Safe and Robust Subgame Exploitation in Imperfect Information Games

Zhenxing Ge [1 2] **Zheng Xu** [1] **Tianyu Ding** [3] **Linjian Meng** [1] **Bo An** [4 2] **Wenbin Li** [1] **Yang Gao** [1 5]

Abstract

Opponent exploitation is an important task for players to exploit the weaknesses of others in games. Existing approaches mainly focus on balancing between exploitation and exploitability but are often vulnerable to modeling errors and deceptive adversaries. To address this problem, our paper offers a novel perspective on the safety of opponent exploitation, named *Adaptation Safety*. This concept leverages the insight that strategies, even those not explicitly aimed at opponent exploitation, may inherently be exploitable due to computational complexities, rendering traditional safety overly rigorous. In contrast, adaptation safety requires that the strategy should not be more exploitable than it would be in scenarios where opponent exploitation is not considered. Building on such adaptation safety, we further propose an Opponent eXploitation Search (OX-Search) framework by incorporating real-time search techniques for efficient online opponent exploitation. Moreover, we provide theoretical analyses to show the adaptation safety and robust exploitation of OX-Search, even with inaccurate opponent models. Empirical evaluations in popular poker games demonstrate OX-Search's superiority in both exploitability and exploitation compared to previous methods.

1. Introduction

Recent advances in real-time strategy search techniques (Burch et al., 2014; Ganzfried & Sandholm, 2015a; Brown & Sandholm, 2017; Brown et al., 2018; Liu et al., 2023) have achieved notable accomplishments in games such as Texas Holdem Poker (Moravčík et al., 2017; Brown & Sandholm, 2018; 2019), Hanabi (Lerer et al., 2020), Mahjong (Li et al., 2020), and Dark Chess (Zhang & Sandholm, 2021). These successes highlight the effectiveness of real-time search in complex game scenarios. However, despite their theoretical robustness, these strategies often exhibit excessive caution against suboptimal opponents (Albrecht & Stone, 2018; Liu et al., 2022), thus missing opportunities for higher payoffs. To better exploit these weaknesses, there have been efforts to construct opponent models based on historical actions (Southey et al., 2005; Ganzfried & Sandholm, 2011; Bard et al., 2013; He & Boyd-Graber, 2016; Tian et al., 2019), with the aim of exploiting these opponents more effectively and maximizing profits (Albrecht & Stone, 2018; Zheng et al., 2018; Liu et al., 2022).

Although numerous methodologies exist to model opponents, opponent exploitation is highly challenging in practice (Ganzfried & Sandholm, 2011; 2015b). During gameplay, the predictions generated by these models may suffer from inaccuracy due to limited data availability for constructing the models or the variability of an opponent's strategy. Consequently, players need to ensure the robustness of their strategies to effectively exploit opponents (Bernasconi-de Luca et al., 2021), even in the presence of modeling errors. Additionally, exploiting an opponent inherently risks being exploited in return, especially when facing a deceptive adversary. This challenge is commonly referred to as the "being taught and exploited" problem (Sandholm, 2007). The aforementioned concerns necessitate the development of an opponent exploitation approach that addresses both **safety**—securing the profit against the deceptive opponents, and **robust exploitation**—upholding the effective exploitation in the presence of modeling inaccuracies.

However, most previous works predominantly excel in either ensuring strategic safety (Ganzfried & Sandholm, 2015b) or in achieving robust exploitation (Zheng et al., 2018; Bernasconi-de Luca et al., 2021; Fu et al., 2022), yet fail to effectively incorporate both of these two critical aspects. Few exceptions including (Liu et al., 2022) aim to search for a Pareto optimal balancing safety and robust exploitation, but their strategies may not always guarantee safety or increase profits against opponents.

[1]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu, China. [2]School of Computer Science and Engineering, Nanyang Technological University, Singapore. [3]Microsoft Corporation, Redmond, Washington, USA. [4]Skywork AI, Singapore. [5]School of Intelligence Science and Technology, Nanjing University, Suzhou Campus, Suzhou, Jiangsu, China.. Correspondence to: Yang Gao <gaoy@nju.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

To address this problem, this paper first redefines safety in the context of online exploitation for two-player zero-sum games, introducing the notion of *adaptation safety*. This concept diverges from the conventional safety definition, which is benchmarked against the Nash Equilibrium (NE) as seen in previous works (Ganzfried & Sandholm, 2015b; Liu et al., 2022). We notice that in large-scale games, where computing an exact NE is exceptionally challenging, the strategies used by players are inherently exploitable. Given the impracticality of achieving NE-level safety without opponent exploitation, the traditional safety becomes an overly strict benchmark and limits potential profit. In contrast, an exploitation strategy that adheres to adaptation safety is designed to be no more exploitable than a predefined initial strategy, or 'blueprint'. This definition allows players to adapt their strategies while avoiding the "being taught and exploited" problem.

Furthermore, to swiftly respond to potential changes in opponent models and effectively exploit their weaknesses, we propose an *Opponent-eXploitation Search (OX-Search)* framework, which integrates real-time search techniques for computationally efficient online adaptation. Unlike traditional opponent exploitation methods that often treat safety and exploitation as conflicting goals, balanced through an exploitation level hyperparameter (Johanson et al., 2007; Liu et al., 2022), OX-Search aims to exploit the opponent's weakness within the bounds of adaptation safety during real-time search. In essence, OX-Search strives to identify an exploitation strategy that is best against opponents, without exceeding the risk level of the initial blueprint. The adaptation safety search scheme plays a vital role in mitigating the impact of modeling errors by bounding the worst-case profit loss at each infoset.

The main contributions can be summarized as follows.

- **Introduction of the "Adaptation Safety" and the OX-Search framework**: We introduce the concept of adaptation safety and present the Opponent-eXploitation Search (OX-Search) framework, incorporating real-time search techniques for efficient online opponent exploitation. Adaptation safety offers a novel perspective on opponent exploitation, thus addressing the challenge of balancing exploitation with safety in dynamic environments.

- **Theoretical analysis of OX-Search**: We provide a comprehensive theoretical analysis of OX-Search, elucidating its properties and guarantees. Specifically, it establishes that OX-Search ensures safety by being no more exploitable than the blueprint strategy. Moreover, OX-Search guarantees an anticipated increase of the worst-case exploitation bounds despite the existence of opponent modeling errors.

- **Development of a specialized gadget game**: We develop a novel gadget game tailored to OX-Search, which facilitates the use of advanced equilibrium-finding techniques, thereby extending the framework's ability to handle large-scale games.

- **Empirical evaluation and superior performance**: We demonstrate the superior performance of OX-Search compared to other opponent exploitation methods through empirical evaluation on popular poker variants such as Leduc Hold'em and Flop Hold'em Poker. The results highlight the efficacy of OX-Search in achieving high levels of exploitation and safety.

2. Related Works

This paper focuses on the online opponent exploitation problem in two-player zero-sum games, which is closely related to the works on the real-time search in imperfect information games and opponent exploitation.

2.1. Real-time Search in Imperfect Information Games

Real-time search has been widely used in solving real-world imperfect information problems (Moravčík et al., 2017; Brown & Sandholm, 2018; 2019; Li et al., 2020; Zhang & Sandholm, 2021; Ge et al., 2023). Typical real-time search methods employ precomputed blueprints and improve them in various ways.

One kind of approach involves utilizing online Monte Carlo sampling techniques (Lisý et al., 2015; Šustr et al., 2019; Li et al., 2020), which leverage regret or action values computed during offline training and update them through customized sampling. While they fine-tune the blueprint and reduce exploitability, the conservative nature of the adjustments may limit their ability to effectively adapt to opponent exploitation, potentially forgoing higher payoffs.

Another prevalent approach is subgame solving (Burch et al., 2014; Ganzfried & Sandholm, 2015a; Brown & Sandholm, 2017), which serves as a fundamental technique for several benchmark AI systems (Moravčík et al., 2017; Brown & Sandholm, 2018; 2019). In subgame solving, the player constructs a subgame based on the current state and computes a strategy specifically tailored for that subgame. The player then plays according to this new strategy rather than strictly adhering to the blueprint. Unlike Monte Carlo online sampling methods, safe subgame solving approaches (Moravcik et al., 2016; Brown & Sandholm, 2017) do not restrict the new strategy to be minor variations on the blueprint. Instead, their objective is to ensure that the exploitability is reduced compared to the blueprint. This is accomplished by constructing augmented subgames that offer the opponent the option to abstain from entering the subgame, thereby compelling the player to develop a strategy that matches or

exceeds the blueprint. Additionally, efforts have been made to alleviate the computational demands of subgame solving in large games (Zhang & Sandholm, 2021; Liu et al., 2023) by focusing on shrinking the subgame tree, thus facilitating the application of subgame solving methods to large game scenarios.

2.2. Opponent Exploitation

Within the realm of opponent exploitation, previous works (Gilpin & Sandholm, 2006; Li & Miikkulainen, 2018; Wu et al., 2021; Ge et al., 2022; Yu et al., 2022) have primarily centered around constructing opponent models based on historical data and developing best response strategies. However, these studies have not thoroughly addressed two critical aspects: safety, which involves ensuring profitability against deceptive opponents, and robust exploitation, which requires maintaining effective exploitation even in the face of inaccuracies in opponent modeling.

