Furthermore, the algorithm has ϵ-regret.

Proof. Suppose that the player uses the Multiplicative Weights Update in Algorithm 20 with ϵ-best response. Let $T_1, T_2, \ldots, T_k$ be the time window that the players do not add up a new strategy. Since we have a finite set of strategies A then k is finite. Furthermore,

$$\sum_{i=1}^{k} T_k = T.$$

In a time window T_i, the regret with respect to the best strategy in the set of strategy at time T_i is:

$$\sum_{t=\bar{T}_i}^{\bar{T}_{i+1}} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t} \rangle - \min_{\pi \in A_{\bar{T}_i+1}} \sum_{t=|\bar{T}_i|}^{\bar{T}_{i+1}} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_\pi \rangle \leq 3\tau \left(\sqrt{2T_i \log(i)} + \frac{\log(i)}{8} \right), \tag{4.19}$$

where $\bar{T}_i = \sum_{j=1}^{i-1} T_j$. Since in the time window T_i, the ϵ-best response strategy stays in $\Pi_{\bar{T}_i+1}$ and therefore we have

$$\min_{\pi \in A_{\bar{T}_i+1}} \sum_{t=|\bar{T}_i|}^{\bar{T}_{i+1}} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_\pi \rangle - \min_{\pi \in \Pi} \sum_{t=|\bar{T}_i|}^{\bar{T}_{i+1}} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_\pi \rangle \leq \epsilon T_i.$$

Then, from the Equation (4.19) we have

$$\sum_{t=\bar{T}_i}^{\bar{T}_{i+1}} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t} \rangle - \min_{\pi \in \Pi} \sum_{t=|\bar{T}_i|}^{\bar{T}_{i+1}} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi} \rangle \leq 3\tau \left(\sqrt{2T_i \log(i)} + \frac{\log(i)}{8} \right) + \epsilon T_i. \tag{4.20}$$

Sum up the Equation (4.20) for $i = 1, \dots k$ we have

$$\begin{aligned}
&\sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t} \rangle - \sum_{i=1}^{k} \min_{\pi \in \Pi} \sum_{t=|\bar{T}_i|}^{\bar{T}_{i+1}} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi} \rangle \leq \sum_{i=1}^{k} 3\tau \left(\sqrt{2T_i \log(i)} + \frac{\log(i)}{8} \right) + \epsilon T_i \\
&\implies \sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t} \rangle - \min_{\pi \in \Pi} \sum_{i=1}^{k} \sum_{t=|\bar{T}_i|}^{\bar{T}_{i+1}} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi} \rangle \leq \epsilon T + \sum_{i=1}^{k} 3\tau \left(\sqrt{2T_i \log(i)} + \frac{\log(i)}{8} \right) \qquad (4.21a) \\
&\implies \sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t} \rangle - \min_{\pi \in \Pi} \sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi} \rangle \leq \epsilon T + \sum_{i=1}^{k} 3\tau \left(\sqrt{2T_i \log(i)} + \frac{\log(i)}{8} \right) \\
&\implies \sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t} \rangle - \min_{\pi \in \Pi} \sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi} \rangle \leq \epsilon T + 3\tau \left(\sqrt{2Tk \log(k)} + \frac{k \log(k)}{8} \right). \qquad (4.21b)
\end{aligned}$$

Inequality (4.21a) is due to $\sum \min \leq \min \sum$. Inequality (4.21b) comes from Cauchy-Schwarz inequality and Stirling' approximation. Using Inequality (4.21b), we have

$$\min_{\pi \in \Pi} \langle \bar{\boldsymbol{l}}, \boldsymbol{d}_{\pi} \rangle \geq \frac{1}{T} \sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t} \rangle - 3\tau \left(\sqrt{\frac{2k \log(k)}{T}} + \frac{k \log(k)}{8T} \right) - \epsilon. \tag{4.22}$$

Since the adversary follows a no-regret algorithm, we have

$$\begin{aligned}
&\max_{\boldsymbol{l} \in \Delta_L} \sum_{t=1}^{T} \langle \boldsymbol{l}, \boldsymbol{d}_{\pi_t} \rangle - \sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t} \rangle \leq \sqrt{\frac{T}{2}} \sqrt{\log(L)} \\
&\implies \max_{\boldsymbol{l} \in \Delta_L} \sum_{t=1}^{T} \langle \boldsymbol{l}, \bar{\boldsymbol{d}}_{\pi} \rangle \leq \frac{1}{T} \sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t} \rangle + \sqrt{\frac{\log(L)}{2T}}.
\end{aligned} \tag{4.23}$$

