Sample-Efficient Regret-Minimizing Double Oracle in Two-Player Zero-Sum Extensive-Form Games

Xiaohang Tang XIAOHANG.TANG.20@UCL.AC.UK
Department of Statistical Science, University College London.

Chiyuan Wang WANG2021@STU.PKU.EDU.CN
YuanPei College, Peking University.

Chengdong Ma MCD1619@OUTLOOK.COM
Institute for Artificial Intelligence, Peking University.

Ilija Bogunovic I.BOGUNOVIC@UCL.AC.UK
Department of Electronic and Electrical Engineering, University College London.

Stephen McAleer SMCALEER@CS.CMU.EDU
Carnegie Mellon University.

Yaodong Yang[†] YAODONG.YANG@PKU.EDU.CN
Institute for Artificial Intelligence, Peking University.
[†] *Corresponding author.*

Editor: Csaba Szepesvari

Abstract

Extensive-Form Game (EFG) represents a fundamental model for analyzing sequential interactions among multiple agents and the primary challenge to solve it lies in mitigating sample complexity. Existing research indicates that Double Oracle (DO) can reduce the sample complexity dependence on the information set number $|\mathcal{S}|$ to a restricted game size X in solving two-player zero-sum EFGs. This is attributed to the early convergence of full-game Nash Equilibrium (NE) through iteratively solving restricted games. However, we prove that the most effective DO method, Extensive-Form Double Oracle (XDO), exhibits *exponential* sample complexity of X in the worst case, due to its exponentially increasing restricted game expansion frequency. Here we introduce Adaptive Double Oracle (AdaDO) to significantly alleviate sample complexity to *polynomial* by deploying the optimal expansion frequency. Furthermore, to comprehensively study the principles and influencing factors underlying sample complexity of DO, we introduce a novel theoretical framework Regret-Minimizing Double Oracle (RMDO) to provide directions for designing efficient DO algorithms. Empirical results demonstrate that AdaDO attains the superior approximation of NE with less sample complexity than the strong baselines including Linear CFR, MCCFR and existing DO. Importantly, combining RMDO with warm starting and stochastic regret minimization further improves convergence rate and scalability, thereby paving the way for addressing complex multi-agent tasks.

Keywords: Regret Minimization, Double Oracle, Sample Complexity, Game Theory, Nash Equilibrium

1 Introduction

Extensive-Form Games (EFGs) are widely studied fundamental models in game theory (Hart, 1992; Reny, 1992; Von Neumann and Morgenstern, 2007; Ritzberger et al., 2016). Solving for a Nash equilibrium (NE) is critical for addressing sequential multi-agent decision-making problems, such as board games and auction bidding, by formulating it as a EFG tree. Existing methods have made strides in solving scalable two-player zero-sum EFGs, based on Counterfactual Regret Minimization (CFR) (Zinkevich et al., 2007; Tammelin, 2014; Lanctot et al., 2009; Schmid et al., 2019; Farina et al., 2020; Brown and Sandholm, 2017, 2019b). CFR aims to approximate the NE by visiting all nodes (information sets) within the EFG tree to compute counterfactual regrets and update strategies. Thus, the sample complexity of CFR heavily depends on the number of all information sets (infoset). As the scale of the game expands, the sample complexity of CFR methods becomes prohibitively high.

To efficiently solve EFGs, prior works have attempted to introduce Double Oracle (DO) paradigm (McMahan et al., 2003; Bosansky et al., 2014; Lanctot et al., 2017; McAleer et al., 2021; Dinh et al., 2022), which has sample complexity to reach a specific approximate NE independent on the information sets. The core idea of DO is to approximate NE by resolving an expanding restricted game where players can only choose actions from a subset of the action space. The restricted game is expanded by adding the original game's Best Response (BR) actions to the NE in the restricted game (meta-NE). Since DO's restricted game typically halts its growth in the early stages before reaching the original game, DO can reduce the sample complexity dependence from $|\mathcal{S}|$ to the final restricted game size X. The empirical results in previous works (McAleer et al., 2021; Qin et al., 2022; Tang et al., 2023) reveal that the DO converges faster to a less exploitable strategy than the regret minimization algorithm including strong baseline CFR+, and thus has become a state-of-the-art algorithm for solving research EFGs. However, the theoretical questions of DO methods in extensive-form games, particularly concerning convergence speed, expected iterations, and sample complexity, remain insufficiently formally explored (Bosansky et al., 2014). This motivates the core question we aim to answer:

Q: When designing extensive-form DO algorithms, what factors cause high sample complexity and how can they be avoided?

To understand the causes of sample complexity of the DO algorithms, we first propose a unified framework, **Regret-Minimizing Double Oracle (RMDO)**. By setting different expansion rules of restricted game, RMDO generalizes existing DO methods including DO (Bosansky et al., 2014), XDO (McAleer et al., 2021) and ODO (Dinh et al., 2022). We derive the sample complexity for RMDO framework to reach an ϵ-NE, and formally identify that the most influential factor to a high complexity is an inappropriate way of expanding the restricted game. Specifically, the most empirically effective DO method for EFGs, namely XDO, employs an exponentially decreasing exploitability threshold to expand the restricted game. This results in that XDO needs exponentially increasing iterations with the number of different restricted games, denoted by k, to reach an ϵ-NE. XDO thus has an *exponential* sample complexity in k, where k is only bounded by the final restricted game size X.

Table 1: Main theoretical results of sample complexities for RMDO instances to reach ϵ-NE in extensive-form games. We categorize the algorithms into reaching NE by regret minimization (RM) and stochastic regret minimization (SRM). Denote $|\mathcal{S}|$ as the number of information sets, $X \leq |\mathcal{S}|$ as the final restricted game size, and k as the number of different restricted games. Here we present the degree of dominating terms k, X, and $|\mathcal{S}|$ in the sample complexities. Exp. indicates that the complexity is exponential in the factor.

Reach NE via	Algorithm	Sample Complexity	k	X	$\lvert\mathcal{S}\rvert$
RM	CFR (Zinkevich et al., 2007)	$\tilde{\mathcal{O}}(\lvert\mathcal{A}\rvert\lvert\mathcal{S}\rvert^3/\epsilon^2)$	0	0	3
	XODO (Dinh et al., 2022)	$\tilde{\mathcal{O}}\Big(k^2X^3/\epsilon^2 + k^2X^2\lvert\mathcal{S}\rvert/\epsilon^2\Big)$.	2	3	1
	XDO (McAleer et al., 2021)	$\tilde{\mathcal{O}}(k\lvert\mathcal{A}\rvert X^3/\epsilon^2 + 4^k\lvert\mathcal{A}\rvert X^3/\epsilon_0^2)$	Exp.	3	0
	PDO (ours)	$\tilde{\mathcal{O}}(k\lvert\mathcal{A}\rvert\lvert\mathcal{S}\rvert X^2/\epsilon^2 + k\lvert\mathcal{A}\rvert X^3/\epsilon^2)$	1	3	1
	AdaDO (ours)	$\tilde{\mathcal{O}}(2k\sqrt{\lvert\mathcal{A}\rvert\lvert\mathcal{S}\rvert}X^{\frac{3}{2}}/\epsilon + k\lvert\mathcal{A}\rvert X^3/\epsilon^2)$	1	3	0.5
SRM	MCCFR (Lanctot et al., 2009)	$\tilde{\mathcal{O}}(H\lvert\mathcal{A}\rvert\lvert\mathcal{S}\rvert^2/\epsilon^2)$	0	0	2
	SPDO (ours)	$\tilde{\mathcal{O}}(k\lvert\mathcal{A}\rvert\lvert\mathcal{S}\rvert X^2/\epsilon^2)$,	1	2	1
	SADO (ours)	$\tilde{\mathcal{O}}(\sqrt{\lvert\mathcal{A}\rvert\lvert\mathcal{S}\rvert}kX/\epsilon + k\lvert\mathcal{A}\rvert HX^2/\epsilon^2)$,	1	2	0.5

Based on the theoretical insights of RMDO, we propose a novel instance of RMDO called **Adaptive Double Oracle (AdaDO)**, which employs the theoretically optimal expanding rule to alleviate concerns of exponential sample complexity. AdaDO exhibits only *polynomial* sample complexity to reach ϵ-NE, which matches the complexity lower bound of RMDO framework, and thus is more sample efficient than XDO.

Furthermore, apart from the expanding rule of restricted game, the sample complexity of RMDO is also dependent on the number of different restricted games k and the regret-minimizer for solving restricted games. To further reduce the complexity, we first incorporate warm-starting by initializing strategy when solving a new restricted game with the average strategy learned in previous restricted game. Cold start of vanilla DO requires solving k restricted game from scratch. Additionally, due to the sample-efficiency of stochastic regret minimization (SRM), we propose to adopt Monte-Carlo Counterfactual Regret Minimization (MCCFR) (Lanctot et al., 2009), for the restricted game solving in DO. In theory, we manage to reduce the power of X in the sample complexity of AdaDO and enhance the scalability of Double Oracle methods. We present a comprehensive summary of theoretical results in Table 1, where Periodic Double Oracle (PDO) is naive improved instance by setting a constant expansion frequency but suffering from tuning this constant hyperparameter in practice. Stochastic PDO (SPDO) and Stochastic Adaptive Double Oracle (SADO) are the natural extension of PDO and AdaDO adopting SRM for restricted game solving[1].

Empirical results in research poker games and board games have demonstrated that AdaDO significantly outperforms XDO and can converge to solutions over $10\times$ less ex-

1. This work is an extension of our previous work (Tang et al., 2023), where RMDO and PDO were presented. In this paper, we first extend RMDO by offering a tighter sample complexity. More importantly, we also propose a novel DO method, namely AdaDO that has a reduced sample complexity by $\mathcal{O}(|\mathcal{S}|^{0.5})$ compared to previous RMDO/PDO. Finally, we introduce warm starting and Stochastic Regret Minimization for RMDO to further improve of two-player zero-sum game-solving.

ploitable than the strong regret minimization baseline, Linear Counterfactual Regret Minimization. Notably, we observe a substantial improvement of AdaDO with the warm starting technique, especially in a variant of Kuhn Poker, which enables DO to converge significantly less exploitable solutions than DO without warm starting. In the setting of reaching NE via SRM, the instances of Stochastic RMDO can also generate a significantly less exploitable strategy compared to MCCFR. These results validate that AdaDO establishes a new state-of-the-art in EFG solving and provides a promising direction for addressing complex multi-agent decision-making tasks using DO methods.

2 Preliminaries and Related Works

In this section, we introduce the background of this paper including the formulation of Two-Player Extensive-Form Games, regret-minimization algorithms for solving EFGs, sample complexity in games, and Double Oracle methods.

2.1 Two-Player Extensive-Form Games

In this paper, our focus centers on Two-Player Zero-Sum Extensive-Form Games (EFGs) with perfect recall (all historical events can be remembered). EFGs are depicted using a game tree, wherein nodes correspond to players $i \in \mathcal{P} = \{1, 2\}$. To model stochastic events, such as card dealing in Poker, Imperfect Information EFGs utilize a **Chance Player** denoted as $\mathbf{c}$. A crucial concept is the notion of a **History** (h), which represents a sequence of actions taken by the players and events uniquely attached to a node on the game tree such as dealt hand cards and public cards in Texas Hold'em Poker. For a given history h, $A(h)$ represents the set of available actions, and $P(h)$ denotes the player required to make a decision at h. Denote $h \cdot a$ as the history right after taking action a at history h. Terminal histories, comprising the set Z, are linked to nodes where the game's payoff can be determined. The payoff at the terminal history $z \in Z$ is denoted as $v_i(z)$, representing the value for player i at z, with the payoff range represented by Δ.

Each player $i \in \mathcal{P}$ possesses an **Information Set** (Infoset/Infostate s_i), which corresponds to the set of indistinguishable histories from player i's perspective. For example in poker, histories where opponent has different hand cards are indistinguishable to us. The set $\mathcal{S}_i$ contains all the infosets where player i must make decisions, and $\mathcal{S}$ represents the set of information sets for all players: $\mathcal{S} = \cup_{i \in \mathcal{P}} \mathcal{S}_i$. The set $A(s_i)$ consists of all available actions for player i at s_i. Player i's **Strategy** is denoted by π_i, where $\pi_i(s_i, a)$ represents the probability of player i taking action $a \in A(s_i)$ at the information set s_i. The joint strategy of both players is represented as $\pi = (\pi_1, \pi_2)$. The **Reaching probability** $x^\pi(h)$ signifies the probability of reaching history h when players use joint strategy π. More specifically, $x^\pi(h) = \prod_{i \in \mathcal{P} \bigcup \mathbf{c}} x_i^\pi(h)$, where $x_i^\pi(h) = \prod_{h' \cdot a \subset h} \pi_{\mathcal{P}(h')}(h', a)$ represents player i's contribution. $x^\pi(s) = \sum_{h \in s} x^\pi(h)$ for any infoset s. Denote iteration as t, throughout the paper, we denote the iteration index as t, employing superscript t and subscript i notation to identify temporal progression and player differentiation, respectively.

Given a joint strategy $\pi = (\pi_1, \pi_2)$, the **expected value** of history h for player i is denoted as $v_i(h)$. If reaching probability $x^\pi(h) = 0$, then $v_i^\pi(h) = 0$; otherwise, we have:

$$v_i^\pi(h) = \sum_{z(h)\in Z} \frac{x^\pi(z(h))}{x^\pi(h)} v_i(z(h)),\ x^\pi(h) > 0, \tag{1}$$

where $z(h)$ denotes the terminal history z that passes through the history h. The value (utility) function of strategy π is defined as:

$$v_i(\pi_i, \pi_{-i}) = \sum_{z\in Z} v_i^\pi(z) x^\pi(z). \tag{2}$$

The **best response (BR)** of player i to the strategy of the other player (π_{-i}) is denoted as $\mathbb{BR}_i(\pi_{-i}) = \arg\max_{\pi_i} v_i(\pi_i, \pi_{-i})$. An ϵ-**Nash Equilibrium (NE)** strategy π^* satisfies the following conditions for every $i \in \mathcal{P}$:

$$v_i(\pi^*) \geq \max_{\pi_i} v_i(\pi_i, \pi_{-i}^*) - \epsilon. \tag{3}$$

Particularly, an exact NE is an ϵ-NE when $\epsilon = 0$. The **Exploitability** $e(\pi)$ is defined as the difference between the sum of values for each player when playing the best response ($\mathbb{BR}(\pi_{-i})$) and the value achieved when players playing π. Additionally, the **Support Size** of NE (π^*) refers to the number of actions that have positive probabilities at the infoset s in NE π^*, denoted by $\text{supp}^{\pi^*}(s)$.

2.2 Regret Minimization and Stochastic Regret Minimization

Regret minimization methods approximate NE in EFGs if the methods have a sublinear regret upper bound or an average regret converging to zero. We first define the core theoretical metric of online learning in games:

Definition 1 (Regret)**.** *Given $\{\pi^t |\ t = 1, \cdots, T\}$ is a sequence of strategies produced by an algorithm, the cumulative regret of this algorithm is defined as*

$$R_i^T = \max_\pi \sum_{t=1}^{T} v_i(\pi, \pi_{-i}^t) - v_i(\pi_i^t, \pi_{-i}^t), \tag{4}$$

Average regret is then defined as $\bar{R}_i^T = R_i^T/T$. Counterfactual Regret Minimization (CFR) (Zinkevich et al., 2007) is a regret minimization algorithm that minimizes cumulative regret by minimizing local counterfactual regret at each infoset via traversing the full game tree depth-firstly. CFR has been widely studied in Poker AI (Burch et al., 2012; Moravčík et al., 2017; Brown and Sandholm, 2019a, 2017, 2019b; Brown et al., 2020a).