Certain methodologies, such as those proposed in (He & Boyd-Graber, 2016; Zheng et al., 2018), constrain the scope of potential exploitation strategies when applying opponent exploitation. For example, DRON (He & Boyd-Graber, 2016) leverages expert networks to exploit opponents by selecting strategies from a predefined set. Deep BPR+ (Zheng et al., 2018) utilizes Bayesian optimization to reuse a library of strategies for exploitation. These methods lack safety guarantees and can be highly ineffective if used with an improper strategy set.

In the context of repeated games, the work of (Ganzfried & Sandholm, 2015b) explores safe exploitation methods that risk only those utilities that have been won over NE in expectation. They establish the existence of a non-equilibrium safe exploitation method. Although the method is theoretically sound, calculating the utility expected from playing an NE is computationally intensive, and finding the exact NE strategy might not be feasible in large games.

Another approach (Bernasconi-de Luca et al., 2021) considers modeling errors in opponent exploitation. They construct a trust region around the opponent model based on historical actions, delineating a feasible strategy space in which utility is bounded against the opponent's strategy within this region. A strategy is then selected through a process akin to the Upper Confidence Bound (UCB) approach (Abbasi-Yadkori et al., 2011). GSCU (Fu et al., 2022) employs a similar decision-making procedure, wherein greedy exploitation is pursued when certainty is high, and resorts to a conservative strategy amid uncertainty. While robust against modeling errors, these approaches may still be vulnerable to exploitation by deceptive opponents.

An alternative approach, known as the p-Restricted Nash Response (RNR) (Johanson et al., 2007), seeks to optimize an objective that integrates safety and exploitation considerations. RNR assumes the opponent follows the modeled strategy with probability p and can select any strategy with probability $1 - p$. This can be treated as an equilibrium-finding problem. Although RNR has been shown to compute a solution that is Pareto-optimal between safety and exploitation, it fails to furnish guarantees regarding safety and robust exploitation.

More recently, the Safe Exploitation Search (SES) method (Liu et al., 2022) has been proposed, building upon the idea of RNR (Johanson et al., 2007). SES incorporates real-time solving techniques into the opponent exploitation procedure and strikes a balance between safety and exploitation during subgame solving. However, it is noteworthy that the strategy resulting from this combination may not necessarily be safer than the blueprint strategy, nor surpass the efficacy of previous strategies against the opponent, as inferred from their theoretical analysis.

These existing methods have primarily concentrated on either safety or robust exploitation, yet often fall short in simultaneously addressing both. However, this dual focus is particularly vital in practical applications, as neglecting either aspect can render the exploitation strategy vulnerable, potentially leading to suboptimal performance. Our work aims to address this problem by developing a novel approach that integrates real-time search techniques with a new perspective of adaptation safety in two-player zero-sum games. While our focus is formulating a safe exploitation algorithm, it is noteworthy that it can be complemented with existing agent modeling techniques (Albrecht & Stone, 2018) for estimating opponent's strategies.

3. Notations and Background

3.1. Extensive-form Games

Extensive-form games (EFGs) with imperfect information serve as a highly effective framework for modeling sequential decision-making problems. Such games are represented by $G = \langle N, H, P, \{u_i\}, \mathcal{I} \rangle$. In an extensive-form game, the player set $N = \{1, 2, \dots\} \cup \{c\}$ includes all the players participating in the game along with a unique player known as the chance player, denoted by c, who acts according to a predetermined probability distribution. The history set H represents the sequence of actions that have taken place in the game. When an action a leads from history h to history h', it is denoted as $h \cdot a = h'$. Additionally, the notation $h \sqsubset h'$ signifies that there is a sequence of actions leading from history h to history h'. For each $h \in H$, the acting player can choose an action from the action set $\{a \mid h \cdot a \in H\}$, denoted by $A(h)$. The set of terminal histories are defined as Z, and it comprises nodes where the action set is empty. These terminal histories correspond to the leaf nodes of the

game tree. For any history $h \in H \backslash Z$, a player function assigns the acting player at h, *i.e.,* $P(h) = i \in N$. Upon reaching each of the leaf nodes $z \in Z$, the utility function $u_i(z)$ specifies the payoff received by player i. Imperfect information is captured by the notion of information sets, or infosets. The set $\mathcal{I}$ represents the partition of the history set H into these infosets. Each infoset $I_i \in \mathcal{I}_i$ corresponds to the information that is available to player i. Within player i's infoset, the player cannot distinguish between two histories h and h', implying that the player function and the action set are identical for all histories in I_i. Hence, they can be denoted as $P(I_i)$ and $A(I_i)$, respectively.

An imperfect information subgame is a collection of histories structured as a forest of trees and is closed under the descendant relation and infosets for any player. For any history h in subgame S, if $h' \sqsupset h$, then $h' \in S$. Additionally, for any player i, if $h, h' \in I_i$, then $h' \in S$. We further denote the earliest reach history set as S_{top}. That is, $S_{top} = \{h \in S \mid \forall h' \sqsubset h, h' \notin S\}$.

3.2. Strategies and Counterfactual Values in EFGs

A strategy $\sigma_i(I_i)$ for player i represents a probability distribution over valid actions $A(I_i)$. The probability of selecting a specific action a is denoted by $\sigma_i(I_i, a)$. The joint probability of reaching a history h when all players adhere to their strategies σ is given by $\pi^\sigma(h)$. The contribution of player i to the reach probability, assuming other players and the chance player picks actions leading to h, is $\pi_i^\sigma(h) = \Pi_{h' \cdot a \sqsubset h, P(h')=i} \sigma_i(h', a)$. Let $-i$ be all the players excluding player i, and σ_{-i} be their strategies, the contribution of the chance and all players other than i is denoted by $\pi_{-i}^\sigma(h)$.

The expected utility for player i, given the strategy profile of all players $\sigma = \langle \sigma_i, \sigma_{-i} \rangle$, is $\sum_{z \in Z} (\pi^\sigma(z) u_i(z))$, and we denote it by $u_i(\sigma_i, \sigma_{-i})$ for simplicity. A best response strategy $BR(\sigma_{-i})$ is a strategy that maximizes player i's payoff against σ_{-i}. Formally, $BR(\sigma_{-i}) = \arg\max_{\sigma_i'} u_i(\sigma_i', \sigma_{-i})$. A Nash Equilibrium is a self-enforcing strategy profile $\sigma^* = (\sigma_i^*, \sigma_{-i}^*)$ that no one has an incentive to unilaterally change its strategy, which means $\forall i \in N, u_i(\sigma_i^*, \sigma_{-i}^*) \geq \max_{\sigma_i'} u_i(\sigma_i', \sigma_{-i}^*)$.

The exploitability of a strategy σ_i in a two-player zero-sum game is represented by $\exp(\sigma_i)$, which quantifies how much player i would lose if the opponent plays according to $BR(\sigma_i)$ compared to the Nash Equilibrium strategy σ_i^*. Formally, $\exp(\sigma_i) = u_i(\sigma_i^*, \sigma_{-i}^*) - u_i(\sigma_i, BR(\sigma_i))$.

Let $\pi^\sigma(h, h')$ be the probability of reaching h' given that h is reached if $h \sqsubseteq h'$, and $\pi^\sigma(h, h') = 0$ otherwise. The expected value at history h is $v_i^\sigma(h) = \sum_{z \in Z} \pi^\sigma(h, z) u_i(z)$ and $v_i^\sigma(h, a)$ is the expected value assumed action a is chosen. The counterfactual value of infoset I_i is the sum of the expected value of histories $h \in I_i$ weighted with the other players' reach probability, given I_i is reached. Formally, $v_i^\sigma(I_i) = \frac{\sum_{h \in I_i} (\pi_{-i}^\sigma(h) v_i^\sigma(h))}{\sum_{h \in I_i} \pi_{-i}^\sigma(h)}$, and the counterfactual value of action a is $v_i^\sigma(I_i, a) = \frac{\sum_{h \in I_i} (\pi_{-i}^\sigma(h) v_i^\sigma(h, a))}{\sum_{h \in I_i} \pi_{-i}^\sigma(h)}$. The player i's counterfactual best response $CBR(\sigma_{-i})$ is a best response strategy that also maximizes the expected value at infoset I_i even with $\pi_i^{\sigma_i}(I_i) = 0$. That is, $CBR(\sigma_{-i})(I_i, a) \geq 0$ only if $v_i^\sigma(I_i, a) \geq \max_{a' \in A(I_i)} v_i^\sigma(I_i, a')$. The counterfactual best response value $CBV_i^{\sigma_{-i}}(I)$ is further defined as the counterfactual value player i will receive when playing according to $CBR(\sigma_{-i})$ against σ_{-i} at I. Formally, $CBV_i^{\sigma_{-i}}(I) = v_i^{\langle CBR(\sigma_{-i}), \sigma_{-i} \rangle}(I)$, and $CBV_i^{\sigma_{-i}}(I, a) = v_i^{\langle CBR(\sigma_{-i}), \sigma_{-i} \rangle}(I, a)$.