Using the Inequalities (4.22) and (4.23) we have

$$\begin{aligned}
\langle \bar{\boldsymbol{l}}, \bar{\boldsymbol{d}}_{\pi} \rangle &\geq \min_{\pi \in \Pi} \langle \bar{\boldsymbol{l}}, \boldsymbol{d}_{\pi} \rangle \geq \frac{1}{T} \sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t} \rangle - 3\tau \left(\sqrt{\frac{2k \log(k)}{T}} + \frac{k \log(k)}{8T} \right) - \epsilon \\
&\geq \max_{\boldsymbol{l} \in \Delta_L} \sum_{t=1}^{T} \langle \boldsymbol{l}, \bar{\boldsymbol{d}}_{\pi} \rangle - \sqrt{\frac{\log(L)}{2T}} - 3\tau \left(\sqrt{\frac{2k \log(k)}{T}} + \frac{k \log(k)}{8T} \right) - \epsilon.
\end{aligned}$$

Similarly, we also have:

$$\begin{aligned}\langle \bar{\boldsymbol{l}}, \bar{\boldsymbol{d}}_\pi \rangle &\leq \max_{\boldsymbol{l}\in\Delta_L} \sum_{t=1}^{T} \langle \boldsymbol{l}, \bar{\boldsymbol{d}}_\pi \rangle \leq \frac{1}{T} \sum_{t=1}^{T} \langle \boldsymbol{l}_t^{\pi_t}, \boldsymbol{d}_{\pi_t} \rangle + \sqrt{\frac{\log(L)}{2T}} \\ &\leq \min_{\pi\in\Pi} \langle \bar{\boldsymbol{l}}, \boldsymbol{d}_\pi \rangle + 3\tau \left(\sqrt{\frac{2k\log(k)}{T}} + \frac{k\log(k)}{8T} \right) + \epsilon.\end{aligned}$$

Take the limit $T \to \infty$, we then have:

$$\max_{\boldsymbol{l}\in\Delta_L} \sum_{t=1}^{T} \langle \boldsymbol{l}, \bar{\boldsymbol{d}}_\pi \rangle - \epsilon \leq \langle \bar{\boldsymbol{l}}, \bar{\boldsymbol{d}}_\pi \rangle \leq \min_{\pi\in\Pi} \langle \bar{\boldsymbol{l}}, \boldsymbol{d}_\pi \rangle + \epsilon.$$

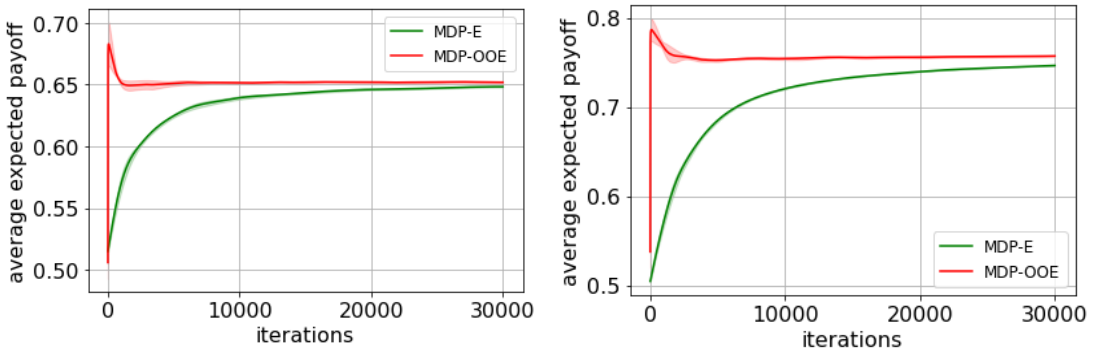
Thus $(\bar{\boldsymbol{l}}, \bar{\boldsymbol{d}}_\pi)$ is the ϵ-Nash equilibrium of the game. □

4.10 Appendix B: Additional Experimental Results

We provide further experiment results to demonstrate the performance of MDP-OOE and MDP-E.