To obtain the counterfactual regret, we compute player i's expected values $v_i(\cdot)$ based on Equation (1) and the instantaneous regrets at iteration $t \leq T$ of taking action a in infoset s

$$r_i^t(s_i, a) = \sum_{h\in s_i} x_{-i}^{\pi^t}(h)[v_i^{\pi^t}(h \cdot a) - v_i^{\pi^t}(h)], \tag{5}$$

where $x_{-i}^{\pi^t}(h)$ is the product of all players' contribution (including chance player) to the probability $x^{\pi^t}(h)$ except player i, which is also called counterfactual reaching probability. The counterfactual regrets of CFR can be computed by uniform average of r_i over all iterations:

$$R_i^T(s,a) = \sum_{t=1}^{T} r_i^t(s,a)/T. \tag{6}$$

Denote $R_i^{t,+}(s,a) = \max\{0, R_i^t(s,a)\}$. In two-player zero-sum games, if both players apply **regret matching** (Zinkevich et al., 2007) to strategy updates:

$$\pi_i^{t+1}(s,a) = \begin{cases} R_i^{t,+}(s,a)/\sum_a R_i^{t,+}(s,a), & \text{if} \sum_a R_i^{t,+}(s,a) > 0; \\ 1/|A(s)|, & \text{else} \end{cases}, \tag{7}$$

the (cumulative) regret bound of CFR is as following:

$$R_i^T \leq O(\Delta|\mathcal{S}_i|\sqrt{|\mathcal{A}_i|T}). \tag{8}$$

CFR needs to traverse the entire game tree to estimate $v_i^{\pi^t}(h)$ in Equation (5) and develop strategy for all states according to regret matching, which can be intractable in large games. Hence rather than computing exact regret, Stochastic Regret Minimization (SRM) (Farina et al., 2020) samples nodes for traversal to estimate regret and then minimize regret. The family of SRM, including outcome-sampling Monte-Carlo CFR (MCCFR) and external-sampling MCCFR, have the following regret bound with at least probability $1-p$:

$$R_i^T \leq O\big((1+\sqrt{\frac{2}{p}})\frac{1}{\delta}\Delta|\mathcal{S}_i|\sqrt{|\mathcal{A}|T}\big), \tag{9}$$

for any $p \in (0,1]$ and exploration parameter $\delta > 0$. Specifically, $\delta = 1$ in external-sampling MCCFR (Lanctot et al., 2009). Based on stochastic regret minimiziation, some works extend tabular CFR method to using neural networks as function approximators (Brown et al., 2019a; Steinberger et al., 2020; Brown et al., 2020b; McAleer et al., 2021). The coefficient $1+\sqrt{\frac{2}{p}}$ was later improved to $\sqrt{\log(1/p)}$ by Farina et al. (2020). For simplicity, we retain the original coefficient in our analysis, but it can be directly replaced with this tighter bound.

2.3 Sample Complexity in Games

Previous regret minimization works report regret bound as the theoretical metric for evaluating the efficiency (Zinkevich et al., 2007; Lanctot et al., 2009; Brown and Sandholm, 2019a; Brown et al., 2019b), as in Equation (8) and (9). The derivation from regret upper bound to the *Iteration Complexity* is straightforward, i.e. the required iterations to reach a specific ϵ-Nash Equilibrium (ϵ-NE) in the worst case. Cesa-Bianchi and Lugosi (2006); Blum and Monsour (2007); Waugh (2009) state that if both players in a two-player zero-sum game adopt an algorithm with a sublinear regret bound, then the average strategies of both players will converge to a Nash Equilibrium:

Lemma 2. *In CFR (Zinkevich et al., 2007), since the average regret is $\mathcal{O}(|\mathcal{S}_i|\sqrt{|\mathcal{A}_i|}/\sqrt{T})$, the average strategy is a $\mathcal{O}(|\mathcal{S}|\sqrt{|\mathcal{A}|})/\sqrt{T})$-NE.*

The proof is in Appendix 2. According to Lemma 2, if the average regret can converge to zero with $T \to \infty$, the resulting average strategy is an asymptotic approximation that converges to NE. Specifically, to ensure the average strategy reaches ϵ-NE, it is required that $\mathcal{O}(|\mathcal{S}|\sqrt{|\mathcal{A}|}/\sqrt{T}) \leq \epsilon$. Thus, the required number of iterations T for CFR to reach ϵ-Nash Equilibrium satisfies

$$T \geq \mathcal{O}(|\mathcal{S}|^2|\mathcal{A}|/\epsilon^2). \tag{10}$$

The RHS of the Inequality (10) is the *Iteration Complexity* of CFR.

However, merely studying iteration complexity is insufficient. Since the time complexity of conducting single iteration varies in different algorithms, directly comparing iteration complexity is inaccurate and unfair. We offer three examples to elaborate this problem.

- **CFR vs. MCCFR.** Although CFR and MCCFR have the same magnitude of iteration complexity Zinkevich et al. (2007); Lanctot et al. (2009), CFR needs to visit all information sets in each iteration, while MCCFR only needs to visit a subset of the information sets. Comparing them with merely iteration complexity is unfair. In facts, it has been empirically demonstrated that MCCFR has superior efficiency in solving large-scale EFGs Brown and Sandholm (2019b).

- **Different MCCFR variants like outcome-sampling and external-sampling MCCFR.** Both methods have the same iteration complexity. However, their time complexities of single iteration are also different, which makes the comparison between them hard by only comparing iteration complexity.

- **Double Oracle methods.** Time complexities are not the same in all iterations in Double Oracle (DO), due to the expanding restricted games. The restricted game of Double Oracle in different iteration could have game tree with significantly different sizes. Thus, merely deriving the required number of iterations is insufficient to accurately evaluating complexity of DO methods.

Therefore, we propose to leverage a more accurate theoretical metric *Sample Complexity* for algorithms comparisons, defined as following:

Definition 3 (Sample Complexity)**.** *Sample Complexity of an algorithm to reach ϵ-Nash Equilibrium is the multiplication of Iteration Complexity and the time complexity of conducting single iteration.*

Sample complexity measures the runtime complexity accurately, since it can be treated as the required number of visited nodes on the game tree to reach approximate NE. For instance, outcome-sampling MCCFR has the same magnitude of regret as CFR, but its sample complexity is much lower since the runtime complexity of each iteration in CFR is $\mathcal{O}(|\mathcal{S}|)$ while that of MCCFR is only $\mathcal{O}(H)$, where H is the depth of the tree satisfying $H \ll |\mathcal{S}|$. In this way, CFR has the sample complexity

$$\mathcal{O}(|\mathcal{A}||\mathcal{S}|^2/\epsilon^2) \cdot |\mathcal{S}| = \mathcal{O}(|\mathcal{A}||\mathcal{S}|^3/\epsilon^2). \tag{11}$$

Thus, CFR's sample complexity is cubic in $|\mathcal{S}|$ for reaching ϵ-NE. The sample complexity of outcome-sampling MCCFR is reduced by up to $\tilde{\mathcal{O}}(|\mathcal{S}|)$, being $\mathcal{O}(|\mathcal{A}||\mathcal{S}|^2 H/\epsilon^2)$.

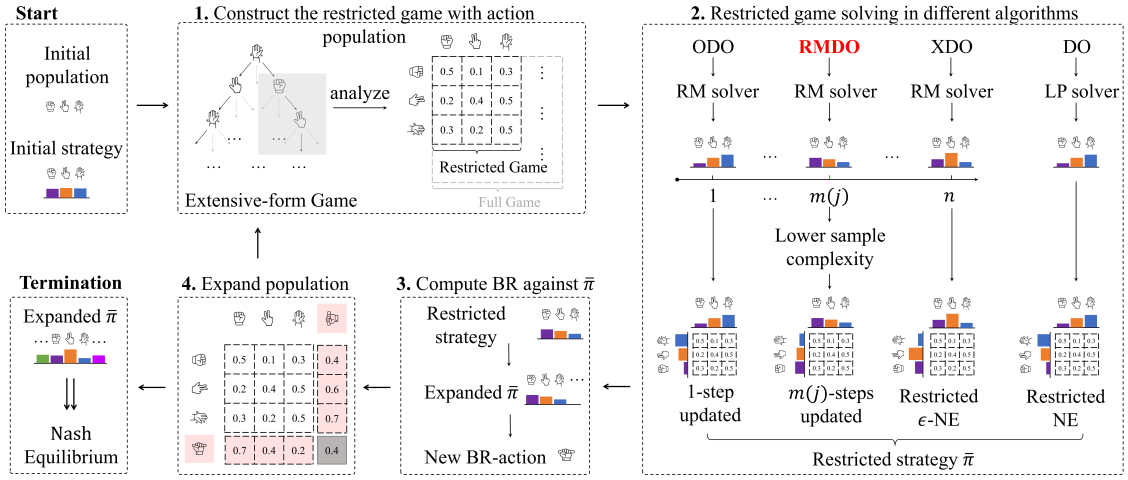

Figure 1: Flow chart of existing Double Oracle algorithms: Double Oracle (DO), Extensive-form Double Oracle (XDO), Online Double Oracle (ODO), and the method we proposed, namely Regret-Minimizing Double Oracle (RMDO).

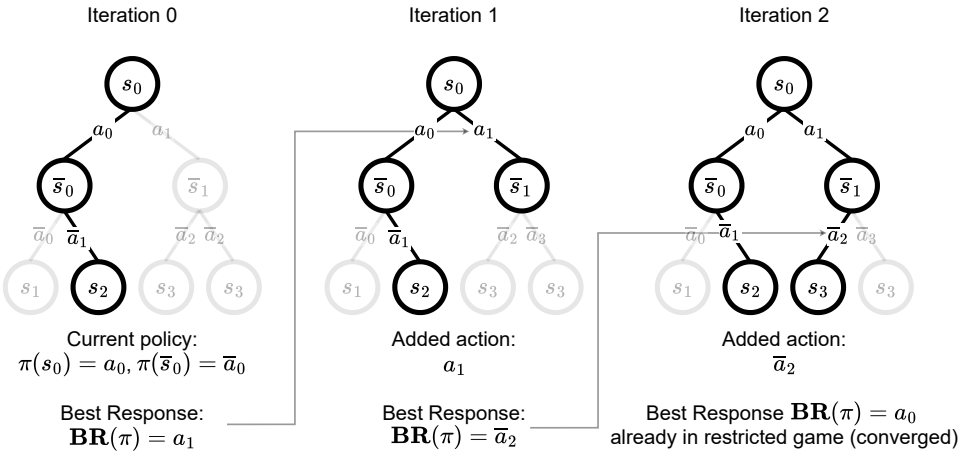

Figure 2: DO process in Extensive-form Games, where nodes are infosets and edges are actions. The transparent nodes and edges are not included in the restricted game in that iteration. Restricted game solving (Step 2. in Figure 1) is to apply Extensive-form RM solver like CFR. BR computing (Step 3. in Figure 1) is via dynamic programming in EFG. The population expansion (Step 4. in Figure 1) is equivalent to add edge and the connected node to the restricted game tree.

2.4 Double Oracle Methods

The Double Oracle (DO) technique (McMahan et al., 2003) originated as a method for addressing Normal-form Games (NFGs). At each time step t, DO maintains a population

of actions denoted as Π_t for both players and proceeds to construct a restricted game by considering only the actions in Π_t. The Nash Equilibrium (NE) of this restricted game is then obtained through linear programming, and subsequently, the best response to this NE is incorporated into the population. This iterative process continues until the population reaches a stable state without further changes.

Notably, DO offers a key advantage in solving large games by restricting attention to much smaller subgames (i.e. restricted games in this paper) (Wilson, 1972; Koller and Megiddo, 1996). This idea was later generalized in Policy Space Response Oracles (PSRO) (Lanctot et al., 2017) by applying approximate best responses, a multi-agent reinforcement learning framework closely related to empirical game-theoretic analysis (EGTA) (Wright, 2016; Schvartzman and Wellman, 2009). In both settings, small empirical games are constructed over discovered strategies in order to reason at the meta level and ultimately solve the original large game. PSRO can be applied to a wide range of settings, including extensive-form zero-sum games (McAleer et al., 2020, 2022) and general-sum games (Marris et al., 2021; Liu et al., 2024).

PSRO leverages regret minimization as an approximate solver for restricted games, and reinforcement learning as approximate best responder, rather than computing an exact Nash Equilibrium (NE) and best response (BR). However, PSRO suffers from high iteration complexity in extensive-form games because it operates on the normal-form representation of extensive-form strategies (McAleer et al., 2021). More recent methods instead adopt strategy representations that are better aligned with the structure of extensive-form games (Bosansky et al., 2014; McAleer et al., 2021). One particularly efficient method for extensive-form games is Extensive-Form Double Oracle (XDO), which directly expands restricted games with extensive-form BR at each information set. XDO introduces an exploitability threshold ϵ to determine when strategy updates within a restricted game should terminate. Specifically, XDO repeatedly applies regret minimization in the restricted game until an ϵ-NE is reached. It then halves ϵ and constructs a new restricted game with the best-response actions to the current restricted-game equilibrium. The strategy is then reset, and the procedure repeats. XDO guarantees that the average strategy in the final restricted game converges to an NE in two-player extensive-form games.

In contrast, Online Double Oracle (ODO) (Dinh et al., 2022) performs only a single iteration of regret minimization before computing a BR to the current restricted-game equilibrium. If the resulting best-response actions are not already contained in the restricted game, ODO expands the restricted game accordingly and resets the strategy; otherwise, it continues with the same procedure. The average strategy generated by ODO is likewise guaranteed to converge to an NE. Empirical results from prior work (McAleer et al., 2021; Qin et al., 2022; Tang et al., 2023) suggest that, in poker and board-game settings, DO-based methods reduce exploitability faster than standard regret minimization. Despite the empirical success of DO, its theoretical properties in extensive-form games, particularly regarding convergence rates and sample complexity, remain largely unexplored, which makes it difficult to further improve these algorithms (Bosansky et al., 2014). In this work, we develop a theoretical framework for Double Oracle methods to study the complexity of approximating Nash equilibria. Based on this framework, we identify the main sources of complexity and propose corresponding algorithmic designs to reduce them and thereby accelerate convergence. Although prior work has explored techniques such as warm-starting

to speed up convergence in PSRO (Zhou et al., 2022; Wang et al., 2019), these methods inherit the limitations of PSRO and are therefore restricted to normal-form representations of extensive-form strategies. In this work, we investigate warm-starting methods that are directly adapted to extensive-form strategies for complexity reduction.

3 Regret-Minimizing Double Oracle

In this section, we propose Regret-Minimizing Double Oracle (RMDO), a novel and versatile Double Oracle framework that integrates regret minimization to approximate the Nash Equilibrium (NE) of Extensive-Form Games (EFGs). To the best of our knowledge, this study represents the first comprehensive analysis of convergence rate and sample complexity of regret-minimization-based Double Oracle for EFGs.

RMDO is an iterative algorithm, which consists of the same elements as the previous DO methods but extended to extensive form. Denote current iteration by t.

Definition 4 (**Population**). *Population* $\Pi_t \subseteq \mathcal{S} \times \mathcal{A}$ *is a set of infoset-action pairs.*

For each element $(s, a) \in \Pi_t$, by treating infoset s as a set of nodes in the EFG tree, the action a represents edges connecting s and the outcome infosets of taking a, denoted by $Succ(s, a)$. Therefore, the population Π_t consists of the edges (actions) and nodes (infosets) that can be used to construct a restricted game.

Definition 5 (**Restricted Game** (McAleer et al., 2021)). *A restricted extensive-form game* $\mathbf{G}_t$ *is constructed by transforming the original base game to limit the set of allowable actions at each information state* s *to the actions in the population* Π_t.

DO methods aim to iteratively solve an expanding restricted game until the approximate Nash Equilibrium (NE) strategy of the restricted game also serves as an approximate NE strategy of the original game. Empirically, achieving a sufficiently accurate approximate NE of the original game does not necessarily require the restricted game to expand to the full size of the original game.

Definition 6 (**Time Windows**). *Time windows* $\{T_j | j \in \mathbb{N} \cap [1, k],\ k \in \mathbb{N}^+\}$ *is the partition of the set of all iterations* $\mathbb{N} \cap [1, T]$*, where the two iterations are in the same time window if the populations in those two iterations are the same. In other words,* $\forall j \in \mathbb{N} \cap [1, k],\ \forall t_0, t_1 \in \mathbb{N} \cap [1, k],\ t_0, t_1 \in T_j$ *if and only if* $\Pi_{t_0} = \Pi_{t_1}$.

Here, the total number of time windows and the number of distinct restricted games are equal, denoted by k. Time windows partition the iterations into several groups, where the restricted game remains unchanged within each group. Previous DO methods apply a game solver during each time window, and computes best-response actions to expand the restricted game, thereby transitioning into a new time window.