4. Exploitation in Subgame Refinement

4.1. Definition of Adaptation Safety

Previous works (Ganzfried & Sandholm, 2015b; Liu et al., 2022) have primarily focused on safety guarantees that ensure the expected payoff of an exploitation strategy is not lower than that of a Nash Equilibrium against a best response player. While these safety guarantees guard against worst-case scenarios and prevent further exploitation in toy games, they are not suitable for real-world games where computing the exact equilibrium is a daunting task. In such games, strategies that players can employ are inherently exploitable, rendering the traditional safety concept, which is based on Nash Equilibrium, inadequate and excessively rigorous.

To address this limitation, we introduce the notion termed *adaptation safety* for opponent exploitation methods. Adaptation safety captures the idea that the strategy a player employs could be exploitable and might result in a decrease in profits compared to NEs. In light of this, it is natural to ask why a player would not seek to exploit the opponent while accepting the risk of a potential loss in profit, as long as the loss is not greater than what would be incurred using the current strategy. By constraining the risk in such a way, the exploitation strategy is rendered safe as it cannot be exploited beyond this point. The formal definition of adaptation safety is provided in Definition 4.1.

Definition 4.1. (Adaptation Safety) An opponent exploitation method is adaptation safe if, for any blueprint strategy σ, it yields an exploitation strategy σ' such that $\exp(\sigma') \leq \exp(\sigma)$.

The concept of adaptation safety can be viewed as an extension of traditional safety, as traditional safety is a special case of adaptation safety where the blueprint is a NE. The new safety perspective offers a pragmatic and realistic ap-

proach to safety in opponent exploitation. Recognizing the inherently exploitable nature of the strategy that the player employs, it ensures that the exploitation strategy is no less profitable than the blueprint strategy.

In essence, adaptation safety eases the constraints on the range of possible exploitation strategies, allowing for a more flexible exploration of strategies. Below is an illustrative example that demonstrates a strategy capable of exploiting the opponent beyond what the blueprint strategy would allow while adhering to the principle of adaptation safety.

Example 4.2. Consider the traditional two-player zero-sum game, Rock-Paper-Scissors, where each player chooses rock, paper, or scissors simultaneously. The rules are simple: rock defeats scissors, scissors defeats paper, and paper defeats rock. The winner receives a payoff of +1, while the loser gets -1. If both players choose the same option, it is a tie, and they both receive a payoff of 0.

Now assuming the blueprint is $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$ for rock, paper and scissors respectively. If the opponent always chooses scissors, the expected value of the blueprint strategy would be $-\frac{1}{4}$. However, if the player were to recognize the opponent's pattern and apply a safe exploitation method, the player could turn to an exploitation strategy $(\frac{1}{2}, \frac{1}{4}, \frac{1}{4})$ that has the same exploitability as the blueprint. With this adapted strategy, the player now has a higher expected value of $\frac{1}{4}$ against this opponent, thus improving by $\frac{1}{2}$.

4.2. Opponent-Exploitation Search

We now introduce a new exploitation framework called *Opponent-eXploitation Search (OX-Search)*. Let σ be the precomputed blueprint. Without loss of generality, we assume the existence of an opponent model $\hat{\sigma}_1$, and through the implementation of OX-Search on a specific subgame $S \in \mathbb{S}$, we refine player 2's strategy σ_2^S. Consequently, Player 2 opts to adhere to σ_2^S, as opposed to σ, within S.

Upon reaching an infoset I_2 in S_{top} of S, let $\hat{p}(I_1) = \sum_{h \in I_1} \pi_1^{\hat{\sigma}_1}(h)$ be the estimated reach probability of the opponent, OX-Search seeks to find a subgame strategy σ_2^S that maximizes the following objective:

$$\sum_{I_1^i \in S_{top}} \hat{p}(I_1^i) \left(CBV_1^{\sigma}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i) \right), \quad (1)$$

which is subject to the safety constraints:

$$CBV_1^{\sigma}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i) \geq 0, \quad (2)$$

for all $I_1^i \in Stop$. Here, the term $CBV_1^{\sigma}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i)$ represents the improvement in strategy σ_2^S at infoset I_1^i. The objective in Equation (1) maximizes the improvement over the estimated distribution $\hat{p}$ while satisfying the safety constraints (2) at the same time.

The safety constraints (2) are commonly used in safe subgame solving methods, and they ensure that the refined strategy σ_2^S does not decrease the expected payoff against optimal opponent at each opponent's infoset. Additionally, the constraints prevent the opponent from gaining an advantage through altering its reach probability over I_1^i. Therefore, strategy σ_2^S is safer than player 2's strategy σ_2 in strategy profile σ both inside and outside the subgame S, leading to adaptation safety in the OX-Search context.

We provide the formal description in Theorem 4.3.

Theorem 4.3. *Let $\mathbb{S}$ be a set of disjoint subgames S. Assume that OX-Search is performed on each subgame within $\mathbb{S}$. The resulting refined strategy is denoted as σ_2'. The exploitability of the refined strategy adheres to the following inequality:*

$$\exp(\sigma_2') \leq \exp(\sigma_2)\,. \quad (3)$$

Theorem 4.3 guarantees that the exploitability of the refined strategy σ_2' is no greater than that of the original strategy σ_2. This implies that OX-Search ensures the refined strategy achieves exploitability equal to or better than the original strategy. Furthermore, by enforcing the safety constraints (2), OX-Search guarantees a increment of the profit lower bound against the opponent, even in the presence of errors or inaccuracies in the predictions or opponent modeling. This highlights the robust exploitation of OX-Search in handling uncertainties and deviations from the opponent model, as further illustrated in Theorem 4.4.

Theorem 4.4. *Let $\epsilon = \max_{I_1^i \in S_{top}, \hat{p}(I_1^i) \neq 0} \frac{\hat{p}(I_1^i) - p(I_1^i)}{\hat{p}(I_1^i)} \leq 1$ be the metric quantifying the estimation error between the true distributions $p(I_1^i)$ and the estimated distributions $\hat{p}(I_1^i)$. We use $BR_1^{[S,\sigma_1]}(\sigma)$ to denote the strategy for player 1, which maximizes its utility in subgame S against σ_{-1} under the constraint that $BR_1^{[S,\sigma_1]}(\sigma)$ and σ_1 differs only inside S. Let $\delta = \max_{\sigma_2'} \min_{I_1^i \in S_{top}} \left(CBV_1^{\sigma_2}(I_1^i) - CBV_1^{\sigma_2'}(I_1^i) \right) \geq 0$. The expected payoff of OX-Search $u_2^{\langle BR_1^{[S,\sigma_1]}(\sigma_2'), \sigma_2' \rangle}(S)$ at subgame S is lower bounded by*

$$u_2^{\langle BR_1^{[S,\sigma_1]}(\sigma_2'), \sigma_2' \rangle}(S) \geq u_2^{\langle BR_1^{[S,\sigma_1]}(\sigma_2), \sigma_2 \rangle}(S) + (1-\epsilon)\delta. \quad (4)$$

Since OX-Search only leverage the knowledge of the prediction about the opponent's strategy before the reaching infoset, Theorem 4.4 assume the the opponent plays optimally throughout the remaining portion of the game. As suggested in Theorem 4.4, applying OX-Search elevates the expected payoff by at least $(1-\epsilon)\delta$, when the opponent plays optimally thereafter. To further exploit the opponent and amplify profit, players can employ OX-Search repeatedly at each newly encountered information set in a nested fashion, which maintains adaptation safety at the same time.

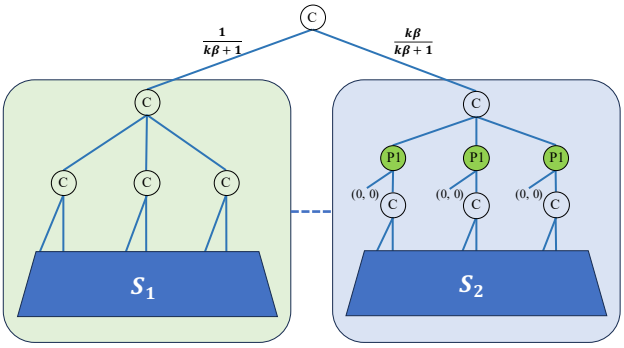


Figure 1. The constructed gadget game of OX-Search. C represents the chance node and P1 represents player 1's action node.

The iterative use of OX-Search enables the player to continuously refine its strategy and exploit opportunities for higher payoff, while simultaneously ensuring a level of robustness against the opponent's actions and modeling uncertainties.

It is important to underscore the significant differences between OX-Search and the SES (Liu et al., 2022) method, which uses a convex combination of exploitation and safety. 1) OX-Search provides theoretical guarantees regarding both safety and potential profit. In contrast, SES can only deliver an approximate safety guarantee under stringent assumptions, failing to ensure robust exploitation amidst estimation errors. 2) SES requires a hyperparameter to control the level of exploitation. Identifying an optimal hyperparameter can be daunting, thereby potentially restricting the practical utility of SES. 3) OX-Search can be easily extended for nested applications without sacrificing safety. Contrarily, applying SES in a nested manner could lead to severe problems due to potential loss stemming from a lack of safety and robustness against estimation errors.