In Figure 4.3, by considering the different number of loss vectors ($L = 7$), we test whether the performance difference between MDP-OOE and MDP-E is consistent with regard to the number of loss vectors. As we can see in Figure 4.3, MDP-OOE also outperforms MDP-E with the number of loss functions $L = 7$. The result further validates the advantage of MDP-OOE over MDP-E in the setting of a small support size of the NE.

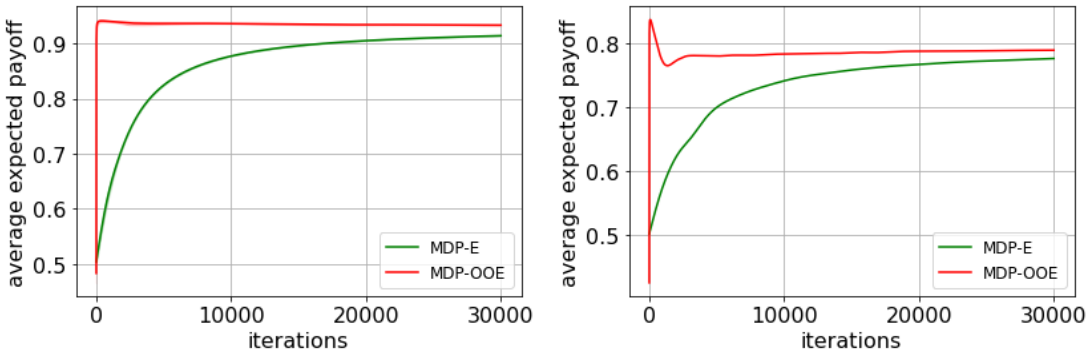
In Figure 4.4, we consider a larger set of agent's actions in each state ($A = 500$). As we can see in Figure 4.4, the difference in performance between MDP-OOE and MDP-E becomes more significant when a larger action set is considered in both cases when $L = 3$ and $L = 7$, as expected by our theoretical results.



(A) 7 actions in each state

(B) 100 actions in each state

FIGURE 4.3: Performance comparisons in average payoff in random games with $L = 7$



(A) 500 actions in each state with opponent's pure strategies $L = 3$

(B) 500 actions in each state with opponent's pure strategies $L = 7$

FIGURE 4.4: Performance comparisons in average payoff in random games

Chapter 5

Conclusion and Future Work

The contributions of this thesis are in three different yet connected settings: two-player zero-sum games, online linear optimization, and online Markov decision processes. In each setting, we propose and develop new algorithms with theoretical guarantees against the strategic adversary. These theoretical properties are also backed up by empirical performances.

5.1 Two-player Zero-sum games

In the two-player zero-sum games setting, we create an algorithm called LRCA that achieves last round convergence to a minimax equilibrium and no-dynamic regret against the strategic adversary. Our research demonstrates that LRCA is effective against a wide range of commonly used no-regret algorithms that the adversary may employ such as the multiplicative weights update algorithm and the general follow-the-regularized-leader algorithm. Additionally, we find that LRCA is efficient against any no-regret algorithms that satisfy the "stability" property.

The outcomes presented in this study on two-player zero-sum games setting provide numerous encouraging avenues for future study. Firstly, one drawback of the LRCA algorithm is that it requires NE information in its update to achieve the convergence guarantee. Even though we relax this requirement to allow ϵ-NE and suggest a way to achieve this information against the strategic adversary, the NE requirement in the strategy update limits the application of LRCA in some situations. Thus, in the future, we hope to develop a new algorithm with similar dynamic regret and last round convergence like LRCA while the update strategy does not require the knowledge of NE.

Secondly, it is also desirable to extend our findings in zero-sum games to general-sum games. Even though many applications can be cast as completely competitive games (i.e., zero-sum games), there are a large set of problems in which they can only be

formulated as a general-sum game. Therefore, the new understanding of the last round convergence result in general-sum games will broaden the current literature, while the focus is currently on the average coarse correlated equilibrium convergence (Cesa-Bianchi and Lugosi, 2006). Note that the analysis of last round convergence in zero-sum games relies heavily on John von Neumann's minimax theorem, which does not hold in the general-sum games setting. Thus, new analyses and techniques need to be developed to tackle this challenging problem.