Herein, we present the distinguishing innovation of RMDO that fundamentally differentiates it from previous DO methods.

Definition 7 (**Frequency Function**). *Frequency function* $m(\cdot) : \mathbb{N} \cap [1, k] \to \mathbb{N}^+$ *is a mapping from time window index to the frequency of computing Best Response in* T_j.

Algorithm 1 Regret-Minimizing Double Oracle

Input: Frequency function $m(\cdot)$, window index $j = 0$, uniform random strategy π^0.
Set population $\Pi_1 = \mathbb{BR}_i(\pi^0)$ for $i \in \{1, 2\}$
Construct restricted game $\mathbf{G}_1$ with Π_1
for $t = 1, \cdots, \infty$ **do**
  Run one iteration of CFR in $\mathbf{G}_t$
  if $t \mod m(j) = 0$ **then**
    Compute last-window average strategy following Eq. 14
    $\Pi_{t+1} = \Pi_t \cup \mathbf{BR}_i(\tilde{\pi}^t_{-i})$ for $i \in \{1, 2\}$
    if $\Pi_{t+1} \neq \Pi_t$ **then**
      Start new window: $j = j + 1$
      Reset strategy π^{t+1}
      Construct restricted game $\mathbf{G}_{t+1}$ with Π_{t+1}
    end if
  end if
end for

In essence, regret-minimization-based DO alternates between executing regret minimization and computing the best response. We alter the protocol of DO to that best response computing can be executed before the restricted game is solved. We use the frequency function to determine how often the best response is computed during regret minimization. Therefore, frequency function plays a crucial role in RMDO and establishing it as a general framework. Unlike existing DO methods, RMDO offers the flexibility to control the expansion rate of the restricted game. By designing an efficient expansion scheme via $m(\cdot)$, RMDO can effectively balance regret minimization and best-response computation, leading to a more rapid reduction in exploitability.

Presented in Algorithm 1, the formal RMDO procedure is as follows. At each iteration t, assuming the current time window is j, the restricted game $\mathbf{G}_t$ is constructed by restricting the pure strategies in the population Π_t for players. Within $\mathbf{G}_t$, regret minimization is conducted by traversing the game tree, computing the regret of each infoset (node), and updating the strategy using any regret-minimization algorithms. At the outset of the procedure, when $t = 0$, the construction of the restricted game and the strategy update are bypassed since Π_0 is empty. The expected value at $t = 0$ is computed based on the joint strategy π following a uniformly random policy. As the procedure progresses, when $t > 0$ and the current time window is T_j, the joint average strategy of current window $\bar{\pi} = (\bar{\pi}_1, \bar{\pi}_2)$ is expanded to the original game every $m(j)$ iteration by setting the probabilities of actions not in the population to zero. Then the original game best response (BR) to the expanded current-window average strategy [2], is computed by considering all actions in the original game: $\mathbf{a}_i^t = \arg\max_{\pi_i \in \Pi} v_i(\pi_i, \pi_{-i})$, for both players. $\mathbf{a}_i^t$ for $i = 1, 2$ are both merged to the population Π_{t+1}. Finally, if the population changes ($\Pi_{t+1} \neq \Pi_t$), a new time window is initiated, and π_i^{t+1} is reset to a uniform random strategy.

2. In this paper, we assume all the BR strategies are pure strategies.

3.1 Convergence Guarantee and Sample Complexity

Then we investigate the convergence guarantee and sample complexity of RMDO. We first define an important statistic related to the support size of Nash Equilibrium and prove that k is bounded by this statistic. This is the key to the convergence guarantee. The reason is as following. Regret minimization algorithms can converge to ϵ-NE by iteratively updating strategy in a static game. But in RMDO, a regret minimizer is employed in the restricted game expanding over time. Thus if the restricted game stops expanding at some finite iteration, the convergence of RMDO is guaranteed. The following lemma proves that the number of different restricted games is finite regardless the final iteration T.

Definition 8. *Denote the number of time windows from iteration $t = 0$ to T as k. Define*

$$X = \sum_i |\mathcal{S}_{i,k}|, \tag{12}$$

as the size of the final restricted game.

We then show that the statistics above are bounded:

Lemma 9. $\min_{\pi \in \Pi^*} \max_{s \in S} supp^{\pi}(s) < k \leq X \leq |\mathcal{S}| = \sum_i |\mathcal{S}_i|$, *where k is the number of restricted game during the whole process of Double Oracle.*

RMDO will converge by doing regret minimization in the final restricted game, after $k-1$ times of restricted game expanding. Then X, standing for the number of infosets in the largest one of these games, can be small in small support games (Bošanskỳ et al., 2016). In practice, X and k are usually strictly less than $|\mathcal{S}|$, but X is usually larger than k. For example, in Hold'em poker games, at the infoset where we have the weakest hand cards but are facing with a large value bet by the opponents, the equilibrium strategy will always take the action of fold regardless of any other actions. Such appearance of deterministic action choice in specific states can significantly reduce X by the number of nodes exponential in the tree depth H, and thus lead to $X < |\mathcal{S}|$.

Now we investigate the sample complexity of RMDO. There are two kinds of average strategies produced by RMDO. **Overall average strategy (OAS)** is the reach-weighted (sequence-form) average strategy following definition in (Zinkevich et al., 2007) over all time windows

$$\bar{\pi}_i^t(s_i, a) = \sum_{t=0}^{T} x^{\pi^t}(s_i)\pi_i^t(s_i, a) / \sum_{t=0}^{T} x^{\pi^t}(s_i). \tag{13}$$

Last-window average strategy (LAS) is the average strategy in the final time window

$$\tilde{\pi}_i^t(s_i, a) = \sum_{t \in T_k} x^{\pi^t}(s_i)\pi_i^t(s_i, a) / \sum_{t \in T_k} x^{\pi^t}(s_i), \tag{14}$$

where k is in hindsight the number of time windows (i.e. the number of different restricted games) of training till reaching the ϵ-NE. In the empirical study, OAS is much worse than LAS in terms of the decreasing speed of exploitability during training (Tang et al., 2023). Besides, OAS is only used in Online Double Oracle, and not used for developing new methods in this paper, we put the regret bound analysis of OAS in Appendix (Theorem 19). Here we only introduce the sample complexity of LAS:

Theorem 10. *Denote A_j as the action space in the j-th restricted game, $S_{i,j}$ as the set of all infosets of player i in the j-th restricted game. The sample complexity of LAS of RMDO to reach ϵ-NE is:*

$$\sum_{j=1}^{k} \tilde{\mathcal{O}}\Big(A_j|\mathcal{S}|(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 m(j) + A_j(\sum_i |\mathcal{S}_{i,j}|)^3/\epsilon^2 + \sum_i |\mathcal{S}_{i,j}|m(j)\Big), \tag{15}$$

Compared to the result derived in our previous work (Tang et al., 2023), we reduce the dependence on the infoset of the original game, denoted by $|\mathcal{S}|$. In Theorem 10, Equation (15) is in the form of $a/m(j) + bm(j)$, for some $a > 0, b > 0$, thus it has a lower bound independent of frequency function $a/m(j) + bm(j) \geq 2\sqrt{ab}$. Additionally, we can derive the sample complexity of existing DO methods, and accordingly propose more sample-efficient RMDO instances.

3.2 Existing Frequency Schemes

The complexity of approximating NE using RMDO is influenced by the choice of frequency function $m(j)$ for best response computation, as demonstrated in the previous section through theoretical analysis. For analysis on existing DO methods, we present various existing RMDO instantiations in this section.

3.2.1 Online Double Oracle for Extensive-Form Games

We introduce an extension of the Online Double Oracle (ODO) algorithm (Dinh et al., 2022) called Extensive-Form Online Double Oracle (XODO). This algorithm integrates the Sequence-form Double Oracle (DO) framework with Counterfactual Regret Minimization (CFR) to effectively address extensive-form games with overall average strategy (OAS). XODO computes the best response actions to the average strategy in each iteration. Therefore, XODO is equivalent to RMDO algorithm with $m(j) = 1$, using OAS to approximate the NE of the original game. We then provide the sample complexity of XODO (proof in Appendix B).

Proposition 11. *XODO is an instance of RMDO when $m(\cdot) \equiv 1$, thus given the regret minimizer with $\tilde{\mathcal{O}}(|\mathcal{S}_i|\sqrt{|\mathcal{A}|T})$ regret upper bound, the sample complexity for the OAS of XODO to reach ϵ-NE is*

$$\tilde{\mathcal{O}}\Big(k^2X^3/\epsilon^2 + |\mathcal{S}|k^2X^2/\epsilon^2\Big). \tag{16}$$

3.2.2 Extensive-Form Double Oracle

The Extensive-form Double Oracle (XDO) algorithm is initialized with a given threshold ϵ_0, which is divided by two each time the local exploitability of the regret minimizer meets the threshold. The local exploitability is the exploitability in the restricted game. In time window T_j, the algorithm performs regret minimization for more than $4^j|\mathcal{S}_{i,j}|^2A_{i,j}/\epsilon_0^2$ iterations before computing the best response. Here $A_{i,j}$ and $S_{i,j}$ denote the action space and infoset space in the j-th time window of player i. Finally the average strategy in the last window is outputted when the convergence condition is met. If XDO converges, the

last-window average strategy is $\epsilon_0/2^k$-NE. To investigate the complexity of reaching ϵ-NE, it is assumed without loss of generality that $\epsilon_0/2^k \leq \epsilon$.

Thus, RMDO can generalize to XDO with $m(j) = 4^j|\mathcal{S}_{i,j}|^2 A_{i,j}/\epsilon_0^2$ and the last-window average strategy. Based on Theorem 10, we can determine the expected iterations and sample complexity for XDO to reach ϵ-NE.

Proposition 12. *XDO is an instance of RMDO. In the worst case, its frequency function* $m(j) = 4^j|\mathcal{S}_{i,j}|^2 A_{i,j}/\epsilon_0^2$, *where* $j = 0, 1, \cdots, k-1$. *Thus the sample complexity bound of XDO to reach* ϵ*-NE is*

$$\tilde{\mathcal{O}}(k|\mathcal{A}|X^3/\epsilon^2 + |\mathcal{A}|X^3 4^k/\epsilon_0^2). \tag{17}$$

Proposition 12 is a specific instance of Theorem 10 with an appropriate choice of $m(j)$ (see proof in Appendix B). Lemma 9 states that $k \leq |\mathcal{S}|$; thus, in the worst-case scenario when $k = |\mathcal{S}|$, XDO has an exponential sample complexity in the number of infosests, because the restricted game stopping condition of XDO decays exponentially. Therefore, XDO suffers from a large theoretical sample complexity. We demonstrate the algorithmic distinctions between existing DO and RMDO via a flowchart presented in Figure 1.

3.3 Causes of High Sample Complexity

We delve into an in-depth examination of the sources contributing to the sample complexity of existing RMDO instances, and explore potential approaches to mitigate and minimize these complexities in the following sections. We investigate from the perspectives of three elements in the sample complexity (Equation (15)):

- The selection of the frequency function $m(j)$ significantly impacts the performance of RMDO. In XDO, the exponential frequency function, caused by exponentially decaying stopping threshold for restricted games, contributes to an exponential sample complexity in terms of $|\mathcal{S}|$ in the worst case. In Section 3.4 and Section 4.1, we will propose two instances of RMDO that only have polynomial sample complexity.
- The multiplier k appearing in sample complexity is due to the cold starting of solving restricted games. When constructing a new restricted game, RMDO reset strategy and cumulative regret, resulting in k independent procedures of game solving. Consequently, the appearance of k in the sample complexity arises. In section 4.2, we present an algorithmic design called warm starting to potentially address this.
- The power of the dominating term X (3 in XDO and ODO) is still high. Inspired by the sample efficient method, stochastic regret minimization, in Section 4.3, we propose Stochastic Regret-Minimizing Double Oracle as a solution to enhance scalability and reduce the the high power of X in the complexity analysis.

3.4 Periodic Double Oracle: The First Attempt to Reduce Complexity

The exponentially growing frequency function $m(j)$ of XDO leads to an exponential increase in sample complexity with respect to k. On the other hand, XODO's inflexibility arises from the fact that it performs best response computation in each iteration, neglecting the balance

between regret minimization and best response computation. In this section, to mitigate the large increase in sample complexity caused by a large value of k, and to balance the two computations, we introduce Periodic Double Oracle (PDO). PDO is to simply set a constant frequency of restricted game expanding of RMDO.

Periodic Double Oracle (PDO) is derived from Regret-Minimizing Double Oracle (RMDO) by introducing a constant frequency value, denoted as $m(j) = c > 1$. In practice, we treat c as a hyperparameter that can be tuned. PDO reaches NE with its last-window average strategy, so we can derive the sample complexity using Theorem 10.

Proposition 13. *Given constant c, since PDO computes BR every c iterations, it is an instance of RMDO when $m(\cdot) \equiv c$ and its sample complexity to reach ϵ-NE:*

$$\tilde{\mathcal{O}}(k|\mathcal{A}||\mathcal{S}|X^2/c\epsilon^2 + k|\mathcal{A}|X^3/\epsilon^2 + ckX). \tag{18}$$

Proof By applying $m(\cdot) = c$ in Theorem 10, the sample complexity of PDO is

$$\sum_{j=1}^{k} \tilde{\mathcal{O}}\Big(A_j|\mathcal{S}|(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 c + A_j(\sum_i |\mathcal{S}_{i,j}|)^3/\epsilon^2 + \sum_i |\mathcal{S}_{i,j}|c\Big) \tag{19}$$

$$\leq \tilde{\mathcal{O}}(k|\mathcal{A}||\mathcal{S}|X^2/c\epsilon^2 + k|\mathcal{A}|X^3/\epsilon^2 + ckX). \tag{20}$$

■

Periodic Double Oracle (PDO) is more sample efficient than previous the Double Oracle variants by simply adopting a constant frequency $m(j) = c > 1$, and tuning this hyperparameter c during execution to solve a game. In comparison to XODO, since $c > 1$, PDO's upper bound for the dominant term in sample complexity is strictly less than that of XODO. Additionally, PDO exhibits a sample complexity only linear in k, making it significantly more sample-efficient than XDO, as it eliminates the exponential term in k, and slightly better than ODO, whose sample complexity has k^2 term.

However, tuning the hyperparameter c in PDO can be challenging in practice. The optimal periodicity c varies depending on the game. More importantly, using a fixed frequency for all restricted games of different sizes is overly simplistic. The best response computation scheme for expanding restricted games, which may differ significantly in size, should be adaptive rather than fixed.

In the next section, we introduce a new instantiation of RMDO that adaptively adjust the frequency according to the restricted game, achieving the sample complexity lower bound of RMDO through an adaptive frequency function.

4 Further Improvement of RMDO

In this section, we introduce further improvements of RMDO instances from three perspectives discussed in Section 3.3, aiming at reducing sample complexity caused by frequency function $m(\cdot)$, number of restricted games k, and the last restricted game size X.

4.1 Adaptive Double Oracle

Although PDO has reduced the sample complexity to polynomial, tuning the frequency constant c is hard since in different restricted game with different size, we need to retune it, making it hard to generalize. Thus, we finally propose Adaptive Double Oracle (AdaDO), featuring an *optimal* expansion frequency function adaptively for restricted games of different sizes. Therefore, the tuning is less demanding. Additionally, the frequency function of AdaDO is provably the optimal choice in terms of theoretical sample complexity.

In this section, we propose a new instantiation of RMDO that has the lowest sample complexity among all RMDO applying different expansion frequencies, called **Adaptive Double Oracle (AdaDO)**. AdaDO has dynamic periodicity and reach NE with last-window average strategy. We first introduce the adaptive frequency function for AdaDO:

Definition 14 (Frequency Function of AdaDO)**.** *Denote A_j as $\max_{i\in\mathcal{P},s\in S_i}|\mathcal{A}_{i,j}(s)|$, where $\mathcal{A}_{i,j}(s)$, $S_{i,j}$ are defined as the set of actions at infoset s and the set of infosets in time window T_j, respectively. The adaptive frequency function of AdaDO is*

$$m(j) = \sqrt{A_j|\mathcal{S}|\sum_i |\mathcal{S}_{i,j}|}/\epsilon. \tag{21}$$

Here, the choice of $m(j)$ in this context is dependent on ϵ, allowing us to set it according to the desired precision of the result. Furthermore, the frequency function $m(j)$ becomes window-dependent. In contrast to PDO, which selects a frequency function and maintains a fixed periodicity value across all windows, our approach involves computing the statistics of the restricted game in the current window and adjusting the frequency accordingly.