4.3. Gadget Game

While OX-Search can be solved using Linear Programming (LP), it has been shown that LP may not effectively utilize the structural properties of extensive-form games, making it challenging to apply to large games (Davis et al., 2019; Liu et al., 2022). A common method to overcome this issue involves the use of the Constrained Counterfactual Regret Minimization (CFR) method (Davis et al., 2019). However, the direct application of Constrained CFR to OX-Search necessitates the computation of the exact gradient value $\nabla_{\sigma_2^t} CBV^{\sigma_2^t} 1(I_1^i)$ for each $I_1^i \in S_{top}$ at each iteration t, which is a non-trivial task. In order to expedite the strategy-solving process and make it compatible with advanced equilibrium-finding algorithms for extensive-form games, we need to construct a gadget game, in which the NE is the solution to objective (1) and constraint (2).

We begin by transforming the problem formulation. Given the feasibility of constraint (2), the problem is equivalent to

$$\min_{\sigma_2^S} \max_{\sigma_1, \boldsymbol{\lambda} \geq 0} \sum_{I_1^i \in S_{top}} \hat{p}(I_1^i) \left(v_1^{\langle \sigma_1, \sigma_2^S \rangle}(I_1^i) - CBV_1^{\sigma}(I_1^i) \right) + \sum_{I_1^i \in S_{top}} \lambda_i \left(v_1^{\langle \sigma_1, \sigma_2^S \rangle}(I_1^i) - CBV_1^{\sigma}(I_1^i) \right), \tag{5}$$

where $\boldsymbol{\lambda} = \langle \lambda_1, \lambda_2, \dots \rangle$ is the Lagrange multiplier.

To facilitate the construction of the gadget game, similar to (Davis et al., 2019), we operate under the assumption that $\boldsymbol{\lambda}$ is upper-bounded, *i.e.*, $\forall i, \lambda_i \leq \beta$. This allows us to rewrite Equation (5) as follows:

$$\min_{\sigma_2^S} \max_{\sigma_1, 0 \leq \boldsymbol{\lambda} \leq \beta} \sum_{I_1^i \in S_{top}} \hat{p}(I_1^i) \left(v_1^{\langle \sigma_1, \sigma_2^S \rangle}(I_1^i) - CBV_1^{\sigma}(I_1^i) \right) + \beta \sum_{I_1^i \in S_{top}} \left(\frac{\lambda_i}{\beta} \left(v_1^{\langle \sigma_1, \sigma_2^S \rangle}(I_1^i) - CBV_1^{\sigma}(I_1^i) \right) + \frac{\beta - \lambda_i}{\beta} 0 \right). \tag{6}$$

Considering that β is a fixed value, we can further simplify the problem by introducing the normalise operation. Suppose the number of I_1^i in S_{top} is k, solving Equation (6) is equivalent to solving:

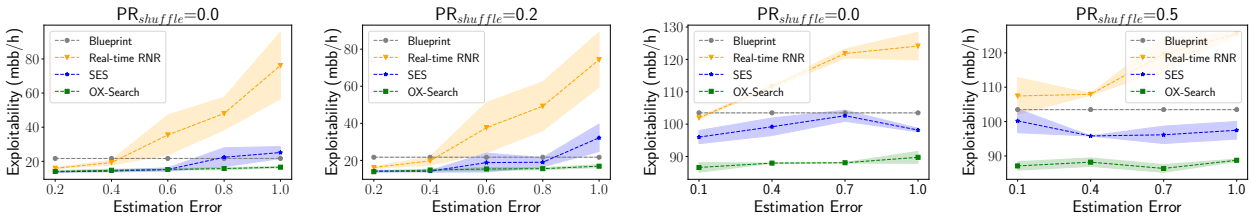
$$\min_{\sigma_2^S} \max_{\sigma_1, 0 \leq \boldsymbol{\lambda} \leq \beta} \frac{1}{k\beta + 1} \sum_{I_1^i \in S_{top}} \hat{p}(I_1^i) M_1^{\langle \sigma_1, \sigma_2^S \rangle}(I_1^i) + \frac{k\beta}{k\beta + 1} \cdot \frac{1}{k} \sum_{I_1^i \in S_{top}} \left(\frac{\lambda_i}{\beta} M_1^{\langle \sigma_1, \sigma_2^S \rangle}(I_1^i) + \frac{\beta - \lambda_i}{\beta} \cdot 0 \right). \tag{7}$$

where $M_1^{\langle \sigma_1, \sigma_2^S \rangle}(I_1^i) = v_1^{\langle \sigma_1, \sigma_2^S \rangle}(I_1^i) - CBV_1^{\sigma}(I_1^i)$.

As a result, we can treat $\boldsymbol{\lambda}$ as player 1's strategy for selecting each information set I_1^i, and the optimal $\boldsymbol{\lambda}$ can be identified during the equilibrium-finding process in Equation (7). This approach facilitates the construction of the gadget game and enables efficient strategy solving.

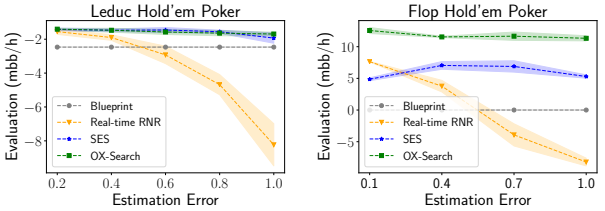
Constructing the gadget game, as shown in Figure 1, involves several steps:

1. The original subgame is duplicated into two identical parts, denoted as S_1 and S_2. Only player 1 can distinguish between these two parts. For each history in S_1 and S_2, the utility is adjusted by subtracting the corresponding counterfactual value. Specifically, for all $I_1^i \in S_{top}$, $h \in I_1^i$, and $h \sqsubset z$, the utility is updated as $u_1'(z) = u_1(z) - CBV_1^{\sigma}(I_1^i)$ and $u_2'(z) = -u_1'(z)$.
2. A chance node is added as the root of the gadget game. This chance node has two possible outcomes: selecting the left part with probability $\frac{1}{k\beta+1}$, or choosing the right part with probability $\frac{k\beta}{k\beta+1}$. The outcome of the chance node is visible to player 1.



(a) Leduc Hold'em. (b) Leduc Hold'em. (c) Flop Hold'em Poker. (d) Flop Hold'em Poker.

Figure 2. Exploitability of OX-Search and other methods in Leduc Hold'em and Flop Hold'em Poker. (**Lower is better**).



(a) Leduc Hold'em. (b) Flop Hold'em Poker.

Figure 3. Head-to-head evaluation of OX-Search and other exploitation methods against evolving opponents. (**Higher is better**).

3. In the left part of the game, the chance node selects each of player 1's information sets $I_1^i \in S_{top}$ with a probability proportional to $\hat{p}(I_1^i)$. The following chance node then selects the actual history with a probability proportional to $\pi^{\sigma}_{-1}(h)$. This part corresponds to the first term in Equation (7).

4. In the right part of the game, the first chance node randomly selects one information set $I_1^i \in S_{top}$ using a uniform distribution. The following player 1's node is an option node that allows player 1 to decide whether to enter the subgame or not. If player 1 opts not to enter, it receives a payoff of 0. If player 1 decides to enter, the game proceeds, and the subsequent chance node selects the history h with a probability proportional to $\pi^{\sigma}_{-1}(h)$. This part corresponds to the second term in Equation (7).

Since only player 1 can distinguish between S_1 and S_2, player 2's strategy remains the same in both S_1 and S_2. Consequently, player 1's counterfactual value and counterfactual regret in the corresponding information sets of S_1 and S_2 remain consistent. Therefore, the NE generated by CFR of the gadget game is exactly the solution of Equation (7). Moreover, this solution can be computed efficiently in real-time, as evidenced by prior research (Ganzfried & Sandholm, 2015a; Moravcik et al., 2016; Brown & Sandholm, 2017; Liu et al., 2022).

While the NE is the solution when the optimal $\boldsymbol{\lambda}^*$ lies in the interval $[0, \beta]^k$, this equivalence may not persist if $\boldsymbol{\lambda}^* \notin [0, \beta]^k$, and the resulting strategy might not be adaptation safe. Fortunately, we can identify this by examining player 1's NE at each option node and increasing the value of β if player 1 chooses to enter with a 100% probability (or very close, in the case of an approximate NE) at some option nodes, as outlined in Theorem 4.5.

Theorem 4.5. *Let σ^{S*} be the NE of the gadget game. If there is no optimal $\boldsymbol{\lambda}^* \in [0, \beta]^k$, then exists option node h_o such that $\sigma^{S*}(h_o, enter) = 1$.*

Furthermore, if a minimum acceptable safety violation exists, we can opt for a fixed value of β as per Theorem 4.6, eliminating the need for constantly increasing its value.

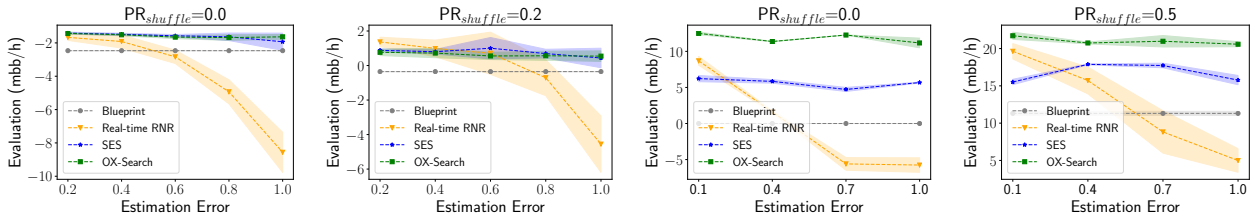
Theorem 4.6. *Let $\Delta = \max_{z \in Z} u_1(z) - \min_{z \in Z} u_1(z)$. σ_2^S is solved in Equation (7). The exploitability of refined strategy σ_2^S is bounded by:*

$$\exp(\sigma_2^S) - \exp(\sigma) \leq \frac{\Delta}{\beta}. \tag{8}$$

The above theorems provide crucial guidelines for determining the appropriate value of β, as well as ensuring the safety and exploitability bounds of the refined strategy in OX-Search.