Thirdly, given the last round convergence result in the two-player setting, one should wonder if the same can hold true for multiple-player games. Put differently, given a game such as every other player is strategic adversaries (following a no-regret algorithm), does there exist an algorithm for the agent to exploit the extra knowledge in the game? How many agents need to collaborate in the game to achieve the last round convergence to the NE of all players? Answering these questions will stretch the edge of the understanding in learning against the strategic adversary.

5.2 Online Linear Optimization

In the online linear optimization setting, first, we develop a novel algorithm called OSO, which achieves the no-regret property and can exploit strategic adversaries during game play by conducting no-regret analysis within the DO framework. Secondly, we introduce AFTRL, a method that can exploit the strategic adversary to achieve $O(1)$ external regret or $O(1)$ forward regret while still maintaining the state-of-the-art regret bound of $O\left(\sqrt{\sum_{t=1}^{T} \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_q^2}\right)$ in the worst-case scenario. Thirdly, we explore the idea of the (A,B)-Prod algorithm and suggest a new algorithm called Prod-BR, which achieves a stronger performance guarantee in our setting. Specifically, Prod-BR achieves $O(\sqrt{T})$ dynamic regret against no-external regret adversary while maintaining $O(\sqrt{T\log(T)})$ external regret in the worst-case scenario. Fourthly, in a special case of AFTRL with entropy regularizer, AMWU, we prove that this new algorithm will lead to last round convergence in two-player zero-sum games, making it an efficient game-solver in many practical applications.

To enhance the contributions of our work in this setting, we aim to pursue two promising avenues for improvement in future studies. Firstly, while AMWU has shown great success in experiments, its theoretical convergence rate is not known yet. Thus, in the future, we want to extend the analysis of AMWU to provide a concrete convergence rate for AMWU. We conjecture that AMWU will have a linear last round convergence rate, which explains its superior performance compared to MWU and OMWU.

Secondly, the success of OSO relies on the assumption that the size of the effective strategy set k is small. While in practice, we empirically show that there exists a linear relationship between k and the support size of NE, thus explaining the success of OSO

in many practical applications. However, in theory, we provide a counter example validating that in general, k may not depend on the support size of the NE. Understanding the relationship between k and the support of NE, at least in some classes of games, is a fundamental step to advance the development of the DO/PSRO line of work, in which they have shown tremendous success in applications but still lacking concrete theoretical back-up.

5.3 Online Markov Decision Processes

In the setting of Online Markov Decision Processes, we demonstrate the effectiveness of the MDP-E algorithm against strategic adversaries, achieving a policy regret bound of $\mathcal{O}(\sqrt{T\log(L)} + \tau^2\sqrt{T\log(|A|)})$. Furthermore, we show that the average strategies of agents will converge to a NE of the game. In situations where the NE support size is small, we introduce a new no-regret algorithm, called MDP-OOE, that maintains the no-regret property and has a policy regret bound of $\mathcal{O}(\tau^2\sqrt{Tk\log(k)} + \sqrt{T\log(L)})$ against the strategic adversary. MDP-OOE combines the benefits of both the Double Oracle and MDP-E algorithms, enabling it to play games with large action spaces. Finally, we propose the LRC-OMDP algorithm to achieve last round convergence guarantees against no-external regret algorithms. Specifically, when the adversary follows a no-external regret algorithm, LRC-OMDP guarantees last round convergence to a NE, making it the first such result for OMDPs.

The findings presented in this study on OMDPs offer several promising directions for future research. Firstly, while MDP-OOE demonstrates superior performance compared to MDP-E in both theoretical and experimental settings, its requirement to compute best response oracles in each iteration results in increasing time complexity. Despite Theorem 4.12 offering an alternative to utilizing ϵ-best response, there remains an opportunity to enhance the efficiency of the MDP-OOE algorithm in relation to the best response oracle. Additionally, a key open question that MDP-OOE shares with the OSO/DO method pertains to the precise relationship between the size of the effective strategy set and the support size of NE. Thus, further investigation into this relationship is pivotal for advancing the development of both OSO and MDP-OOE algorithms.