Now we investigate AdaDO's sample complexity. We prove that such adpative frequency function is the optimal choice in terms of the sample complexity.

Proposition 15 (AdaDO)**.** *AdaDO is an instance of RMDO when $m(j)$ satisfies Equation (21). Thus the sample complexity of AdaDO matches the sample complexity **lower bound** of RMDO sample complexity for all frequency function, which is:*

$$\tilde{\mathcal{O}}\Big(2k\sqrt{|\mathcal{A}||\mathcal{S}|}X^{\frac{3}{2}}/\epsilon + k|\mathcal{A}|X^3/\epsilon^2\Big). \tag{22}$$

In theory, among all Double Oracle methods that instantiated from RMDO with different frequency function, AdaDO is provably the most sample efficient method. Compared to PDO, AdaDO has lower order of $|\mathcal{S}|$ from 1 to 0.5, which is the dominating term in the sample complexities of all regret minimization algorithms. Although it has not eliminated all the cubic terms of X, the complexity has been significantly reduced by merely picking a frequency function without changing regret minimizers or best responders.

The formal algorithm is outlined as follows. Initialize the strategy and restricted game using the standard Double Oracle method. Upon entering a new restricted game, perform one iteration of Counterfactual Regret Minimization (CFR). This step involves traversing the entire game tree of the current restricted game, allowing for the computation of A_j (the action space size) and $\sum_i |\mathcal{S}_{i,j}|$ (the sum of information set sizes). Compute the frequency of Best Response computation in the current window, denoted as $m(j)-1$, where subtracting

Algorithm 2 Adaptive Double Oracle (Practical version)

Input: Empirical frequency function of AdaDO $\hat{m}(\cdot)$ in Equation (23), early stop tolerance δ, exploitability function $e(\cdot)$, exploitability checking frequency c for early stop.

Set population $\Pi_1 = \mathbb{BR}_i(\pi^0)$ for $i \in \{1, 2\}$
Construct restricted game $\mathbf{G}_1$ with Π_1
for $t = 1, \cdots, \infty$ **do**
  Run one iteration of CFR in $\mathbf{G}_t$
  if $t \mod \hat{m}(j) = 0$ or ($t \mod c = 0$ and $|e(\tilde{\pi}^{t-c}) - e(\tilde{\pi}^t)|/e(\tilde{\pi}^{t-c}) < \delta$) **then**
    Compute average strategy following Eq. 14
    $\Pi_{t+1} = \Pi_t \cup \mathbf{BR}_i(\tilde{\pi}^t_{-i})$ for $i \in \{1, 2\}$
    if $\Pi_{t+1} \neq \Pi_t$ **then**
      Start new window: $j = j + 1$
      Reset strategy π^{t+1}
      Construct restricted game $\mathbf{G}_{t+1}$ with Π_{t+1}
    end if
  end if
end for

one accounts for the fact that the first iteration of regret minimization is dedicated to obtaining game-related statistics A_j and $\sum_i |\mathcal{S}_{i,j}|$. Subsequently, we will introduce how we empirically deal with $|\mathcal{S}|$ in the frequency function in Equation (21).

A potential empirical problem when applying AdaDO is that the choice of adaptive frequency function in AdaDO is derived by reducing a theoretical (worst-case) complexity. But the empirical complexity can be much smaller, making AdaDO which has the theoretically optimal frequency scheme perform suboptimal empirically. This phenomenon aligns with previous discussions on the impact of this gap on algorithmic design for extensive-form games Brown and Sandholm (2016). To address this, in the practical version shown in Algorithm 2, we adopt a *discounted empirical frequency function*

$$\hat{m}(\cdot) = \alpha \cdot \epsilon m(\cdot)/\sqrt{|\mathcal{S}|}, \text{ where } \alpha > 0. \tag{23}$$

We reduce the absolute value of frequency while preserving the increasing rate of frequency along with the expansion of restricted game. The critical design of frequency function to maintain a low sample complexity is to control the increasing rate of BR computation frequency. XDO has an exponential increasing rate, leading to an exponential sample complexity. Given that $|\mathcal{S}|/\epsilon$ can be large in practice, it is removed. And according to our experiments in Section 4.1, AdaDO with $\alpha \in (0.1, 1]$ already has strong performance. Besides, we execute early stop of restricted game regret minimization when the exploitability is decreasing slowly.

AdaDO is not the first RMDO method employing dynamic frequency function. XDO, which largely follows the original DO process, passively pick a dynamic frequency function. Specifically, in each restricted game, XDO refrains from computing the Best Response and continues regret minimization until its average strategy reaches local ϵ-NE of that restricted game. As analyzed in Section 3, the threshold ϵ in XDO decreases exponentially with the

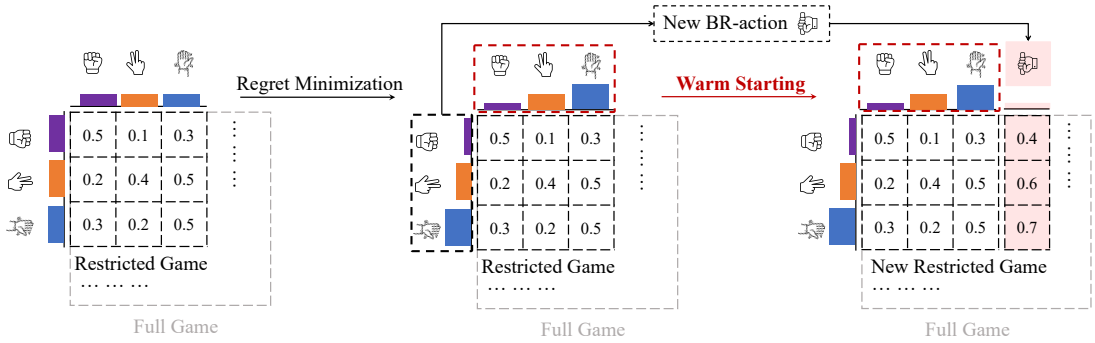

Figure 3: Restricted game expanding and warm starting of Regret Minimizing Double Oracle in EFGs. In restricted game, the regret minimizer will keep updating the regret and average strategy. After $m(\cdot)$ iterations of regret minimization, we compute best response actions (BR) to the restricted strategy (violet, orange and blue bars). If there are new BR actions, we expand the restricted game with them.

time window, leading to its exponential sample complexity. In contrast, AdaDO actively select the frequency function to reach the lower bound of the sample complexity of RMDO, which is only linear in k and polynomial in $|\mathcal{S}|$ in the worst case.

4.1.1 Comparison to Regret Minimization

We next compare Counterfactual Regret Minimization (CFR) with the newly proposed instances of Regret-Minimizing Double Oracle (RMDO). While theoretical complexities are not necessarily indicative of reaching approximate Nash Equilibrium (NE), we still consider sample complexity as a comparable metric for CFR and existing DO methods, given that we compute complexities in a similar manner.

Double Oracle methods are known to be efficient in games with NE characterized by small support. To illustrate this, we compare the dominant term of CFR's complexity $\tilde{\mathcal{O}}(|\mathcal{S}|^3|\mathcal{A}|/\epsilon^2)$ with AdaDO's complexity $\tilde{\mathcal{O}}(2k\sqrt{|\mathcal{A}||\mathcal{S}|}X^{\frac{3}{2}}/\epsilon + k|\mathcal{A}|X^3/\epsilon^2)$, where X is the size of the last restricted games, which empirically tend to small if the support of NEs in the original game is small. When it is satisfied that $\tilde{\mathcal{O}}(|\mathcal{S}|^3) > \tilde{\mathcal{O}}(kX^3)$, CFR has a higher sample complexity compared to AdaDO, i.e. $\tilde{\mathcal{O}}(|\mathcal{A}||\mathcal{S}|^3/\epsilon^2) > \tilde{\mathcal{O}}(kX^3|\mathcal{A}|/\epsilon^2)$. Specifically, since the first term of AdaDO's complexity $\tilde{\mathcal{O}}(2k\sqrt{|\mathcal{A}||\mathcal{S}|}X^{\frac{3}{2}}/\epsilon)$ is less than CFR's complexity $\tilde{\mathcal{O}}(|\mathcal{A}||\mathcal{S}|^3/\epsilon^2)$ due to $X \leq |\mathcal{S}|$, the dominating term of sample complexities of AdaDO are less than that of CFR under this small NE support condition. Therefore, this comparison confirms that Double Oracle methods exhibit lower sample complexity when the NE support is small. This represents the first discussion in DO literature regarding the conditions under which DO outperforms regret minimization methods theoretically.

4.2 Double Oracle with Warm Starting

In this section, we propose to improve Double Oracle methods with warm starting, in order to accelerate the convergence of the algorithm. The motivation is to reduce the complexity

caused by solving k restricted games independently, as discussed at the end of Section 3.2. Since RMDO trains each of the k restricted games from scratch without transferring knowledge from previous ones, we introduce an approach that enhances convergence speed by integrating the Double Oracle framework with **Warm Starting**, a technique that transfers the regret learned in a prior restricted game to the next one for faster convergence.

When initiating a new restricted game, the standard procedure clears the regret and re-initializes the average strategy with a uniform distribution over actions. This prevents the algorithm from reusing the knowledge acquired while solving previous restricted games, effectively reducing the process to learning k independent games and giving rise to a high-order dependence on k in the sample complexity. Because the regret accumulated in a restricted game encodes the values of its different branches, clearing the regret upon entering a new restricted game forces the algorithm to relearn the values of branches it has already explored, which wastes computational resources. Warm starting mitigates this inefficiency: rather than treating the k restricted games as separate tasks, RMDO leverages prior knowledge for faster convergence.

As illustrated in Figure 3, the regret minimizer updates the regret and average strategy within the current restricted game; after $m(\cdot)$ iterations, we compute best-response (BR) actions to the restricted average-strategy (the violet, orange, and blue bars). Whenever new BR actions are found, we expand the restricted game with them. At this expansion step, instead of cold starting, we warm start the newly constructed restricted game by carrying over the regret and cumulative strategy associated with the previously explored actions. Specifically, we initialize the counterfactual regret of every action that already appeared in the previous restricted game with its previous regret. This is justified by the nesting relation of the restricted games, $\mathbf{G}_1 \subset \mathbf{G}_2 \subset \cdots \subset \mathbf{G}_k$. We likewise warm start the cumulative strategy with that of the previous restricted game. For the new actions introduced in the current restricted game, we initialize their regret and strategy with a fixed value $\varepsilon > 0$ to ensure exploration. Suppose that at iteration t new BR actions are added and a new restricted game $\mathbf{G}_j$ is constructed. We then initialize the regret of each $s, a \in \mathbf{G}_j$ as

$$R_i^t(s,a) = \begin{cases} R_i^{t-1}(s,a), & s,a \in \mathbf{G}_{j-1}, \\ \varepsilon, & \text{otherwise.} \end{cases} \tag{24}$$

Although warm starting is designed specifically to reduce the complexity caused by k, theoretical improvements are difficult to establish because the action and infoset spaces are inconsistent across restricted games. Unlike warm starting in CFR (Brown and Sandholm, 2016), where the warm-started game remains fixed and the regret bound provably improves, DO with warm starting does not readily admit a theoretical complexity improvement due to the dynamically changing restricted game. Nevertheless, in Section 5 we observe empirically across a range of benchmark games that warm starting accelerates convergence significantly.

Replacing cold starting with warm starting implicitly mitigates the sample complexity caused by k. A second contributor to the sample complexity of RMDO is X, which arises from the need to traverse the entire restricted game tree during regret minimization. This cost becomes especially pronounced in large games, where even a restricted game can be substantial in size. To address it, we propose a scalable RMDO framework, Stochastic

Regret-Minimizing Double Oracle. This framework employs stochastic regret minimization (e.g. MCCFR) together with approximate best-response methods, in order to overcome the scalability limitations of RMDO and enable more efficient handling of large-scale games.

4.3 Stochastic Regret-Minimizing Double Oracle

The core idea behind **Stochastic Regret-Minimizing Double Oracle (SRMDO)** is to replace regret minimization in the RMDO framework with stochastic regret minimization (SRM) for the restricted game solving. While computing the oracle Best Response (BR) still requires traversing the entire game tree, the number of times computing BR is significantly less than that of executing regret minimization. We also introduce the sample complexity associated with leveraging approximate BR in SRMDO to enhance its scalability. Importantly, we demonstrate that replacing the regret minimizer with Monte-Carlo Counterfactual Regret Minimization (MCCFR) can help AdaDO reduce the theoretical sample complexity by $\mathcal{O}(X)$.

In this paper, we employ outcome-sampling Monte-Carlo Counterfactual Regret Minimization (MCCFR) (Lanctot et al., 2009) as the stochastic regret minimizer. This choice is made to ensure maximum scalability, as outcome-sampling MCCFR is the only method in the CFR family that learns from bandit feedback. Regarding the approximate best responder, any method, including Reinforcement Learning or No-Regret Learning, can be applied. However, it must be capable of reaching high precision δ-Best Response (with high probability), where $\delta < \epsilon$. The complexity of computing one iteration of approximate Best Response computation is denoted as $\mathcal{O}(H)$.

Theorem 16 (SRMDO)**.** *Suppose that at iteration T, there are k restricted games. Then, with probability at least $1 - pk$, the Last-Window Average Strategy (LAS) produced by SRMDO reaches an ϵ-NE when using an oracle Best Response. The sample complexity required is as follows:*

$$\tilde{\mathcal{O}}(kH|\mathcal{A}|P^2X^2/\epsilon^2 + \sum_{j=1}^{k} H_j m(j) + |\mathcal{A}||\mathcal{S}|P^2X^2/\epsilon^2 m(j)), \tag{25}$$

and the following sample complexity when employing δ-BR:

$$\tilde{\mathcal{O}}(kHP^2|\mathcal{A}|X^2/(\epsilon-\delta)^2 + \sum_{j=1}^{k} H_j m(j) + H|\mathcal{A}|P^2X^2/(\epsilon-\delta)^2 m(j)), \tag{26}$$

if $\delta < \epsilon$, where $p \in (0, 1]$, $P = 1 + \sqrt{\frac{2}{p}}$, $H_j = \max_i H_{i,j}$ is the depth of the restricted game tree of $\mathbf{G}_j$, and the runtime complexity of δ-BR is $\tilde{\mathcal{O}}(|H|)$.

The sample complexity of Stochastic Regret-Minimizing Double Oracle (SRMDO) has been significantly reduced by decreasing the power of X in the term that is independent of the frequency function from 3 to 2. This reduction is crucial, as even with the optimal choice of frequency function, the power of the first term cannot be influenced. Additionally, it is observed that employing approximate Best Response can further reduce the remaining term from $\tilde{O}(X^3)$ to $\tilde{O}(X^2)$. However, the benefit introduced by approximate Best Response is

relatively small, as selecting an appropriate frequency function can easily achieve the same power reduction of the later two terms in Equation (25). Therefore, for the remainder of this paper, we will use oracle Best Response for simplicity.

It is also reasonable to apply periodic and adaptive frequency functions in SRMDO for $m(j)$, which we refer to as **Stochastic Periodic Double Oracle (SPDO)** and **Stochastic Adaptive Double Oracle (SADO)**, respectively. The extension of PDO to SPDO and won't be discussed in detail here. To get the sample complexity of SPDO, we only need to set $m(j) = c$ in Equation (25). We demonstrate SADO here since it requires a new adaptive function that can help decrease the power of X in all the dominating terms from 3 to ≤ 2.

Theorem 17 (SADO)**.** *For any $p \in (0, 1]$, and $P = 1 + \sqrt{\frac{2}{p}}$, By employing exact Best Response and the following frequency function*

$$m(j) = \frac{\sqrt{|\mathcal{S}|}P}{\epsilon} \cdot \sqrt{\frac{|\mathcal{A}_j|(\sum_i |\mathcal{S}_{i,j}|)^2}{H_j}}. \tag{27}$$

and the LAS of SRMDO with probability at least probability $1 - pk$ to reach ϵ-NE, requiring the following the sample complexity:

$$\tilde{\mathcal{O}}(2\sqrt{|\mathcal{A}||\mathcal{S}|}kPX/\epsilon + k|\mathcal{A}|HX^2/\epsilon^2), \tag{28}$$

which also matches the lower bound of the sample complexity of SRMDO with various frequency functions.