5. Experiment

For a thorough assessment of OX-Search, our evaluation employs three key metrics: (I) safety in the face of the worst-case opponent; (II) effectiveness against evolving opponent strategies; and (III) robust exploitation in scenarios involving modeling errors. Extensive experiments are conducted on Leduc Hold'em (Southey et al., 2005; Wu et al., 2021) and Flop Hold'em Poker (FHP) (Brown et al., 2018; Liu et al., 2022) to evaluate the performance of OX-Search. Leduc Hold'em represents a small-scale poker game, while FHP represents a large-scale one. We compare OX-Search with Safe Exploitation Search (SES) (Liu et al., 2022) and a real-time search variant of RNR (Real-time RNR) (Johanson et al., 2007), as adapted in Liu et al. 2022. For SES and



(a) Leduc Hold'em. (b) Leduc Hold'em. (c) Flop Hold'em Poker. (d) Flop Hold'em Poker.

Figure 4. Head-to-head evaluation of OX-Search and other opponent exploitation methods against different opponents. (**Higher is better**).

Real-time RNR, we set the exploitation level hyperparameter to 0.3, as it has proven to yield the best performance with inaccurate opponent models in these two games (Liu et al., 2022).

5.1. Experiment Settings

For all experiments, the performance is evaluated against several distinct opponents, where the action probability is randomly shifted with probability $\Pr_{\text{shuffle}}$ from an approximate NE strategy. The constructed opponent is close to NE and is competitively strong (Liu et al., 2022). The estimation error of opponent modeling is also implemented by introducing noise into the reach probabilities (Liu et al., 2022), which is measured by the ℓ_1 distance between $\hat{p}$ and p, namely $||p - \hat{p}||_1$. Please refer to Appendix B for details.

5.2. Safety Against Worst-case Opponents

We first evaluate the exploitability of the OX-Search framework. Exploitability measures the maximum potential gain an opponent can achieve by misleading the player to create an inaccurate model of the opponent's strategy.

The results in Figure 2 reveal that OX-Search consistently demonstrates lower exploitability compared to the blueprint strategy in both Leduc Hold'em and FHP. This suggests that the strategies derived from OX-Search are less vulnerable to worst-case opponents, thereby enhancing the safety of our opponent exploitation approach. Notably, the exploitability of OX-Search also appears to be insensitive to increases in modeling error. While the SES was able to reduce exploitability in FHP, it was less effective in Leduc Hold'em. This disparity in performance can be attributed to our use of a finer-grained abstraction in FHP for subgame solving, whereas in Leduc Hold'em, we implemented the strategy without employing any form of abstraction. Furthermore, RNR shows an increase in exploitability concurrent with rising modeling errors, likely due to its assumptions about the opponent's behavior in the remainder of the subgame and its vulnerability to modeling inaccuracies.

5.3. Effectiveness against Evolving Opponent Strategies

In addition to worst-case scenarios, we also evaluate OX-Search in situations where opponents might change their strategies, potentially making the constructed opponent model less reliable. For this test, we set the opponent model to be the strategy shifted by $\Pr_{\text{shuffle}} = 0.2$ in Leduc Hold'em and 0.5 in FHP. The expected value is then computed against opponents with $\Pr_{\text{shuffle}} = 0.0$. As demonstrated in Figure 3, OX-Search consistently outperforms other methods in adapting to these strategic evolutions in both games.

5.4. Robustness in the Presence of Modeling Errors

Even with a static opponent strategy, the constructed model may have inaccuracies due to the complexities of opponent modeling (Albrecht & Stone, 2018). Therefore, it is crucial to assess the resilience of opponent-exploitation methodologies against such modeling errors.

As shown in Figure 4, OX-Search maintains comparable performance to SES in Leduc Hold'em and outperforms both SES and RNR in FHP. While RNR demonstrates better exploitation in Leduc Hold'em under conditions of minor modeling errors, its performance significantly deteriorates with increasing errors. Combined with the findings in Figure 2, these results suggest that OX-Search is not only as or more efficient in exploitation as other methods but also maintains its non-exploitability in worst-case scenarios.

6. Conclusion

In this paper, we have introduced the new notion of adaptation safety for opponent exploitation and developed the OX-Search framework, which exploits components based on this principle. Our theoretical analysis confirms that OX-Search is adaptation-safe and capable of exploiting opponents, even when faced with inaccurate opponent models.

This work opens two promising avenues for future research: (I) Exploring the extension of the OX-Search framework with advanced techniques, like estimating counterfactual best values (Brown & Sandholm, 2017), could further en-

hance its applicability. It is important to investigate whether such extensions can still uphold the principles of adaptation safety, and to understand the specific safety guarantees they provide; (II) While our current framework focuses on utilizing opponent reach probabilities from opponent models, it does not consider predictions of the opponent's future actions. Future research could aim to develop a more comprehensive search framework that incorporates these predictions while preserving adaptation safety.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (62192783, 62106100), the Natural Science Foundation of Jiangsu (BK20221441), Young Elite Scientists Sponsorship Program by CAST (2023QNRC001), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

Impact Statement

The techniques we have developed are very general and fundamental. Tools like those presented in this paper have the potential to level the playing field, enabling players with less education and experience to achieve the same proficiency as experts, thereby promoting a more equitable distribution of value.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. *Advances in Neural Information Processing Systems*, 24, 2011.

Albrecht, S. V. and Stone, P. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018.

Bard, N., Johanson, M., Burch, N., and Bowling, M. Online implicit agent modelling. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 255–262, 2013.

Bernasconi-de Luca, M., Cacciamani, F., Fioravanti, S., Gatti, N., Marchesi, A., and Trovò, F. Exploiting opponents under utility constraints in sequential games. *Advances in Neural Information Processing Systems*, 34: 13177–13188, 2021.

Brown, N. and Sandholm, T. Safe and nested subgame solving for imperfect-information games. *Advances in Neural Information Processing Systems*, 30:689–699, 2017.

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

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

Brown, N., Sandholm, T., and Amos, B. Depth-limited solving for imperfect-information games. *Advances in Neural Information Processing Systems*, 31, 2018.

Burch, N., Johanson, M., and Bowling, M. Solving imperfect information games using decomposition. In *AAAI conference on Artificial Intelligence*, 2014.

Davis, T., Waugh, K., and Bowling, M. Solving large extensive-form games with strategy constraints. In *AAAI Conference on Artificial Intelligence*, volume 33, pp. 1861–1868, 2019.

Fu, H., Tian, Y., Yu, H., Liu, W., Wu, S., Xiong, J., Wen, Y., Li, K., Xing, J., Fu, Q., et al. Greedy when sure and conservative when uncertain about the opponents. In *International Conference on Machine Learning*, pp. 6829–6848. PMLR, 2022.

Ganzfried, S. and Sandholm, T. Game theory-based opponent modeling in large imperfect-information games. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 533–540, 2011.

Ganzfried, S. and Sandholm, T. Endgame solving in large imperfect-information games. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 37–45, 2015a.

Ganzfried, S. and Sandholm, T. Safe opponent exploitation. *ACM Transactions on Economics and Computation (TEAC)*, 3(2):1–28, 2015b.

Ge, Z., Yang, S., Tian, P., Chen, Z., and Gao, Y. Modeling rationality: Toward better performance against unknown agents in sequential games. *IEEE Transactions on Cybernetics*, 2022.

Ge, Z., Xu, Z., Ding, T., Li, W., and Gao, Y. Efficient subgame refinement for extensive-form games. In *Neural Information Processing Systems*, 2023.

Gilpin, A. and Sandholm, T. A texas hold'em poker player based on automated abstraction and real-time equilibrium computation. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 1453–1454, 2006.

He, H. and Boyd-Graber, J. L. Opponent modeling in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 1804–1813, 2016.

Johanson, M., Zinkevich, M., and Bowling, M. Computing robust counter-strategies. *Advances in neural information processing systems*, 20, 2007.

Johanson, M., Burch, N., Valenzano, R., and Bowling, M. Evaluating state-space abstractions in extensive-form games. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 271–278, 2013.

Lanctot, M., Waugh, K., Zinkevich, M., and Bowling, M. Monte carlo sampling for regret minimization in extensive games. *Advances in Neural Information Processing Systems*, 22, 2009.

Lerer, A., Hu, H., Foerster, J., and Brown, N. Improving policies via search in cooperative partially observable games. In *AAAI Conference on Artificial Intelligence*, volume 34, pp. 7187–7194, 2020.

Li, J., Koyamada, S., Ye, Q., Liu, G., Wang, C., Yang, R., Zhao, L., Qin, T., Liu, T.-Y., and Hon, H.-W. Suphx: Mastering mahjong with deep reinforcement learning. *arXiv preprint arXiv:2003.13590*, 2020.