Secondly, the LRC-OMDP algorithm offers the first instance of last round convergence to a NE against a strategic adversary in OMDPs. However, this achievement hinges on the assumption of prior knowledge of the agent's NE, a common assumption in literature (Conitzer and Sandholm, 2007; Chakraborty and Stone, 2014). Relaxing this assumption could extend the practical applications of the LRC-OMDP algorithm. Additionally, this thesis does not provide a derivation of the convergence rate of LRC-OMDP.

Further research, both theoretical and empirical, is therefore necessary to gain a comprehensive understanding of the efficiency of LRC-OMDP in practical games, which will facilitate further improvement of the algorithm.

Finally, our paper introduces a novel approach to address the hardness of playing against a non-oblivious adversary. While we emphasize that the strategic adversary covers a wide range of important practical situations and thus deserves further attention from the research community, we acknowledge the existence of other important types of adversaries in the literature, such as the m-memory bounded adversary (Arora et al., 2012a). Going forward, our ambitious goal is to develop a unified algorithm that achieves both no-dynamic regret and last round convergence against a broad class of adversaries.

Bibliography

Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellért Weisz. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, pages 3692–3702, 2019.

Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *21st Annual Conference on Learning Theory*, pages 263–273, 2008.

Ilan Adler, Constantinos Daskalakis, and Christos H Papadimitriou. A note on strictly competitive games. In *Internet and Network Economics: 5th International Workshop, Rome, Italy, December 14-18, 2009. Proceedings 5*, pages 471–474. Springer, 2009.

Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. *arXiv preprint arXiv:1206.6400*, 2012a.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012b.

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

James Bailey and Georgios Piliouras. Fast and furious learning in zero-sum games: vanishing regret with non-vanishing step sizes. In *Advances in Neural Information Processing Systems*, pages 12977–12987, 2019.

James P Bailey and Georgios Piliouras. Multiplicative weights update in zero-sum games. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 321–338, 2018.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Ulrich Berger. Fictitious play in 2× n games. *Journal of Economic Theory*, 120(2): 139–154, 2005.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations research*, 63(5):1227–1244, 2015.

Nicholas Bishop, Le Cong Dinh, and Long Tran-Thanh. How to guide a non-cooperative learner to cooperate: Exploiting no-regret algorithms in system design. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, pages 1464–1466, 2021.

HF Bohnenblust, S Karlin, and LS Shapley. Solutions of discrete, two-person games. *Contributions to the Theory of Games*, 1:51–72, 1950.

Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold'em poker is solved. *Science*, 347(6218):145–149, 2015.

Felix Brandt, Felix Fischer, and Paul Harrenstein. On the rate of convergence of fictitious play. In *International Symposium on Algorithmic Game Theory*, pages 102–113. Springer, 2010.

George W Brown. Some notes on computation of games solutions. Technical report, RAND Corporation Santa Monica, California, 1949.

George W Brown. Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13(1):374–376, 1951.

Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pages 1283–1294. PMLR, 2020.

Yang Cai, Ozan Candogan, Constantinos Daskalakis, and Christos Papadimitriou. Zero-sum polymatrix games: A generalization of minmax. *Mathematics of Operations Research*, 41(2):648–655, 2016.

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

Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66:321–352, 2007.

Doran Chakraborty and Peter Stone. Multiagent learning in the presence of memory-bounded agents. *Autonomous agents and multi-agent systems*, 28(2):182–213, 2014.

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Non-stationary reinforcement learning: The blessing of (more) optimism. *Available at SSRN 3397818*, 2019.

Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. In *Conference on Learning Theory*, pages 6.1–6.20. JMLR Workshop and Conference Proceedings, 2012.

Vincent Conitzer and Tuomas Sandholm. Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1-2):23–43, 2007.

Wojciech Marian Czarnecki, Gauthier Gidel, Brendan Tracey, Karl Tuyls, Shayegan Omidshafiei, David Balduzzi, and Max Jaderberg. Real world games look like spinning tops. *arXiv preprint arXiv:2004.09468*, 2020.

Constantinos Daskalakis and Qinxuan Pan. A counter-example to karlin's strong conjecture for fictitious play. In *55th Annual Symposium on Foundations of Computer Science*, pages 11–20. IEEE, 2014.

Constantinos Daskalakis and Ioannis Panageas. Last-iterate convergence: Zero-sum games and constrained min-max optimization. *10th Innovations in Theoretical Computer Science*, pages 27:1–27:18, 2019.