Theorem 17 implies that despite the necessity for the full traversal of the game tree with oracle Best Response, the complexity of Stochastic Regret-Minimizing Double Oracle (SRMDO) with an appropriate adaptive frequency function can still be reduced by $\mathcal{O}(X)$. In practice, similar to AdaDO, we still leverage a discount factor α, applying a practical frequency $m(j) = \alpha\sqrt{|\mathcal{A}_j|(\sum_i |\mathcal{S}_{i,j}|)^2/H_j}$.

5 Experiments

In this section, we first examine the efficiency of Periodic Double Oracle (PDO) compared to other baselines including XDO, XODO, and regret minimization methods for game solving. We also analyze the support of PDO in different games. Subsequently, we explore the effect of warm starting and Adaptive Double Oracle (AdaDO). Finally, we delve into the empirical performance of Stochastic Regret-Minimizing Double Oracle, encompassing Stochastic Periodic Double Oracle (SPDO) and Stochastic Adaptive Double Oracle (SADO).

To assess the efficiency in game solving, we evaluate performance mainly through plots of exploitability (the distance to Nash Equilibrium) versus the number of visited nodes. Due to different frequency of computing exploitability across algorithms, some curves might not have consistent ending in the number of visited nodes. Such plots offer a clear visualization of the required complexity to achieve specific precision of Nash Equilibrium. Here we only include the visited nodes in the algorithms including both (stochastic) regret minimization and best response computation, but exclude the complexity in computing the exploitability for fair comparison. Additionally, in the stochastic regret minimization setting, we provide

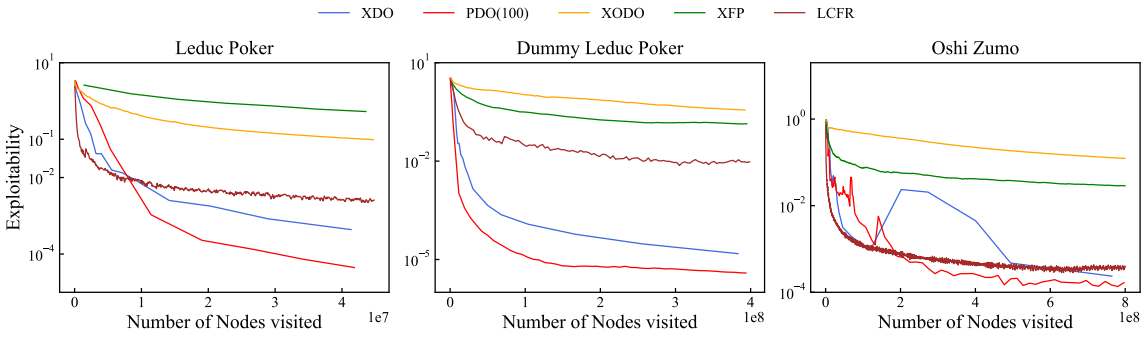

Figure 4: Exploitability-Visited Nodes Performance of Extensive-form Double Oracle (XDO), Periodic Double Oracle with its periodicity, i.e. PDO(c), Extensive-form Online Double Oracle (XODO), Extensive-form Fictitious Self-Play (XFP), and Linear Counterfactual Regret Minimization (LCFR). Our algorithm PDO achieves the lower exploitability than any other methods.

the number of visited nodes required to reach a specific level of exploitability for different algorithms.

We conducted tests on perfect and imperfect-information extensive-form poker games, including Blotto, Kuhn Poker, Leduc Poker, and their variants, namely Large Kuhn Poker, Leduc Poker Dummy, and Leduc Poker with 10 cards. Detailed descriptions of the games can be found in Appendix D. The implementation is primarily based on the library OpenSpiel (Lanctot et al., 2019). The selected baselines include Extensive-Form Fictitious Self-Player (XFP) (Heinrich et al., 2015) and Linear Counterfactual Regret Minimization (LCFR) (Brown and Sandholm, 2019a) for regret minimization setting, and Outcome-Sampling Monte-Carlo Counterfactual Regret Minimization (simplified as MCCFR) for the stochastic regret minimization setting.

5.1 Periodic Double Oracle

We initially examine the performance of well-tuned Periodic Double Oracle (PDO) in comparison to baselines and existing Regret-Minimizing Double Oracle (RMDO) instances (Figure 4). Our findings reveal that PDO surpasses XDO, XODO, and baseline methods (XFP and LCFR) by a significant margin in games such as Leduc Poker, Leduc Poker Dummy, and Oshi Zumo. PDO has a more stable exploitability curve and a faster convergence in general compared to XDO. Specifically, PDO achieves 10^{-4} exploitability faster than any other methods, even though it may not perform as well as LCFR in the early stages of Leduc Poker and Oshi Zumo. In summary, PDO is capable of converging faster to lower exploitability.

We then study the support of Double Oracle (DO) methods by investigating the support of average strategies of the well-tuned PDO when reaching ϵ-NE, where $\epsilon = 10^{-3}$. The metrics include the minimum support percentage, defined as $\min_{s \in S} \text{supp}^{\bar{\pi}}(s)/|A(s)|$, and Average Support Percentage $\sum_{s \in S} \text{supp}^{\bar{\pi}}(s)/|A(s)||\mathcal{S}|$.

The analysis of the minimum support in Double Oracle (DO) strategies reveals a noteworthy efficiency compared to Linear CFR. The average strategy of Linear CFR consis-

	Large Kuhn Poker		*Leduc Poker*		*Kuhn Poker*		*Oshi Zumo*	
	PDO	**LCFR**	**PDO**	**LCFR**	**PDO**	**LCFR**	**PDO**	**LCFR**
Min. Support	50%	100%	33%	100%	50%	100%	25%	100%
Avg. Support	76%	100%	85%	100%	83%	100%	93%	100%

Table 2: Table of minimum and average support percentages of well-tuned Periodic Double Oracle (PDO) when first reaching 10^{-3}-NE. Specifically, the minimum support percentage is defined as $\min_{s \in S} \text{supp}^{\bar{\pi}}(s)/|A(s)|$, and Average Support Percentage $\sum_{s \in S} \text{supp}^{\bar{\pi}}(s)/|A(s)||\mathcal{S}|$. DO methods learn more sparse strategy (i.e. with less support).

tently maintains full support, implying that it assigns non-zero probability to every action. In contrast, DO achieves significantly lower minimum support percentages. This efficiency is indicative of the learning process, as a lower support implies the need to learn the distribution over fewer actions, facilitating faster convergence. Additionally and notably, this table also implies that DO tends to produce more sparse solution (i.e. with less actions with positive probability).

While the mean support may not exhibit a substantial difference from that of Linear CFR, it's crucial to note that this metric operates at the infoset level. In the broader context of the entire game tree of an Extensive-Form Game (EFG), actions excluded from the strategy contribute to pruning entire subtrees rooted by them. This efficient pruning mechanism is expected to enhance sample complexity significantly, showcasing the reason to the efficiency DO in games with small support NE.

5.2 Adaptive Double Oracle and Warm Starting Double Oracle

The evaluation of Adaptive Double Oracle (AdaDO) and warm starting (simplified as suffix -WS) in Figure 5 provides insights into their efficiency compared to PDO, Linear CFR, and CFR+ in different poker games. For PDO, we directly use a well-tuned periodicity $c = 100$.

We first introduce the performance of AdaDO in Figure 5. In Blotto, Leduc Poker and Leduc Poker Dummy, and large Kuhn Poker, the exploitability of AdaDO decreases faster than that of PDO and LCFR. In other games, the exploitability of AdaDO converges to the same magnitude of exploitability to PDO. AdaDO performs better than PDO in most research games and significantly outperforms LCFR and CFR+ in all experiments. It is impressive to observe that AdaDO, not only avoid extensive tuning on the sensitive hyperparameter periodicity in PDO, but can perform faster convergence than PDO and other baselines as well. Since in Figure 4, PDO outperforms other baselines already, AdaDO is now the more efficient method that has the state-of-the-art performance.

We then investigate the effectiveness of warm starting in Figure 5. In Blotto, Leduc Poker 10 cards and Large Kuhn Poker, warm starting significantly reduces the exploitability of Double Oracle methods. Especially in Large Kuhn Poker, both PDO and AdaDO with warm-starting exhibit early drops in exploitability, reaching values over 10^8 less than those of Linear CFR. This suggests that warm starting can accelerate the convergence of Double Oracle. In other games, warm starting has little impact on DO methods. In summary, warm starting overall has positive influence, and in specific games can help reduce

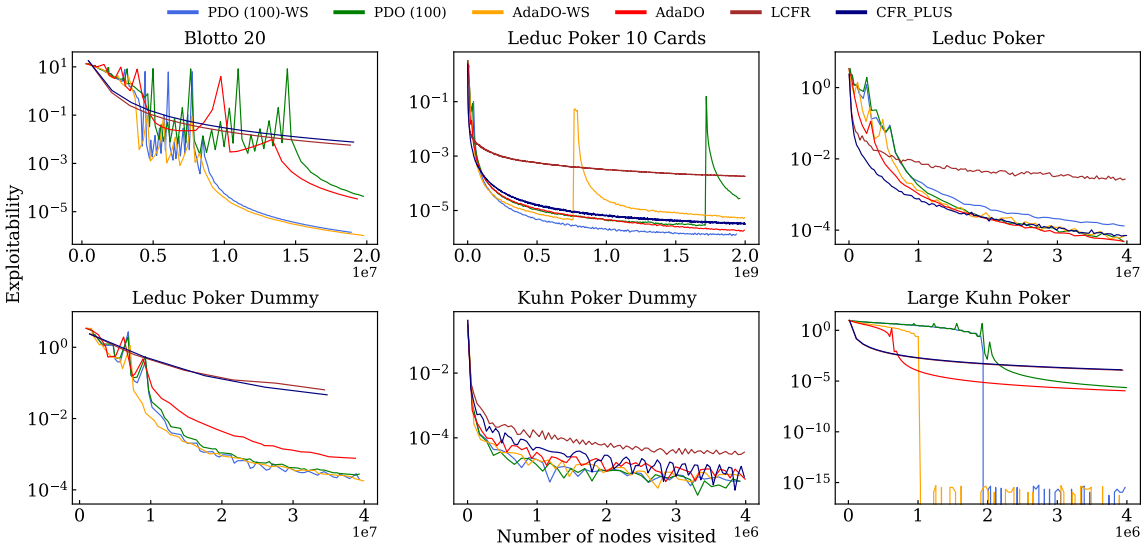

Figure 5: Exploitability-Visited Nodes Performance of PDO with and without warm starting, AdaDO with and without warm starting, and LCFR. Warm starting reduces exploitability significantly in Blotto and Large Kuhn Poker. AdaDO and PDO outperforms LCFR.

the exploitability significantly faster and at most 8 levels of magnitude less exploitable, i.e. higher precision.

AdaDO, leveraging a theoretically optimal frequency function, performs faster convergence than a well-tuned PDO in most games. The impact of warm starting is positive in most games, being more pronounced in Large Kuhn Poker. Importantly, AdaDO achieves these results without hyperparameter tuning, highlighting its potential for efficient use.

5.3 Stochastic Regret-Minimizing Double Oracle

In this section, We present empirical assessments of the Stochastic Regret-Minimizing Double Oracle (SRMDO), which leverage Outcome-Sampling MCCFR for restricted game solving. Our baseline is also Outcome-Sampling MCCFR, thus the comparison between them is fair. We employ oracle Best Response for all instances for simplicity

We study the performance of methods of SRMDO including Stochastic Periodic Double Oracle (SPDO) and Stochastic Adaptive Double Oracle (SADO) in Figure 6. In Blotto games, MCCFR experiences fast exploitability reduction in the early stages but gets stuck at low precision Nash Equilibrium. On the other hand, SPDO and SADO shows a significant improvement in exploitability at later stages, bringing its strategy less exploitable compared to MCCFR. Specifically, in all experiments, SPDO with periodicity 5000 and SADO converge to less exploitability than other algorithms, converging to 10 times less exploitability solution. In Blotto 20, SADO outperforms SPDO and achieves competitive performance with SPDO.

Though SADO performs similarly to SPDO in Blotto games and only outperform SPDO in Kuhn Pokers, the inefficiency of tuning periodicity hyperparameter in PDO is obvious. In Figure 6, SPDO is sensitive to this periodicity since SPDO(1000) and SPDO(5000) has

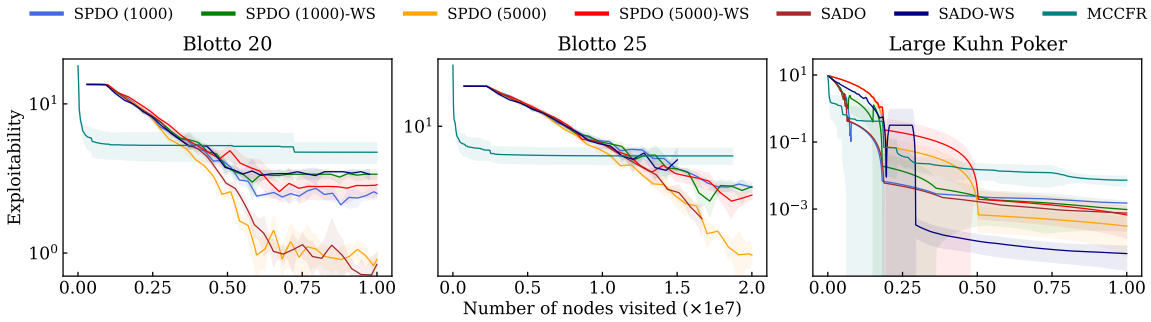

Figure 6: Exploitability experiments of Stochastic PDO (SPDO), and Stochastic Adaptive DO (SADO) with and without warm starting and Outcome-Sampling Monte-Carlo CFR (MCCFR). SPDO and SADO performs similarly good, outperforming MCCFR significantly.

clearly distinct performance. This observation is crucial for understanding the challenges associated with tuning frequency hyperparameter in PDO and SPDO. The results highlight that the optimal choice of frequency can vary significantly among different games. So retuning c is necessary when applying SPDO to new games, emphasizing the demanding nature of hyperparameter tuning in PDO. On the contrary, Stochastic Adaptive Double Oracle (SADO) is executed without extensive tuning but still demonstrates superior results.

Overall, the results highlight the effectiveness of SPDO and SADO. Additionally, the importance of hyperparameter tuning in PDO and SPDO emphasize the need for adaptive approaches like SADO to dispense the complexity of hyperparameter tuning. The potential advantages of SADO, including not requiring tuning and theoretical sample efficiency, make it a promising direction for addressing the challenges posed by extensive-form games. An observable limitation in SRMDO is that warm starting has shown limited effectiveness, outperforming other methods only on Large Kuhn Poker. We provide analysis on this observation in Appendix C

5.4 Ablation Study

In this section, we provide ablation study on the value of frequency discount α in Adaptive Double Oracle methods, including AdaDO and SADO.

In Fig. 7, AdaDO without warm-starting, as well as configurations with $\alpha = 1$ and $\alpha = 0.5$, generally exhibit superior performance. In games characterized by small-support Nash equilibria, such as Leduc Poker Dummy, less frequent best response computation (i.e., larger $\alpha = 10$) yields improved outcomes. Furthermore, warm-starting accelerates the reduction in exploitability.

In Fig. 8, SADO with $\alpha = 1$ and without warm-starting achieves the best overall performance. When warm-starting is applied, SADO with $\alpha = 1$ performs well in Blotto games, whereas SADO with $\alpha = 10$ achieves the fastest reduction of exploitability in Large Kuhn Poker. Warm-starting markedly improves performance in solving Large Kuhn Poker.

We provide further ablation study on warm starting by showing the average performance of RMDO with and without warm starting (Figure 9) and SRMDO methods (Figure 10).