Li, X. and Miikkulainen, R. Dynamic adaptation and opponent exploitation in computer poker. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Lisý, V., Lanctot, M., and Bowling, M. H. Online monte carlo counterfactual regret minimization for search in imperfect information games. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 27–36, 2015.

Liu, M., Wu, C., Liu, Q., Jing, Y., Yang, J., Tang, P., and Zhang, C. Safe opponent-exploitation subgame refinement. *Advances in Neural Information Processing Systems*, 35:27610–27622, 2022.

Liu, W., Fu, H., Fu, Q., and Wei, Y. Opponent-limited online search for imperfect information games. In *International Conference on Machine Learning*, 2023.

Moravcik, M., Schmid, M., Ha, K., Hladik, M., and Gaukrodger, S. Refining subgames in large imperfect information games. In *AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M., and Bowling, M. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

Sandholm, T. Perspectives on multiagent learning. *Artificial Intelligence*, 171(7):382–391, 2007.

Southey, F., Bowling, M. H., Larson, B., Piccione, C., Burch, N., Billings, D., and Rayner, D. C. Bayes' bluff: Opponent modelling in poker. In *Uncertainty Artificial Intelligence*, pp. 550–558, 2005.

Šustr, M., Kovařík, V., and Lisý, V. Monte carlo continual resolving for online strategy computation in imperfect information games. In *International Conference on Autonomous Agents and MultiAgent Systems*, pp. 224–232, 2019.

Tian, Z., Wen, Y., Gong, Z., Punakkath, F., Zou, S., and Wang, J. A regularized opponent model with maximum entropy objective. In *International Joint Conference on Artificial Intelligence*, pp. 602–608, 2019.

Wu, Z., Li, K., Zhao, E., Xu, H., Zhang, M., Fu, H., An, B., and Xing, J. L2e: Learning to exploit your opponent. 2021. URL arXiv:2102.09381.

Yu, X., Jiang, J., Zhang, W., Jiang, H., and Lu, Z. Model-based opponent modeling. *Advances in Neural Information Processing Systems*, 35:28208–28221, 2022.

Zhang, B. and Sandholm, T. Subgame solving without common knowledge. *Advances in Neural Information Processing Systems*, 34:23993–24004, 2021.

Zheng, Y., Meng, Z., Hao, J., Zhang, Z., Yang, T., and Fan, C. A deep bayesian policy reuse approach against non-stationary agents. In *Advence in Neural Information Processing Systems*, pp. 962–972, 2018.

A. Proofs

A.1. Proof of Theorem 4.3

Proof. We adopt the proof structure outlined in (Burch et al., 2014) to establish Theorem 4.3 via induction based on the predecessor relationships within the game. Let $\mathbb{S}_{top} = \cup_{S\in\mathbb{S}} S_{top}$ be the earliest reach state in $\mathbb{S}$. Define $pre(I_1) = 0$ for all $I_1 \in \mathbb{S}_{top}$ and $pre(I_1') = \max_{I_1'\sqsubset I_1} pre(I_1) + 1$. Furthermore, we set $pre(\cdot)$ to 0 for states that cannot lead to $\mathbb{S}$.

According to the definition of σ_2', we have $CBV_1^{\sigma}(I_1) - CBV_1^{\sigma_2'}(I_1) \geq 0$ for all I_1 with $pre(I_1) = 0$.

For the inductive step, we assume $CBV_1^{\sigma}(I_1) - CBV_1^{\sigma_2'}(I_1) \geq 0$ holds for all I_1 with $pre(I_1) = k$. Now, let's consider an arbitrary I_1' with $pre(I_1') = k + 1$:

Case 1. If I_1' is the turn for player 1 to act, then $\forall a \in A(I_1'), pre(I_1' \cdot a) \leq k+1$, $CBV_1^{\sigma}(I_1' \cdot a) - CBV_1^{\sigma_2'}(I_1' \cdot a) \geq 0$. According to the definition of CBR, $CBR(I_1') \in \arg\max_{a\in A(I_1')} CBV_1^{\sigma}(I_1' \cdot a)$, and thus $CBV_1^{\sigma}(I_1') = \max_{a\in A(I_1')} CBV_1^{\sigma}(I_1' \cdot a) \geq \max_{a\in A(I_1')} CBV_1^{\sigma_2'}(I_1' \cdot a) = CBV_1^{\sigma_2'}(I_1')$.

Case 2. If I_1' is not the turn for player 1 to act, then player 2 will play according to σ_2 outside $\mathbb{S}$ and the chance player will always act according to a fixed distribution, $\pi_{-i}^{\langle CBR(\sigma_2),\sigma_2\rangle}(h) = \pi_{-i}^{\langle CBR(\sigma_2'),\sigma_2'\rangle}(h)$ for all $h \in I_1'$. It is followed by

$$
\begin{aligned}
CBV_1^{\sigma}(I_1') =& \frac{\sum_{h\in I_i}(\pi_{-i}^{\langle CBR(\sigma_2),\sigma_2\rangle}(h) v_i^{\langle CBR(\sigma_2),\sigma_2\rangle}(h))}{\sum_{h\in I_i}\pi_{-i}^{\langle CBR(\sigma_2),\sigma_2\rangle}(h)} \\
=& \frac{\sum_{h\in I_i, a\in A(h)}(\pi_{-i}^{\langle CBR(\sigma_2),\sigma_2\rangle}(h\cdot a) v_i^{\langle CBR(\sigma_2),\sigma_2\rangle}(h\cdot a))}{\sum_{h\in I_i}\pi_{-i}^{\langle CBR(\sigma_2),\sigma_2\rangle}(h)} \\
=& \frac{\sum_{h\in I_i, a\in A(h)}(\pi_{-i}^{\langle CBR(\sigma_2'),\sigma_2'\rangle}(h\cdot a) v_i^{\langle CBR(\sigma_2),\sigma_2\rangle}(h\cdot a))}{\sum_{h\in I_i}\pi_{-i}^{\langle CBR(\sigma_2'),\sigma_2'\rangle}(h)} \\
\geq& \frac{\sum_{h\in I_i, a\in A(h)}(\pi_{-i}^{\langle CBR(\sigma_2'),\sigma_2'\rangle}(h\cdot a) v_i^{\langle CBR(\sigma_2'),\sigma_2'\rangle}(h\cdot a))}{\sum_{h\in I_i}\pi_{-i}^{\langle CBR(\sigma_2'),\sigma_2'\rangle}(h)} \\
=& CBV_1^{\sigma'}(I_1')\,.
\end{aligned}
\tag{9}
$$

Therefore, $CBV_1^{\sigma}(I_1') - CBV_1^{\sigma_2'}(I_1') \geq 0$ holds for all I_1' with $pre(I_1') = k+1$. By induction, $CBV_1^{\sigma}(I_1) - CBV_1^{\sigma_2'}(I_1) \geq 0$ holds for all predecessors, including the root of the game, which implies $u_1(BR(\sigma_2), \sigma_2) \geq u(BR(\sigma_2'), \sigma_2')$. Hence,

$$
\exp(\sigma_2') = u_1(\sigma_1^*, \sigma_2^*) - u_1(BR(\sigma_2'), \sigma_2') \leq u_1(\sigma_1^*, \sigma_2^*) - u(BR(\sigma_2), \sigma_2) = \exp(\sigma_2). \tag{10}
$$

□

A.2. Proof of Theorem 4.4

Proof. Let $\sigma_2^m = \arg\max_{\sigma_2'} \min_{I_1^i\in S_{top}} \left(CBV_1^{\sigma_2}(I_1^i) - CBV_1^{\sigma_2'}(I_1^i)\right) \geq 0$, since Equation (1) is maximized,

$$
\begin{aligned}
& \sum_{I_1^i\in S_{top}} \hat{p}(I_1^i)\left(CBV_1^{\sigma}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i)\right) \geq \sum_{I_1^i\in S_{top}} \hat{p}(I_1^i)\left(CBV_1^{\sigma}(I_1^i) - CBV_1^{\sigma_2^m}(I_1^i)\right) \\
\Leftrightarrow \quad & \sum_{I_1^i\in S_{top}} \hat{p}(I_1^i)\left(CBV_1^{\sigma_2^m}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i)\right) \geq 0 \\
\Leftrightarrow \quad & \sum_{I_1^i\in S_{top}} \hat{p}(I_1^i)\left(CBV_1^{\sigma_2}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i)\right) \geq \sum_{I_1^i\in S_{top}} \hat{p}(I_1^i)\left(CBV_1^{\sigma_2}(I_1^i) - CBV_1^{\sigma_2^m}(I_1^i)\right) \\
\Leftrightarrow \quad & \sum_{I_1^i\in S_{top}} \hat{p}(I_1^i)\left(CBV_1^{\sigma_2}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i)\right) \geq \delta
\end{aligned}
\tag{11}
$$

Consider the modeling error ϵ,

$$\begin{aligned}&\sum_{I_1^i \in S_{top}} p(I_1^i)\left(CBV_1^{\sigma_2}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i)\right)\\ =&\sum_{I_1^i \in S_{top}} \left(p(I_1^i) - \hat{p}(I_1^i)\right)\left(CBV_1^{\sigma_2}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i)\right) + \sum_{I_1^i \in S_{top}} \hat{p}(I_1^i)\left(CBV_1^{\sigma_2}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i)\right)\end{aligned} \tag{12}$$