Constantinos Daskalakis, Alan Deckelbaum, and Anthony Kim. Near-optimal no-regret algorithms for zero-sum games. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 235–254. SIAM, 2011.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. In *International Conference on Learning Representations*, 2018.

Steven De Rooij, Tim Van Erven, Peter D Grünwald, and Wouter M Koolen. Follow the leader if you can, hedge if you must. *The Journal of Machine Learning Research*, 15(1):1281–1316, 2014.

Xiaotie Deng, Yuhao Li, David Henry Mguni, Jun Wang, and Yaodong Yang. On the complexity of computing markov perfect equilibrium in general-sum stochastic games. *arXiv preprint arXiv:2109.01795*, 2021.

Yuan Deng, Jon Schneider, and Balasubramanian Sivan. Strategizing against no-regret learners. *Advances in neural information processing systems*, 33, 2019.

Travis Dick, Andras Gyorgy, and Csaba Szepesvari. Online learning in markov decision processes with changing cost sequences. In *International Conference on Machine Learning*, pages 512–520. PMLR, 2014.

Le Cong Dinh. Online learning against strategic adversary. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 1841–1842, 2022.

Le Cong Dinh, Tri-Dung Nguyen, Alain B Zemhoho, and Long Tran-Thanh. Last round convergence and no-dynamic regret in asymmetric repeated games. In *Algorithmic Learning Theory*, pages 553–577. PMLR, 2021.

Le Cong Dinh, Stephen Marcus McAleer, Zheng Tian, Nicolas Perez-Nieves, Oliver Slumbers, David Henry Mguni, Jun Wang, Haitham Bou Ammar, and Yaodong Yang. Online double oracle. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=rrMK6hYNSx.

Eyal Even-Dar, Michael Kearns, Yishay Mansour, and Jennifer Wortman. Regret to the best vs. regret to the average. *Machine Learning*, 72:21–37, 2008.

Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. Online markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

Jerzy Filar and Koos Vrieze. Applications and special classes of stochastic games. In *Competitive Markov Decision Processes*, pages 301–341, New York, 1997. Springer.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Oded Galor. *Discrete dynamical systems*. Springer Science & Business Media, 2007.

Peng Guan, Maxim Raginsky, Rebecca Willett, and Daphney-Stavroula Zois. Regret minimization algorithms for single-controller zero-sum stochastic games. In *55th Conference on Decision and Control (CDC)*, pages 7075–7080. IEEE, 2016.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2015.

Ruitong Huang, Tor Lattimore, András György, and Csaba Szepesvári. Following the leader and fast rates in linear prediction: Curved constraint sets and other regularities. *Advances in Neural Information Processing Systems*, 29, 2016.

Johan Jonasson et al. On the optimal strategy in a random game. *Electronic Communications in Probability*, 9:132–139, 2004.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in neural information processing systems*, pages 4190–4203, 2017.

Guillaume J Laurent, Laëtitia Matignon, Le Fort-Piat, et al. The world of independent learners is not markovian. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 15(1):55–64, 2011.

David S Leslie and Edmund J Collins. Generalised weakened fictitious play. *Games and Economic Behavior*, 56(2):285–298, 2006.

David S Leslie, Steven Perkins, and Zibo Xu. Best-response dynamics in zero-sum stochastic games. *Journal of Economic Theory*, 189:105095, 2020.

Tianyi Lin, Zhengyuan Zhou, Panayotis Mertikopoulos, and Michael Jordan. Finite-time last-iterate convergence for multi-agent learning in games. In *International Conference on Machine Learning*, pages 6161–6171. PMLR, 2020.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

Xiangyu Liu, Hangtian Jia, Ying Wen, Yaodong Yang, Yujing Hu, Yingfeng Chen, Changjie Fan, and Zhipeng Hu. Unifying behavioral and response diversity for open-ended learning in zero-sum games. *arXiv preprint arXiv:2106.04958*, 2021.

Stephen McAleer, John Lanier, Roy Fox, and Pierre Baldi. Pipeline PSRO: A scalable approach for finding approximate nash equilibria in large games. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *AISTATS*, pages 525–533, 2011.

H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 536–543, 2003.

Ruta Mehta, Ioannis Panageas, Georgios Piliouras, Prasad Tetali, and Vijay V Vazirani. Mutation, sexual reproduction and survival in dynamic environments. In *8th Innovations in Theoretical Computer Science Conference, ITCS*, 2017.