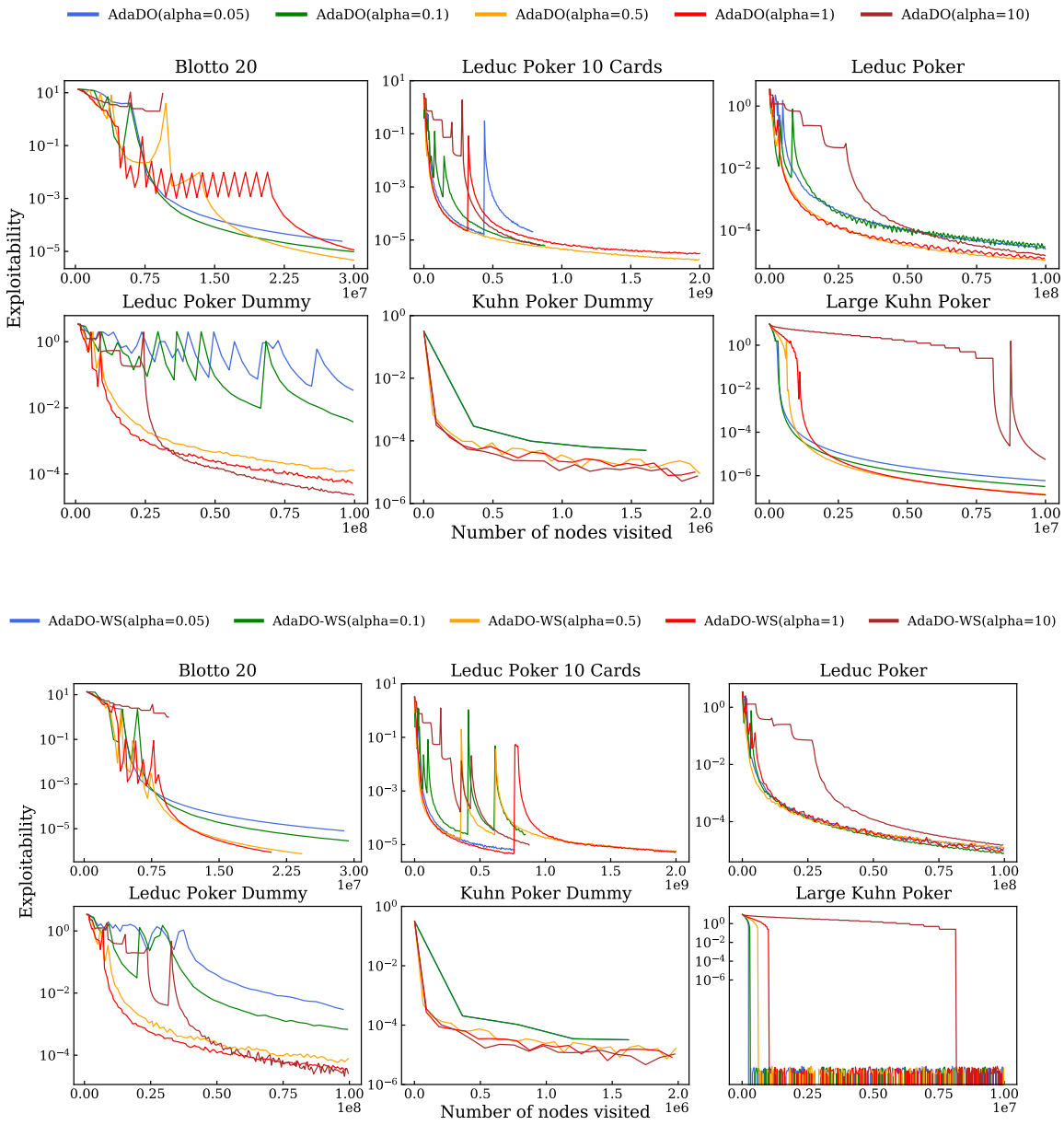

Figure 7: Ablation Study of frequency discount α in AdaDO.

In most games, warm starting help reduce the exploitability, significantly in Large Kuhn Poker and Leduc Poker games.

6 Conclusion

In this paper we propose Regret-Minimizaing Double Oracle (RMDO) to unify the existing Double Oracle algorithms combined with regret minimization for theoretical study and further algorithmic improvement. Based on RMDO framework, we derive the sample com-

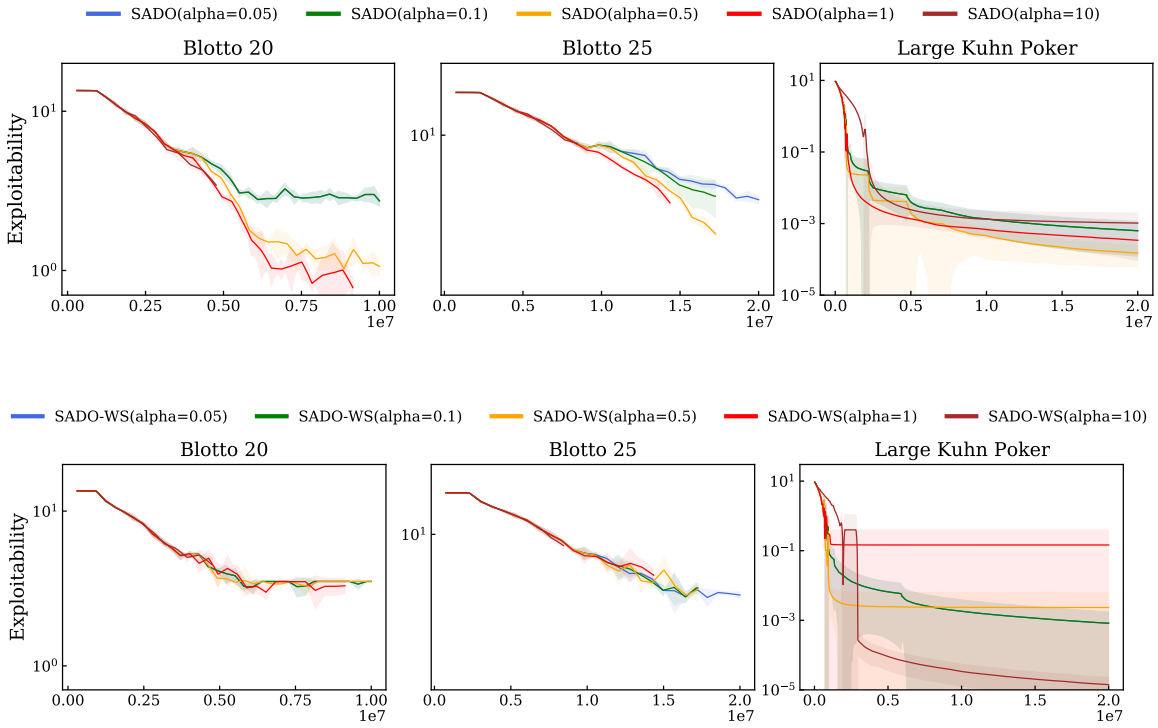

Figure 8: Ablation Study of frequency discount α in SRMDO.

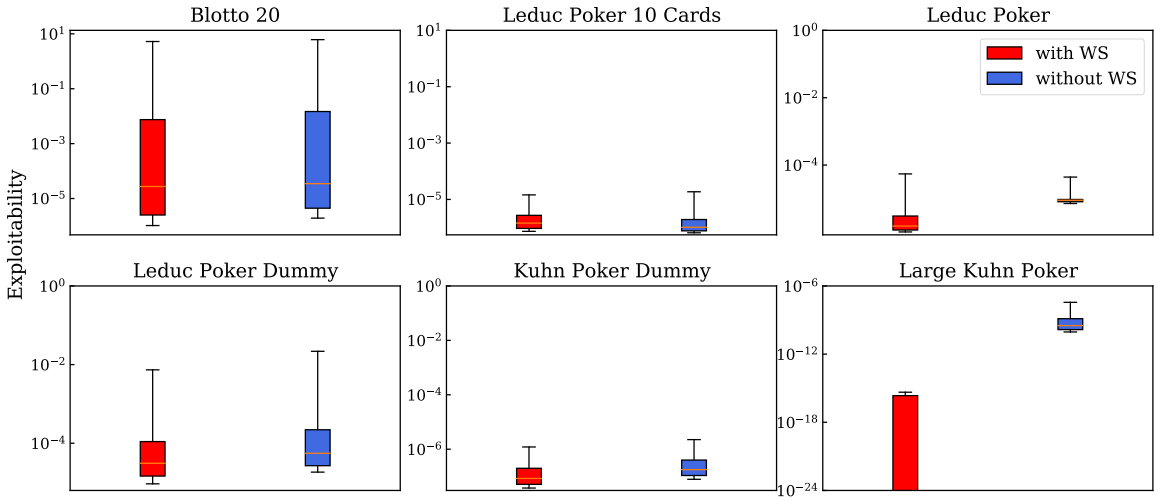

Figure 9: Ablation Study on average exploitability with and without applying warm starting in RMDO methods.

plexity of existing methods Online Double Oracle and Extensive-Form Double Oracle, and prove that the later method, which is the state-of-the-art method for EFGs, suffers from exponential sample complexity. To address this problem, we propose two instances that only has polynomial sample complexity—Periodic Double Oracle, which requires hyperpa-

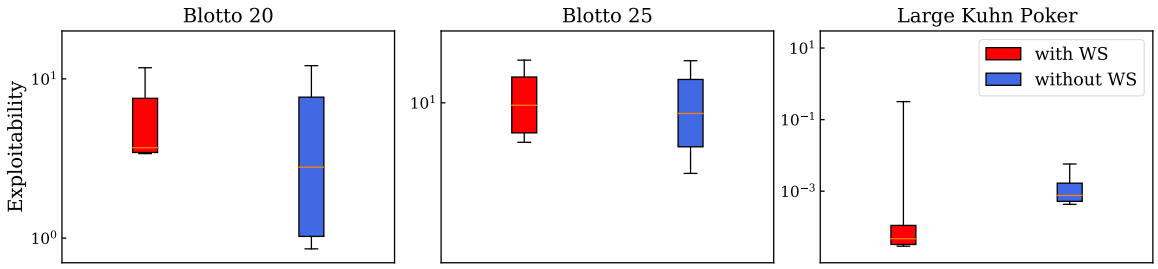

Figure 10: Ablation Study on average exploitability (↓) with and without applying warm starting in Stochastic RMDO methods.

rameter tuning, and the most sample-efficient instance Adaptive Double Oracle. These two innovative DO methods are provably more sample-efficient than the previous DO methods. To further make RMDO more sample efficient and scalable, we propose to adopt warm starting and stochastic regret minimizer for restricted game solving. In the empirical assessments, PDO can converge to lower exploitability than regret minimization methods including LCFR and CFR+ and previous DO methods. AdaDO outperforms a well-tuned PDO in most environments, and exhibits accelerated convergence in exploitability and reach the state-of-the-art performance in most games combined with warm starting and stochastic regret minimizer.

Appendix A. Notations

We offer the table of notations for easier understanding the proofs in the next sections.

Table 3: Table of notations.

Notations	Descriptions
s	Information State/Information Set
a	Action
i	Player
j	Index of time windows/restricted games
t	Iteration/time
π	Strategy/Policy
π_i^t	Strategy/Policy of player i in iteration t
$e(\pi)$	Exploitability of π
$R_i^t(s,a)$	Cumulative regret of player i at iteration t at infoset s taking action a
$\bar{\pi}^t(s,a)$	Average strategy of player i in iteration t
$\mathcal{A}(s)$	The set of action at infoset s
$\mathcal{S}_i$	The set of infosets of player i in the original game
$\mathcal{A}_j(s)$	The set of action at infoset s in time window j
$\mathcal{S}_{i,j}$	The set of infosets of player i in time window j
X	$\max_{s\in S,\pi^*\in\Pi^*}\sum_s \text{supp}^{\pi^*}(s)$
$H_{i,j}$	The largest length of all trajectories of player i in T_j in Stochastic RM
T	Current time or current iterations of training
T_j	The j-th time window
$\mathbf{G}_j$	The j-th restricted game, the restricted game in T_j
k	Number of time windows (restricted games)
$m(\cdot)$	Frequency function

Appendix B. Proofs

Lemma 18 (Local NE to Original NE)**.** *Suppose* **BR** *denotes the best response (BR) operator in the original game, and BR actions to the approximate Nash Equilibrium (ϵ-NE) of a restricted game which is in the action population Π of a restricted game, i.e.* $\mathbf{BR}(\pi^*) \in \Pi$. *Then the ϵ-NE of the restricted game is also the ϵ-NE of the original game.*

Proof Since **BR** is the global best response operator, i.e. the best response in the original game. Then $\mathbf{BR}(\pi^*)$, BR to the local ϵ-NE are also in the current population Π_j. Denote the the local ϵ-NE by π^*, the value of this local optimal satisfies $v_i(\pi^*) \geq v_i(\mathcal{BR}(\pi^*)) - \epsilon = v_i(\mathbf{BR}(\pi^*)) - \epsilon$, where $\mathcal{BR}$ is the local best response operator. This inequality is exactly the condition of the ϵ-NE in the original game. ■

Lemma 9. $\min_{\pi\in\Pi^*}\max_{s\in S} supp^{\pi}(s) < k \leq \sum_i |S_{i,k}| \leq X \leq |\mathcal{S}| = \sum_i |\mathcal{S}_i|$.

Proof We provide the proof of the inequalities above from left to right:

- Since there is at most one action at a infoset added to the population every time when the restricted game is expanded and a new window starts. Thus at $s \in S$, $|\{a|a \in A(s)\}| \leq k$. Since at every infoset, the number of pure strategies in the converged population will be greater than the support of NE. Otherwise, there is a action in the NE strategy but not in the population, which means that the population doesn't converge and leads to contradiction. At $s \in S$, denote Π^* as a set of NE of this game, $|\{a|a \in A(s)\}| \geq \min_{\pi \in \Pi^*} \text{supp}^{\pi}(s)$. So we have $k \geq \max_{s \in S} |\{a|a \in A(s)\}| \geq \max_{s \in S} \min_{\pi \in \Pi^*} \text{supp}^{\pi}(s)$.
- The number of different restricted games is less than or equal to the number of infosets in the largest possible restricted game since at least one new action added, i.e. $\forall j = 1, \cdots, k-1$, $\sum_i |S_{i,j+1}| - \sum_i |\mathcal{S}_{i,j}| \geq 1$. Therefore, $\sum_i |S_{i,k}| \geq k$.
- It's easy to see that X is the number of infosets in the largest possible restricted game constructed by NE. Thus, for any restricted game, the number of the infosets is smaller than or equal to X.
- And since such game is constructed by the actions from the original game, it is less than the number of the infosets in the original games.