As constrained by Equation (2), $CBV_1^{\sigma}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i) \geq 0$, the first term in the right hand side of Equation (12) can be decomposed into two part $\{I_1^{i,+}\}$ and $\{I_1^{i,-}\}$, where $p(I_1^{i,+}) - \hat{p}(I_1^{i,+}) \geq 0$ and $p(I_1^{i,-}) - \hat{p}(I_1^{i,-}) < 0$. Thus,

$$\begin{aligned}&\sum_{I_1^i \in S_{top}} p(I_1^i)\left(CBV_1^{\sigma_2}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i)\right)\\ =&\sum_{I_1^i \in S_{top}} \hat{p}(I_1^i)\left(CBV_1^{\sigma_2}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i)\right) + \sum_{I_1^{i,+}} \left(p(I_1^{i,+}) - \hat{p}(I_1^{i,+})\right)\left(CBV_1^{\sigma_2}(I_1^{i,+}) - CBV_1^{\sigma_2^S}(I_1^{i,+})\right)\\ +&\sum_{I_1^{i,-}} \left(p(I_1^{i,-}) - \hat{p}(I_1^{i,-})\right)\left(CBV_1^{\sigma_2}(I_1^{i,-}) - CBV_1^{\sigma_2^S}(I_1^{i,-})\right)\end{aligned} \tag{13}$$

Apparently, $\sum_{I_1^{i,+}} \left(p(I_1^{i,+}) - \hat{p}(I_1^{i,+})\right)\left(CBV_1^{\sigma_2}(I_1^{i,+}) - CBV_1^{\sigma_2^S}(I_1^{i,+})\right) \geq 0$. Since $p(I_1^{i,-}) - \hat{p}(I_1^{i,-}) < 0$ and $p(I_1^{i,-}) \geq 0$, we have $\hat{p}(I_1^{i,-}) > 0$. Therefore,

$$\begin{aligned}\sum_{I_1^{i,-}} \left(p(I_1^{i,-}) - \hat{p}(I_1^{i,-})\right)\left(CBV_1^{\sigma_2}(I_1^{i,-}) - CBV_1^{\sigma_2^S}(I_1^{i,-})\right) \geq& -\epsilon \sum_{I_1^{i,-}} \hat{p}(I_1^{i,-})\left(CBV_1^{\sigma_2}(I_1^{i,-}) - CBV_1^{\sigma_2^S}(I_1^{i,-})\right)\\ \geq& -\epsilon \sum_{I_1^i \in S_{top}} \hat{p}(I_1^i)\left(CBV_1^{\sigma_2}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i)\right)\end{aligned} \tag{14}$$

Back to Equation (13)

$$\begin{aligned}\sum_{I_1^i \in S_{top}} p(I_1^i)\left(CBV_1^{\sigma_2}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i)\right) \geq& (1-\epsilon) \sum_{I_1^i \in S_{top}} \hat{p}(I_1^i)\left(CBV_1^{\sigma_2}(I_1^i) - CBV_1^{\sigma_2^S}(I_1^i)\right)\\ \geq& (1-\epsilon)\delta,\end{aligned} \tag{15}$$

which means

$$u_2^{\langle BR_1^{[S,\sigma_1]}(\sigma_2'),\sigma_2'\rangle}(S) \geq u_2^{\langle BR_1^{[S,\sigma_1]}(\sigma_2),\sigma_2\rangle}(S) + (1-\epsilon)\delta \tag{16}$$

□

A.3. Proof of Theorem 4.5

Proof. Suppose $\forall h_o$, $\sigma^{S*}(h_o, enter) < 1$. Since σ^{S*} is the NE, then $\sigma_1^{S*}(h_o)$ is a best response to σ_1^{S*}. $\forall h_o$, $\sigma^{S*}(h_o, enter) < 1$ implies $v_1^{\sigma^{S*}}(h_o, enter) = v_1^{\sigma^{S*}}(h_o, out) = 0$ and player 1 can not benefit from changing the strategy at h_o. Therefore, $\exists \boldsymbol{\lambda}' = (\sigma^{S*}(h_o^1, enter)\beta, \sigma^{S*}(h_o^2, enter)\beta, \ldots, \sigma^{S*}(h_o^k, enter)\beta) \in [0,\beta]^k$ is optimal in Equation (6), which conflicts with the condition. Thus, $\exists h_o$, $\sigma^{S*}(h_o, enter) = 1$. □

A.4. Proof of Theorem 4.6

Proof. Let $M^{\sigma_2^S}(I_1^i) = CBV_1^{\sigma_2^S}(I_1^i) - CBV_1^{\sigma}(I_1^i) \geq -\Delta$, Since σ_2^S is the optimal strategy in Equation (7), then

$$\begin{aligned}&\max_{0\leq\boldsymbol{\lambda}\leq\beta} \sum_{I_1^i \in S_{top}} \hat{p}(I_1^i) M^{\sigma_2^S}(I_1^i) + \sum_{I_1^i \in S_{top}} \lambda_i M^{\sigma_2^S}(I_1^i)\\ \leq&\max_{0\leq\boldsymbol{\lambda}\leq\beta} \sum_{I_1^i \in S_{top}} \hat{p}(I_1^i) M^{\sigma_2}(I_1^i) + \sum_{I_1^i \in S_{top}} \lambda_i M^{\sigma_2}(I_1^i) = 0.\end{aligned} \tag{17}$$

Therefore,

$$\max_{0\leq\boldsymbol{\lambda}\leq\beta}\sum_{I_1^i\in S_{top}}\lambda_i M^{\sigma_2^S}(I_1^i)\leq-\sum_{I_1^i\in S_{top}}\hat{p}(I_1^i)M^{\sigma_2^S}(I_1^i)\leq\Delta, \tag{18}$$

Since $\boldsymbol{\lambda}$ is optimal, then if $M^{\sigma_2^S}(I_1^i)\leq 0$, then $\lambda_i=0$. It implies

$$\max_{I_1^i\in S_{top}} CBV_1^{\sigma_2^S}(I_1^i)-CBV_1^{\sigma}(I_1^i)\leq\frac{\Delta}{\beta}. \tag{19}$$

Thus,

$$\exp(\sigma_2^S)-\exp(\sigma)\leq\frac{\Delta}{\beta}. \tag{20}$$

□

B. Implementation Details

Same as the approach outlined in (Liu et al., 2022), the action probability in each infoset is multiplied by a random variable with probability $\Pr_{\text{shuffle}}$. The estimated error is implemented by first dividing the earliest reached state into two sets and then applying reject sampling to add/subtract the given error to/from each set.

For Leduc Hold'em, the blueprint is solved by Monte Carlo CFR (Lanctot et al., 2009) for 1,000,000 iterations. No abstraction technique is used in Leduc Hold'em and the opponent exploitation methods are applied at each state after the public card is dealt. For the chance outcomes of the root node in the gadget game, we set $\frac{1}{k\beta+1}$ to $\frac{1}{16}$ in Leduc Hold'em.

In the case of Flop Hold'em Poker, the blueprint is solved by Monte Carlo CFR for 100,000 iterations, incorporating the abstraction technique (Johanson et al., 2013) at the flop turn. For each public betting history, the infosets are clustered into 200 buckets. The opponent with $\Pr_{\text{shuffle}}=0.0$ is also set to the same as the blueprint. The opponent exploitation method is applied right after the public cards are dealt, with a finer-grained abstraction that has 400 buckets for each public betting history. Given that there are 400 alteration nodes for the opponent in the gadget game, we increase the value of $\frac{1}{k\beta+1}$ to $\frac{1}{51}$, as per Theorem 4.6, to mitigate the potential increase in exploitability.

C. Discussion about OX-Search and Maxmargin Subgame Solving.

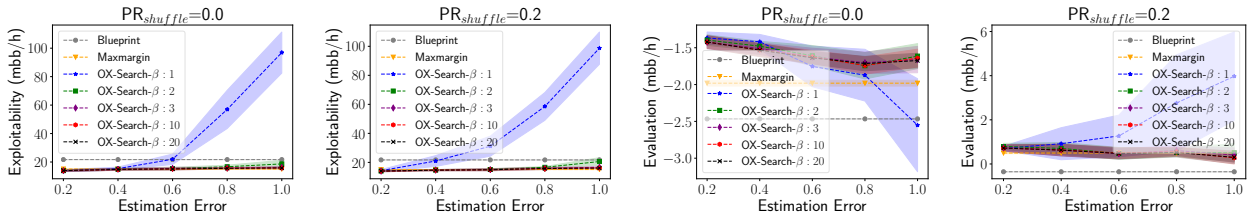
The goal of MaxMargin can be reformulated as $\max_{\sigma_2^S}\min_{\lambda,\sigma_1}\lambda_i(CBV_1^{\sigma}(I_1^i)-v_1^{\langle\sigma_1,\sigma_2^S\rangle}(I_1^i))$, $s.t.\sum\lambda_i=1$, which can be simply constructed as a zero-sum game with λ representing the action probability of player 1's root node.