Panayotis Mertikopoulos, Christos Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2703–2717. SIAM, 2018.

Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *7th International Conference on Learning Representations*, pages 1–23, 2019.

Dragoslav S Mitrinovic and Petar M Vasic. *Analytic inequalities*, volume 61. Springer, 1970.

Dov Monderer and Lloyd S Shapley. Fictitious play property for games with identical interests. *Journal of economic theory*, 68(1):258–265, 1996.

John H Nachbar. "evolutionary" selection dynamics in games: Convergence and limit properties. *International journal of game theory*, 19:59–89, 1990.

John F Nash Jr. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. *Wiley-Interscience*, 1983.

Gergely Neu and Julia Olkhovskaya. Online learning in mdps with linear function approximation and bandit feedback. *Advances in Neural Information Processing Systems*, 34:10407–10417, 2021.

Gergely Neu, Andras Antos, András György, and Csaba Szepesvári. Online markov decision processes under bandit feedback. In *NeurIPS*, pages 1804–1812, 2010.

Gergely Neu, András György, Csaba Szepesvári, and András Antos. Online markov decision processes under bandit feedback. *IEEE Transactions on Automatic Control*, 59(3):676–691, 2013.

J v Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

Nicolas Perez-Nieves, Yaodong Yang, Oliver Slumbers, David H Mguni, Ying Wen, and Jun Wang. Modelling behavioural diversity for learning in open-ended games. In *International Conference on Machine Learning*, pages 8514–8524. PMLR, 2021.

Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Conference on Learning Theory*, pages 993–1019. PMLR, 2013a.

Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013b.

Julia Robinson. An iterative method of solving a game. *Annals of mathematics*, pages 296–301, 1951.

Aviv Rosenberg and Yishay Mansour. Online stochastic shortest path with bandit feedback and unknown transition function. In *Advances in Neural Information Processing Systems*, pages 2212–2221, 2019.

Tim Roughgarden. Algorithmic game theory. *Communications of the ACM*, 53(7):78–86, 2010.

Ankan Saha, Prateek Jain, and Ambuj Tewari. The interplay between stability and regret in online learning. *arXiv preprint arXiv:1211.6158*, 2012.

Amir Sani, Gergely Neu, and Alessandro Lazaric. Exploiting easy data in online optimization. In *Advances in Neural Information Processing Systems*, pages 810–818, 2014.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Shai Shalev-Shwartz and Yoram Singer. Convex repeated games and fenchel duality. *Advances in neural information processing systems*, 19, 2006.

Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39 (10):1095–1100, 1953.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, Massachusetts, 2018.

Yi Tian, Yuanhao Wang, Tiancheng Yu, and Suvrit Sra. Online learning in unknown markov games. In *International conference on machine learning*, pages 10279–10288. PMLR, 2021.

Eric Van Damme. *Stability and perfection of Nash equilibria*, volume 339. Springer, 1991.

Ben Van der Genugten. A weakened form of fictitious play in two-person zero-sum games. *International Game Theory Review*, 2(04):307–328, 2000.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Chen-Yu Wei, Yi-Te Hong, and Chi-Jen Lu. Online reinforcement learning in stochastic games. *Advances in Neural Information Processing Systems*, 30, 2017.

Chen-Yu Wei, Chung-Wei Lee, Mengxiao Zhang, and Haipeng Luo. Linear last-iterate convergence in constrained saddle-point optimization. In *International Conference on Learning Representations*, 2020.

Yaodong Yang and Jun Wang. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv preprint arXiv:2011.00583*, 2020.

Yaodong Yang, Jun Luo, Ying Wen, Oliver Slumbers, Daniel Graves, Haitham Bou Ammar, Jun Wang, and Matthew E Taylor. Diverse auto-curriculum is critical for successful real-world multiagent learning systems. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 51–56, 2021.

Jia Yuan Yu, Shie Mannor, and Nahum Shimkin. Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research*, 34(3):737–757, 2009.

Alexander Zimin and Gergely Neu. Online learning in episodic markovian decision processes by relative entropy policy search. In *Advances in neural information processing systems*, pages 1583–1591, 2013.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. *Advances in neural information processing systems*, 20:1729–1736, 2007.