■

Lemma 2. *Given the average regret of an algorithm $\mathcal{O}(|\mathcal{S}_i|\sqrt{|\mathcal{A}_i|}/\sqrt{T})$, the average strategy of this algorithm is a $\mathcal{O}(|\mathcal{S}|\sqrt{|\mathcal{A}|})/\sqrt{T})$-NE.*

Proof Since the average strategy at time T is defined as $\bar{\pi}_i = \sum_{t=1}^{T} \pi_i^t / T$. Since the value of strategy $v_i(\cdot, \cdot)$ is a linear function, then we have:

$$\max_{\pi_i^{'}} \sum_{t=1}^{T} [v_i(\pi_i^{'}, \pi_{-i}^t) - v_i(\pi_i^t, \pi_{-i}^t)]/T = \max_{\pi^{'}} v_i(\pi^{'}, \bar{\pi}_{-i}^t) - v_i(\bar{\pi}_i, \bar{\pi}_{-i}) \leq \mathcal{O}(|\mathcal{S}_i|\sqrt{|\mathcal{A}_i|}/\sqrt{T}) \tag{29}$$

$$\leq \mathcal{O}(|\mathcal{S}_i|\sqrt{|\mathcal{A}|}/\sqrt{T}) \tag{30}$$

Given the above inequality exists for both i, then in two-player zero-sum game setting, we have:

$$v_{-i}(\cdot, \cdot) = -v_i(\cdot, \cdot), \ \max_{\pi_{-i}^{'}} v_{-i}(\pi_i^t, \pi_{-i}^{'}) = -\min_{\pi_{-i}^{'}} v_i(\bar{\pi}_i, \pi_{-i}^{'})$$

and then

$$v_i(\bar{\pi}) - \min_{\pi_{-i}^{'}} v_i(\bar{\pi}_i, \pi_{-i}^{'}) \leq \mathcal{O}(|S_{-i}|\sqrt{|\mathcal{A}|}/\sqrt{T}). \tag{31}$$

Therefore:

$$\max_{\pi^{'}} v_i(\pi^{'}, \bar{\pi}_{-i}^t) - \sum_i \mathcal{O}(|\mathcal{S}_i|\sqrt{|\mathcal{A}|}/\sqrt{T}) \leq \max_{\pi^{'}} v_i(\pi^{'}, \bar{\pi}_{-i}^t) - \mathcal{O}(|\mathcal{S}_i|\sqrt{|\mathcal{A}|}/\sqrt{T})$$

$$\leq v_i(\bar{\pi}) \leq \min_{\pi_{-i}^{'}} v_i(\bar{\pi}_i, \pi_{-i}^{'}) + \mathcal{O}(|S_{-i}|\sqrt{|\mathcal{A}|}/\sqrt{T}) \leq \min_{\pi_{-i}^{'}} v_i(\bar{\pi}_i, \pi_{-i}^{'}) + \sum_i \mathcal{O}(|S_{-i}|\sqrt{|\mathcal{A}|}/\sqrt{T}), \tag{32}$$

$\bar{\pi}$ is ϵ_T-NE, where $\epsilon = \sum_i \mathcal{O}(|\mathcal{S}_i|\sqrt{|\mathcal{A}|}/\sqrt{T}) = \mathcal{O}(|\mathcal{S}|\sqrt{|\mathcal{A}|}/\sqrt{T})$. ■

Theorem 19 ((Tang et al., 2023)). *In RMDO, suppose the regret minimizer has $\tilde{\mathcal{O}}(|\mathcal{S}_i|\sqrt{|\mathcal{A}|T})$ regret, the weighted-average regret bound of RMDO:*

$$\tilde{\mathcal{O}}(\sum_{j=0}^{k-2} \frac{|T_j|}{T} \cdot [m(j)-1] + \sum_{j=0}^{k-1} \frac{\sqrt{k}|\mathcal{S}_{i,j}||T_j|}{T\sqrt{|T_j|-m(j)+1}}), \tag{33}$$

converges to 0 if $m(j)$ is sublinear in T.

Proof Please refer to the proof of Theorem 3.3 in our previous work (Tang et al., 2023). The only difference is we replace S_i with $S_{i,j}$, which can be easily shown reasonable. ■

Lemma 20 (Iteration Complexity). *It requires in the worst case the following number of iterations for LAS of RMDO to reach ϵ-NE:*

$$\sum_{j=1}^{k} \tilde{\mathcal{O}}(A_j(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 + m(j) - 1). \tag{34}$$

Proof We first prove that, before reaching the ϵ-NE of the original game, in each windows T_j, $j < k-1$, the number of iterations of the last-window average strategy $|T_j| < \tilde{\mathcal{O}}(A_j(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 + m(j) - 1)$ if the regret minimizer has $\tilde{\mathcal{O}}(\sqrt{A_{i,j}}|\mathcal{S}_{i,j}|/|T_j|)$ regret, where $A_j = \max_{i\in\mathcal{P}, s\in S} |\mathcal{A}_{i,j}(s)|$. $A_{i,j} = \max_{s\in S_{i,j}} |\mathcal{A}_{i,j}(s)|$ and $S_{i,j}$ are defined as usual, the set of actions and information sets, respectively; the index j here denotes the time window. The quantity $m(j)-1$ comes from the fact that we conduct BR computation every $m(j)$ iterations. So in the worst case, the algorithm runs for additional $m(j)-1$ iterations before knowing its convergence via BR computation. We prove this claim by contradiction:
Claim: For $j = 1, 2, \cdots, k-1$, $|T_j| < \tilde{\mathcal{O}}(A_j(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 + m(j) - 1)$.
Proof: Suppose the claim above is not correct, which is $|T_j| \geq \tilde{\mathcal{O}}(A_j(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 + m(j) - 1)$, thus the average strategy in current window already reaches the restricted game's local ϵ-NE. Then there must exist a BR computing right after $\tilde{\mathcal{O}}(A_j(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 + m(j) - 1)$ iterations of regret minimization in current window. Suppose this iteration is the t-th one. Since $\mathbf{BR}(\bar{\pi}^t) \in \Pi_j$ and $\bar{\pi}^t$ has reached restricted game's local ϵ-NE, $\bar{\pi}^t$ is also the ϵ-NE in the original game (Lemma 18), which causes contradiction since in all windows T_j, where $j < k$, the average strategy in these windows is not supposed to reach ϵ-NE. Therefore the claim is correct. ■

Then we prove the rest of Lemma 20. For $j \leq k-1$, regret minimizer in the worst case needs $\tilde{\mathcal{O}}(|A_k|(\sum_i |S_{i,k}|)^2/\epsilon^2 + m(j) - 1)$ iterations to reach ϵ-NE. Since we have upper bound of number of iterations in each window before the last one (T_k), we sum up the required iterations in all time window, and get that in the worst case the required number of iterations (i.e. iteration complexity) for the LAS of RMDO to reach ϵ-NE is:

$$\sum_{j=1}^{k} \tilde{\mathcal{O}}(A_j(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 + m(j) - 1). \tag{35}$$

■

Theorem 10. *The sample complexity of last-window average strategy of RMDO to reach ϵ-NE is:*

$$\sum_{j=1}^{k} \tilde{\mathcal{O}}\Big(A_j|\mathcal{S}|(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 m(j) + A_j(\sum_i |\mathcal{S}_{i,j}|)^3/\epsilon^2 + \sum_i |\mathcal{S}_{i,j}|m(j)\Big). \tag{36}$$

Proof In general, there are two parts contribute to the complexity, regret minimization and Best Response computing. We will compute the sample complexity of both separately.

Regret Minimization. We have proved the required iterations in Lemma 20, and since the sample complexity of each iteration is $\sum_i |\mathcal{S}_{i,j}|$, then we have the complexity of regret-minimization part:

$$\sum_{j=1}^{k} \tilde{\mathcal{O}}(A_j(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 + m(j) - 1) \cdot \sum_i |\mathcal{S}_{i,j}|. \tag{37}$$

Best Response Computation. Then we compute the sample complexity in BR computing. In T_j, $j = 1, 2, \cdots, k$, we compute BR every $m(j)$ iterations of regret minimization, thus by multiplying total times of BR computation and the complexity in each computation $\tilde{\mathcal{O}}(|\mathcal{S}|)$, we can get the BR-computing part of sample complexity:

$$\sum_{j=1}^{k} \tilde{\mathcal{O}}(A_j(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 m(j) + 1 - 1/m(j))) \cdot \tilde{\mathcal{O}}(|\mathcal{S}|) \tag{38}$$

Sum up the above two parts of the complexity we can get the final complexity:

$$\begin{aligned}
&\sum_{j=1}^{k} \tilde{\mathcal{O}}(A_j(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 m(j) + 1 - 1/m(j))) \cdot \tilde{\mathcal{O}}(|\mathcal{S}|) \\
+ &\sum_{j=1}^{k} \tilde{\mathcal{O}}(A_j(\sum_i |\mathcal{S}_{i,j}|)^3/\epsilon^2 + \sum_i |\mathcal{S}_{i,j}|m(j) - \sum_i |\mathcal{S}_{i,j}|), \\
\leq &\sum_{j=1}^{k} \tilde{\mathcal{O}}\Big(A_j|\mathcal{S}|(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 m(j) + A_j(\sum_i |\mathcal{S}_{i,j}|)^3/\epsilon^2 + \sum_i |\mathcal{S}_{i,j}|m(j)\Big)
\end{aligned} \tag{39}$$

We take the upper bound as the sample complexity, since if the number of sample points of the algorithm reach RHS of inequality (39), we can guarantee that the LAS of RMDO reaches ϵ-NE. This is common way of analyzing the theoretical sample complexity in game solving (Bai et al., 2022).

We further simplify the sample complexity of RMDO's LAS. We have $\forall j = 1, 2, \cdots, k$, $|\mathcal{A}_j| \leq |\mathcal{A}|$ and $\sum_i |\mathcal{S}_{i,j}| \leq X$. Then according to Lemma 9, the sample complexity for RMDO's LAS to reach ϵ-NE is

$$\sum_{j=1}^{k} \tilde{\mathcal{O}}\Big(A_j|\mathcal{S}|(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 m(j) + A_j(\sum_i |\mathcal{S}_{i,j}|)^3/\epsilon^2 + \sum_i |\mathcal{S}_{i,j}|m(j)\Big). \tag{40}$$

■

Proposition 11. *The sample complexity for XODO to reach ϵ-NE is $\tilde{\mathcal{O}}(2X^3k^2/\epsilon^2)$.*

Proof Setting $m(j) = 1$ in Theorem 19, the regret bound will be $\tilde{\mathcal{O}}(|S_{i,k}|\sqrt{k}\sum_j \sqrt{|T_j|}/T)$. According to Cauchy-Schwartz inequality, $\sum_j \sqrt{|T_j|} \leq \sqrt{k\sum_j |T_j|} = \sqrt{kT}$, then the upper bound becomes $\tilde{\mathcal{O}}(|S_{i,k}|k/\sqrt{T})$. Thus the required iteration T to reach ϵ-NE satisfies:

$$\sum_i \tilde{\mathcal{O}}(|S_{i,k}|k/\sqrt{T}) \leq \epsilon. \tag{41}$$

Therefore, the number of required iterations to obtain ϵ-NE is:

$$T \geq \tilde{\mathcal{O}}(k^2X^2/\epsilon^2). \tag{42}$$

Since XODO computes BR in each iteration and the sample complexity of one BR computation is $\tilde{\mathcal{O}}(|\mathcal{S}|)$, the sample complexity of XODO becomes:

$$\tilde{\mathcal{O}}\Big(\frac{k^2X^3 + k^2X^2|\mathcal{S}|}{\epsilon^2}\Big). \tag{43}$$

■

Proposition 12. *XDO is an instance of RMDO. In the worst case, its frequency function $m(j) = 4^j|\mathcal{S}_{i,j}|^2A_{i,j}/\epsilon_0^2$, where $j = 0, 1, \cdots, k-1$. Thus the sample complexity bound of XDO to reach ϵ-NE is*

$$\tilde{\mathcal{O}}(k|\mathcal{A}|X^3/\epsilon^2 + |\mathcal{A}|X^3 4^k/\epsilon_0^2). \tag{44}$$

Proof The proof of this proposition is similar to XDO sample complexity derivation in our previous work (Tang et al., 2023). Since XDO requires in the j-th restricted game to obtain a ϵ_0/α^j-NE, in the worst-case, XDO is equivalent to RMDO with frequency function $m(j) = |\mathcal{A}_j|(\sum_i |\mathcal{S}_{i,j}|)^2 \cdot \alpha^{2k}/\epsilon_0^2$ ($\alpha = 2$ according to (McAleer et al., 2021)). Then the sample complexity of XDO becomes

$$\sum_{j=1}^{k} \tilde{\mathcal{O}}\Big(A_j|\mathcal{S}|(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 m(j) + A_j(\sum_i |\mathcal{S}_{i,j}|)^3/\epsilon^2 + \sum_i |\mathcal{S}_{i,j}|m(j)\Big) \tag{45}$$

$$= \tilde{\mathcal{O}}\Big(|\mathcal{S}|\epsilon_0^2/\epsilon^2 4^k + k|\mathcal{A}|X^3/\epsilon^2 + |\mathcal{A}|X^3 4^k/\epsilon_0^2\Big). \tag{46}$$

Since $|\mathcal{S}|\epsilon_0^2/\epsilon^2 4^k$ has low order of magnitude, we can omit it and report the remaining part. Then Equation (46) becomes

$$\tilde{\mathcal{O}}\Big(k|\mathcal{A}|X^3/\epsilon^2 + |\mathcal{A}|X^3 4^k/\epsilon_0^2\Big). \tag{47}$$

■

Proposition 15. *AdaDO is an instance of RMDO when $m(j)$ satisfies Equation (21). Thus the sample complexity of AdaDO matches the sample complexity* ***lower bound*** *of RMDO sample complexity for all frequency function, which is:*

$$\tilde{\mathcal{O}}\Big(2k\sqrt{|\mathcal{A}||\mathcal{S}|}X^{\frac{3}{2}}/\epsilon + |\mathcal{A}|X^3/\epsilon^2\Big). \tag{48}$$

Proof We first provide the sample complexity lower bound. According to Theorem 10, the sample complexity of the LAS of RMDO is

$$\sum_{j=1}^{k}\tilde{\mathcal{O}}\Big(A_j|\mathcal{S}|(\sum_i|\mathcal{S}_{i,j}|)^2/\epsilon^2 m(j) + A_j(\sum_i|\mathcal{S}_{i,j}|)^3/\epsilon^2 + \sum_i|\mathcal{S}_{i,j}|m(j)\Big) \tag{49}$$

$$\geq\sum_{j=1}^{k}\tilde{\mathcal{O}}\Big(\sqrt{A_j|\mathcal{S}|}(\sum_i|\mathcal{S}_{i,j}|)^{\frac{3}{2}}/\epsilon + A_j(\sum_i|\mathcal{S}_{i,j}|)^3/\epsilon^2\Big) \tag{50}$$

$$=\tilde{\mathcal{O}}\Big(2k\sqrt{|\mathcal{A}||\mathcal{S}|}X^{\frac{3}{2}}/\epsilon + k|\mathcal{A}|X^3/\epsilon^2\Big). \tag{51}$$

Inequality (50) holds due to Arithmetic Mean–Geometric Mean Inequality, whose equality holds only when

$$m(j) = \sqrt{|A_j||\mathcal{S}|(\sum_i|\mathcal{S}_{i,j}|)}/\epsilon. \tag{52}$$

Equation (51) exists since $\sum_i|\mathcal{S}_{i,j}| \leq \sum_i|S_{i,k}| = X$ denote the size of the final restricted game; $X \gg |\mathcal{A}|$ dominates the complexity. ■

Theorem 16. *The Last-window average strategy (LAS) of SRMDO has the following sample complexity with probability at least $1-pk$ to reach ϵ-NE when employing oracle Best Response:*

$$\tilde{\mathcal{O}}\Big(kHP^2|\mathcal{A}|X^2/\epsilon^2 + \sum_{j=1}^{k}H_j m(j) + |\mathcal{A}||\mathcal{S}|P^2X^2/\epsilon^2 m(j)\Big), \tag{53}$$

and the following sample complexity when employing δ-BR:

$$\tilde{\mathcal{O}}(kHP^2|\mathcal{A}|X^2/(\epsilon-\delta)^2 + \sum_{j=1}^{k}H_j m(j) + H|\mathcal{A}|P^2X^2/(\epsilon-\delta)^2 m(j)), \tag{54}$$

if $\delta<\epsilon$, where $P = 1+\sqrt{\frac{2}{p}}$, $H_j = \max_i H_{i,j}$ is the depth of the restricted game tree of $\mathbf{G}_j$, $H = \max_j H_j$, and the runtime complexity of δ-BR is $\tilde{\mathcal{O}}(|H|)$.

Proof We first compute the number of iterations of the last-window average strategy (LAS) of SRMDO to reach ϵ-NE. Before reaching the ϵ-NE of the original game, in each windows $T_j,\ j<k-1$, with probability $1-p$,

$$|T_j| < \tilde{\mathcal{O}}(P^2A_j(\sum_i|\mathcal{S}_{i,j}|)^2/\epsilon^2 + m(j) - 1), \tag{55}$$

where $P = (1 + \sqrt{\frac{2}{p}})$. Equation (55) holds based on the MCCFR regret upper bound in Equation (9), and the regret to complexity transformation in Lemma 2. The additional $m(j) - 1$ iterations appears in Equation (55) because the worst-case iterations spent in each time window is $m(j)$ more iterations than the iterations to reach local ϵ-NE. This is due to the periodicity of computing best response is $m(j)$.

However, when computing the sample complexity, unlike RMDO, the complexity of each iteration reduce from $\sum_i |\mathcal{S}_{i,j}|$ to H_j, where H_j indicates the depth of the EFG tree. Thus due to the number of times computing BR in each window T_j is $\tilde{\mathcal{O}}(P^2 A_j (\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 m(j) + 1 - 1/m(j))$, and the complexity of each round of BR computing is $\tilde{\mathcal{O}}(|\mathcal{S}|)$, then the sample complexity of SRMDO becomes:

$$
\begin{aligned}
&\sum_{j=1}^{k} \tilde{\mathcal{O}}(P^2 A_j (\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 m(j) + 1 - 1/m(j)) \cdot \tilde{\mathcal{O}}(|\mathcal{S}|) \\
&+ \sum_{j=1}^{k} \tilde{\mathcal{O}}(P^2 A_j (\sum_i |\mathcal{S}_{i,j}|)^2 H_j/\epsilon^2 + H_j m(j) - H_j), \\
\leq &\sum_{j=1}^{k} \tilde{\mathcal{O}}\Big(P^2 A_j |\mathcal{S}| (\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 m(j) + P^2 |\mathcal{A}_j| (\sum_i |\mathcal{S}_{i,j}|)^2 H_j/\epsilon^2 + H_j m(j)\Big) \qquad (56) \\
\leq &\tilde{\mathcal{O}}(kHP^2 |\mathcal{A}| X^2/\epsilon^2 + \sum_{j=1}^{k} H m(j) + |\mathcal{A}||\mathcal{S}| P^2 X^2/\epsilon^2 m(j)), \qquad (57)
\end{aligned}
$$

As for the δ-BR approximate Best Responder, it's similar to the proof above, except in each window, $|T_j| < \tilde{\mathcal{O}}(A_j (\sum_i |\mathcal{S}_{i,j}|)^2/(\epsilon - \delta)^2 + m(j) - 1)$ to ensure the following: we still can derive contraction via the fact that δ-BR of the average strategy (ϵ-NE) in current window still lies in current restricted game leading to ϵ-NE in the original game. Additionally, approximate BR only have complexity $\tilde{\mathcal{O}}(H)$ in each time of computation. Thus it's easy to derive the complexity:

$$
\tilde{\mathcal{O}}(kH|\mathcal{A}|X^2/(\epsilon - \delta)^2 + \sum_{j=1}^{k} H_j m(j) + H|\mathcal{A}|X^2/(\epsilon - \delta)^2 m(j)) \qquad (58)
$$

Let E_i denote the success event in the i-th window and $\Pr(E_i) \geq 1 - p$, then we only know that $\Pr(E_i^c) \leq p$. Therefore, by De Morgan's law and the union bound,

$$
\Pr\left(\bigcap_{i=1}^{k} E_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{k} E_i^c\right) \geq 1 - \sum_{i=1}^{k} \Pr(E_i^c) \geq 1 - kp. \qquad (59)
$$

Thus, the probability of the complexity guarantee to reach ϵ-NE is at least $1 - kp$. ■

Theorem 17. *By employing oracle Best Response and the following frequency function*

$$
m(j) = \frac{\sqrt{|\mathcal{S}|}P}{\epsilon} \cdot \sqrt{\frac{|\mathcal{A}_j|(\sum_i |\mathcal{S}_{i,j}|)^2}{H_j}}. \qquad (60)
$$

and the sample complexity of Last-window average strategy in SRMDO is the following with probability at least $1 - pk$*:*

$$\tilde{\mathcal{O}}(2\sqrt{|\mathcal{A}||\mathcal{S}|}kPX/\epsilon + k|\mathcal{A}|HX^2/\epsilon^2). \tag{61}$$

Proof According to the Equation (56) in the proof of Theorem 16, we have the sample complexity of SRMDO is:

$$\sum_{j=1}^{k} \tilde{\mathcal{O}}\Big(A_j|\mathcal{S}|P^2(\sum_i |\mathcal{S}_{i,j}|)^2/\epsilon^2 m(j) + P^2|\mathcal{A}_j|(\sum_i |\mathcal{S}_{i,j}|)^2 H_j/\epsilon^2 + H_j m(j)\Big). \tag{62}$$

Similar to Theorem 15, Equation (62) has the following lower bound:

$$\sum_{j=1}^{k} \tilde{\mathcal{O}}(2\sqrt{A_j|\mathcal{S}|}P(\sum_i |\mathcal{S}_{i,j}|)/\sqrt{H_j}\epsilon + |\mathcal{A}_j|(\sum_i |\mathcal{S}_{i,j}|)^2 H_j/\epsilon^2) \tag{63}$$

when

$$m(j) = \frac{\sqrt{|\mathcal{S}|}P}{\epsilon} \cdot \sqrt{\frac{|\mathcal{A}_j|(\sum_i |\mathcal{S}_{i,j}|)^2}{H_j}}. \tag{64}$$

As usual, we treat the tight upper bound of Equation (63) as the sample complexity, which is

$$\tilde{\mathcal{O}}(2\sqrt{|\mathcal{A}||\mathcal{S}|}kPX/\epsilon + k|\mathcal{A}|HX^2/\epsilon^2). \tag{65}$$

As for the probability, similar to Eq. 59, the complexity guarantee is with probability at least $1 - pk$. ■

Appendix C. Additional Analysis

Adaptive Double Oracle. As discussed in Section 4.1, AdaDO exhibits an empirical limitation when using the adaptive frequency function. The frequency schedule in AdaDO is derived from a worst-case theoretical complexity bound; however, the empirical complexity can be substantially smaller in practice, causing the theoretically optimal schedule to perform suboptimally. To mitigate this issue, we adopt a discounted empirical frequency schedule together with an early-stopping criterion based on periodically evaluating the exploitability of the restricted game. Specifically, for each restricted game, we run regret minimization for the theoretically prescribed number of iterations multiplied by a discount factor, and stop early when the exploitability $e(\cdot)$ of the average policy $\tilde{\pi}^t$ decreases too slowly. We conduct the following checking every c iterations:

$$\frac{\left|e(\tilde{\pi}^{t-c}) - e(\tilde{\pi}^t)\right|}{e(\tilde{\pi}^{t-c})} < \delta,$$

where $\tilde{\pi}^{t-c}$ denotes the policy at the previous exploitability check, t is the current iteration satisfying $t \bmod c = 0$, and δ is the threshold for the relative decrease rate. We skip the early

stopping checking at the first exploitability computation. Thus, the early stop checking only happens when $t > c$. As for the frequency discount α, overly small value leads to small BR computation frequency, which might waste computations on invaluable BR actions. This has been confirmed by our empirical ablation study in Figure 7, and 8.

Warm Starting Algorithm. When conducting warm starting, we initialize both the regret and strategy by a fixed positive value is necessary to encourage exploration on newly added actions in the restricted game. Otherwise, the regret and strategy will be hard to iterate and stuck at the values at prior restricted games.

Warm Starting Empirical Performance. In SRMDO, warm-starting can sometimes have an adverse effect, as illustrated in Figure 6. That is, warm-starting does not necessarily improve the performance of Double Oracle (DO). Unlike warm-starting in CFR (Brown and Sandholm, 2016), where initialization is performed within a fixed game, warm-starting in DO reuses regret information learned from a previous restricted game to initialize optimization in a newly expanded one. However, once the action space is enlarged, actions that appeared favorable in the previous restricted game may become substantially less useful or even misleading, if newly added actions or pure strategies dominate them in the current restricted game. This also helps explain why obtaining regret guarantees for warm-starting in DO is challenging. This issue is further exacerbated in SRMDO. Because MCCFR visits only a subset of actions and information sets, warm-starting from incomplete regret and strategy tables can provide a harmful initialization for solving the current restricted game.

On Large Kuhn Poker, warm starting leads to a significant performance improvement. This suggests that the regret and strategy learned in earlier restricted games provide effective initialization for later ones. Put differently, the actions introduced in later iterations have limited impact on the equilibrium strategy. However, due to the aforementioned issue, such behavior is not guaranteed across all games.

Appendix D. Games Descriptions

D.1 Blotto

Blotto is a sequential perfect-information game and a revised sequential version of the discrete Colonel Blotto game (Tang et al., 2023). At the beginning of the game, each player is given a set of forces with distinct strengths, which are deployed sequentially onto the battlefield. Under the parameter setting of 20, each player controls 20 forces with strengths from 0 to 19. The game proceeds over 4 successive deployments, forming two complete combat rounds. In each deployment step, the players alternately choose one force to deploy. The outcome of each fight is determined by the difference in strength between the forces deployed by the two players. After each round, the deployed forces are removed, and the next round begins. The overall payoff is the sum of the outcomes from all rounds.

D.2 Large Kuhn Poker

Large Kuhn poker follows the same structure as standard Kuhn poker: two players are each dealt one card from a 3-card deck, and then **alternate betting actions for at most 3 rounds.** The key difference is that instead of a binary action space $\{\text{pass}, \text{bet1}\}$, players

can bet any integer amount within their remaining pot. Specifically, P0 acts first with up to 38 legal actions (`range(38)`), P1 responds with 2 actions (fold or call if P0 bet; and check or bet any integer amount $\in [1, 39]$). The game tree is therefore wider but not deeper, inflating the action space without adding genuine strategic complexity.

On the number of information sets. With initial pot size to be 40, players have up to 39 legal bet sizes. This creates approximately **39 times** information sets of standard Kuhn poker. But notably, the depth remains a fixed number 3. So only the width of the game tree is increased.

On strategic difference from Kuhn poker. The consideration of increasing initial pot of Kuhn Poker to a Large version is to create dense action space (the original space only contains 2 actions) meanwhile maintains the simplicity of the game.

D.3 Other Games

Leduc Poker 10 Card It is the same as vanilla Leduc Poker except the initial number of cards is 10.

Leduc Poker Dummy It is the same as vanilla Leduc Poker except the actions in each information set are duplicated once (McAleer et al., 2021).

Oshi zumo It is a board game (Buro, 2004), where two players have 4 coins in the beginning of the game. There is a token in the middle of a board with length $2K + 1$. K in our case is 6. Players make action of bidding with the coins they have (at least 1). Then the player bid more is able to push the token one step toward its opponent. The objective is to push the token off the opponent side of the board. Payoff is 1 for the winner and -1 for the loser.

References

Yu Bai, Chi Jin, Song Mei, and Tiancheng Yu. Near-optimal learning of extensive-form games with imperfect information. In *International Conference on Machine Learning*, pages 1337–1382. PMLR, 2022.

Avrim Blum and Yishay Monsour. Learning, regret minimization, and equilibria. 2007.

Branislav Bosansky, Christopher Kiekintveld, Viliam Lisy, and Michal Pechoucek. An exact double-oracle algorithm for zero-sum extensive-form games with imperfect information. *Journal of Artificial Intelligence Research*, 51:829–866, 2014.

Branislav Bošanskỳ, Viliam Lisỳ, Marc Lanctot, Jiří Čermák, and Mark HM Winands. Algorithms for computing strategies in two-player simultaneous move games. *Artificial Intelligence*, 237:1–40, 2016.

Noam Brown and Tuomas Sandholm. Strategy-based warm starting for regret minimization in games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359:eaao1733, 12 2017. doi: 10.1126/science.aao1733.

Noam Brown and Tuomas Sandholm. Solving imperfect-information games via discounted regret minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1829–1836, 07 2019a. doi: 10.1609/aaai.v33i01.33011829.

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

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *International conference on machine learning*, pages 793–802. PMLR, 2019a.

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *International conference on machine learning*, pages 793–802. PMLR, 2019b.

Noam Brown, Anton Bakhtin, Adam Lerer, and Qucheng Gong. Combining deep reinforcement learning and search for imperfect-information games. *Advances in Neural Information Processing Systems*, 33:17057–17069, 2020a.

Noam Brown, Anton Bakhtin, Adam Lerer, and Qucheng Gong. Combining deep reinforcement learning and search for imperfect-information games. *arXiv preprint arXiv:2007.13544*, 2020b.

Neil Burch, Marc Lanctot, Duane Szafron, and Richard Gibson. Efficient monte carlo counterfactual regret minimization in games with many player actions. *Advances in neural information processing systems*, 25, 2012.

Michael Buro. Solving the oshi-zumo game. *Advances in Computer Games: Many Games, Many Challenges*, pages 361–366, 2004.

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

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. URL `https://openreview.net/forum?id=rrMK6hYNSx`.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Stochastic regret minimization in extensive-form games. In *International Conference on Machine Learning*, pages 3018–3028. PMLR, 2020.

Sergiu Hart. Games in extensive and strategic forms. *Handbook of game theory with economic applications*, 1:19–40, 1992.

Johannes Heinrich, Marc Lanctot, and David Silver. Fictitious self-play in extensive-form games. In *International conference on machine learning*, pages 805–813. PMLR, 2015.

Daphne Koller and Nimrod Megiddo. Finding mixed strategies with small supports in extensive form games. *International Journal of Game Theory*, 25(1):73–92, 1996.

Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte carlo sampling for regret minimization in extensive games. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009*, pages 1078–1086, 01 2009.

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.

Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, et al. Openspiel: A framework for reinforcement learning in games. *arXiv preprint arXiv:1908.09453*, 2019.

Siqi Liu, Luke Marris, Marc Lanctot, Georgios Piliouras, Joel Z Leibo, and Nicolas Heess. Neural population learning beyond symmetric zero-sum games. *arXiv preprint arXiv:2401.05133*, 2024.

Luke Marris, Paul Muller, Marc Lanctot, Karl Tuyls, and Thore Graepel. Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers. In *International Conference on Machine Learning*, pages 7480–7491. PMLR, 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.

Stephen McAleer, John Lanier, Pierre Baldi, and Roy Fox. XDO: A double oracle algorithm for extensive-form games. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Stephen McAleer, Kevin Wang, Marc Lanctot, John Lanier, Pierre Baldi, and Roy Fox. Anytime optimal psro for two-player zero-sum games. *arXiv preprint arXiv:2201.07700*, 2022.

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.

Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

Song Qin, Lei Zhang, Shuhan Qi, Jiajia Zhang, Huale Li, Xuan Wang, and Jing Xiao. Efdo: Solving extensive-form games based on double oracle. In *2022 4th International Conference on Data Intelligence and Security (ICDIS)*, pages 382–387. IEEE, 2022.

Philip J Reny. Rationality in extensive-form games. *Journal of Economic perspectives*, 6 (4):103–118, 1992.

Klaus Ritzberger et al. *The theory of extensive form games*. Springer, 2016.

Martin Schmid, Neil Burch, Marc Lanctot, Matej Moravcik, Rudolf Kadlec, and Michael Bowling. Variance reduction in monte carlo counterfactual regret minimization (vr-mccfr) for extensive form games using baselines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2157–2164, 2019.

L Julian Schvartzman and Michael P Wellman. Exploring large strategy spaces in empirical game modeling. *Agent Mediated Electronic Commerce (AMEC 2009)*, page 139, 2009.

Eric Steinberger, Adam Lerer, and Noam Brown. Dream: Deep regret minimization with advantage baselines and model-free learning. *arXiv preprint arXiv:2006.10410*, 2020.

Oskari Tammelin. Solving large imperfect information games using cfr+. 07 2014.

Xiaohang Tang, Le Cong Dihn, Stephen Marcus Mcaleer, Yaodong Yang, et al. Regret-minimizing double oracle for extensive-form games. In *International Conference on Machine Learning*, pages 33599–33615. PMLR, 2023.

John Von Neumann and Oskar Morgenstern. Theory of games and economic behavior: 60th anniversary commemorative edition. In *Theory of games and economic behavior*. Princeton university press, 2007.

Yufei Wang, Zheyuan Ryan Shi, Lantao Yu, Yi Wu, Rohit Singh, Lucas Joppa, and Fei Fang. Deep reinforcement learning for green security games with real-time information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1401–1408, 2019.

Kevin Waugh. Abstraction in large extensive games. 2009.

Robert Wilson. Computing equilibria of two-person games from the extensive form. *Management Science*, 18(7):448–460, 1972.

Mason Wright. Using reinforcement learning to validate empirical game-theoretic analysis: A continuous double auction study. *arXiv preprint arXiv:1604.06710*, 2016.

Ming Zhou, Jingxiao Chen, Ying Wen, Weinan Zhang, Yaodong Yang, Yong Yu, and Jun Wang. Efficient policy space response oracles. *arXiv preprint arXiv:2202.00633*, 2022.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems.*, volume 2008, 01 2007.