The gadget game designed for OX-Search (especially the right hand side) is inspired by the Subgame Resolving and the MaxMargin method. Subgame Rsolving method aims to find a strategy σ_2^S such that $CBV_1^{\sigma}(I_1^i)-CBV_1^{\sigma_2^S}(I_1^i)\geq 0$ for all I_1^i . Since OX-Search does not require the subgame margin to be maximized, the gadget game has to give the opponent an oppotunity to "opt out" as in Subgame Resolving. The players' value is also shifted by CBV_1^{σ} in order to reflect the violation of the safety constraints. Thus the right hand side of the gadget game is constructed as a combination of MaxMargin gadget game and Subgame Resolving gadget game according to Equation (7).

While MaxMargin aims to find a better strategy that is less exploitable, it could be regarded as a "conservative" opponent exploitation method, as it tries to maximize the mimium improvement $m=\max_{\sigma_2^S}\min_{I_1^i}CBV_1^{\sigma}(I_1^i)-CBV_1^{\sigma_2^S}(I_1^i)$ at each opponent infoset. Conversely, OX-Search adopts a more flexible approach by not insisting on maximizing the subgame margin $\min_{I_1^i}CBV_1^{\sigma}(I_1^i)-CBV_1^{\sigma_2^S}(I_1^i)$, but rather allowing it to be non-negative. The underlying intuition is that the estimated opponent's strategy implicitly gives us a gift of m at each infoset compared to MaxMargin, such that we can afford to let player 1's value increase beyond the $CBV_1^{\sigma}(I_1^i)-m$ at low reach probability infoset I_1^i and lower the value of other infoset, thereby leveraging the estimated opponent's strategy. The idea shares similarities with Reach Subgame Solving, which improves the MaxMargin by utilizing another kind of "gift" to increase the alternative value, consequently reducing the exploitability of the subgame solving strategy.

D. Ablation Studies of β and Comparison with Maxmargins.

See Figure 5 and Figure 6.



(a) Exploitability. (b) Exploitability. (c) Evaluation. (d) Evaluation.

Figure 5. Performance of OX-Search with different β and Maxmargin subgame solving in Leduc poker.

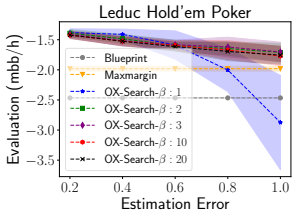
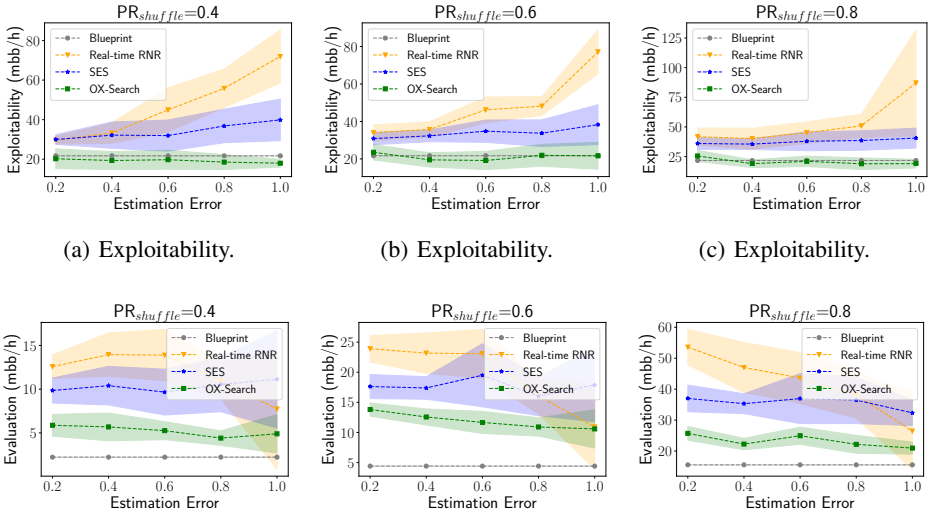


Figure 6. Performance of OX-Search with different β and Maxmargin subgame solving in Leduc poker against evolving opponents.

E. Experiments against Weaker Opponents.

See Figure 7. For weaker opponent such as $\text{PR}_{\text{shuffle}} = 0.4, 0.6, 0.8$, OX-Search demonstrated superior exploitability preservation while SES and Real-time RNR fail even with small estimation errors. However, due to its inherent safety constraints, OX-Search could not exploit these opponents as aggressively as SES and Real-time RNR could, despite achieving notable utility improvements over the blueprint.



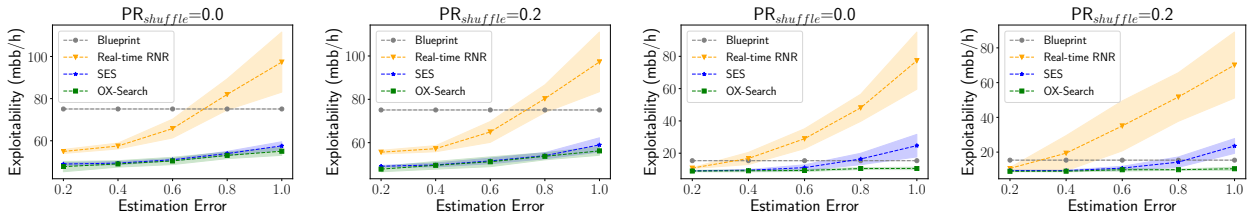
(a) Exploitability. (b) Exploitability. (c) Exploitability.

(d) Evaluation. (e) Evaluation. (f) Evaluation.

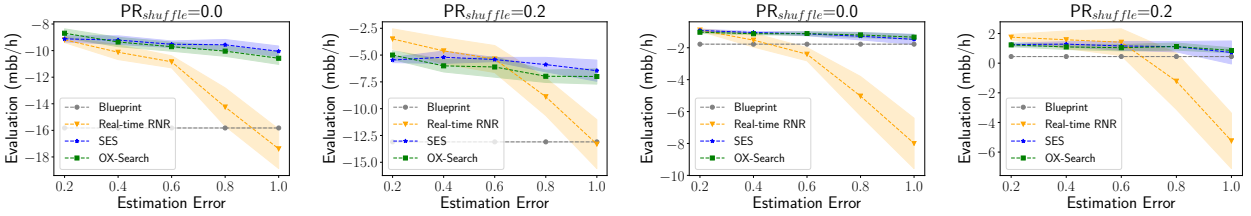
Figure 7. Performance of OX-Search and other methods in Leduc poker.

F. Experiments of Different Blueprint Strategies.

See Figure 8. In contexts with a stronger blueprint, OX-Search uniquely managed to maintain safety and achieve utility improvements even with significant estimation errors (e.g., error = 1), showcasing its robustness and safety-oriented design.



(a) Exploitability using the bluprint strategy of 100K iterations.
(b) Exploitability using the bluprint strategy of 100K iterations.
(c) Exploitability using the bluprint strategy of 10M iterations.
(d) Exploitability using the bluprint strategy of 10M iterations.



(e) Evaluations using the bluprint strategy of 100K iterations.
(f) Evaluations using the bluprint strategy of 100K iterations.
(g) Evaluations using the bluprint strategy of 10M iterations.
(h) Evaluations using the bluprint strategy of 10M iterations.

Figure 8. Performance of OX-Search and other methods in Leduc poker using different bluprint strategies.

For weaker blueprint, OX-Search could still remain comparable performance with SES. We can find that the blueprint strategy indeed influence the final results, since the weaker blueprint gives more space for OX-Search to exploit more about the opponent.

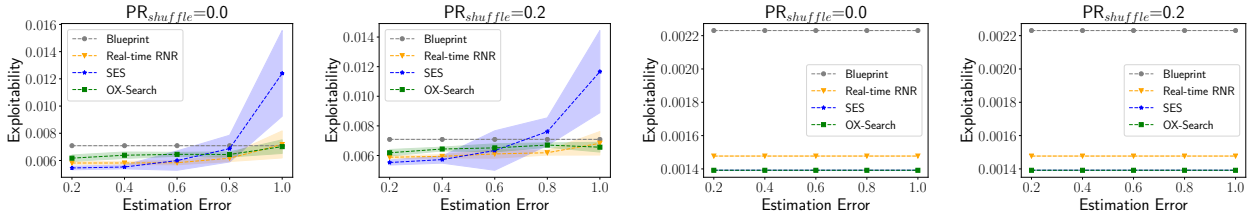
G. Experiments in Other Games

Further experimentation in games like Liar's Dice and Goofspiel, with $k\beta$ set to 15, has provided additional insights as shown in Figure 9. The results in Liar's Dice implies OX-Search can be well generalized to this game, while increasing the β value may help to improve the performance since some result strategy violate the constraints. Goofspiel presented a surprising outcome: exploitability remained consistent regardless of estimation error. We hypothesize this is attributable to Goofspiel's unique information structure, where players' actions are not directly observable, possibly rendering estimation errors less impactful on minimizing CBVs.

H. Further Experiments in FHP.

We conducted additional experiments on MaxMargin subgame solving in Flop Hold'em Poker, as results depicted in Figure 10. The results indicate that MaxMargin can enhance both exploitability and exploitation to some extent. Notably, the improvements to the blueprint strategy by MaxMargin were not as substantial as anticipated based on previous literature. This discrepancy may be attributed to the application of additional techniques in prior works that were not implemented in our current study. Despite this, the experiment is a fair comparison, since we consistently use the same solver across all the experiments.

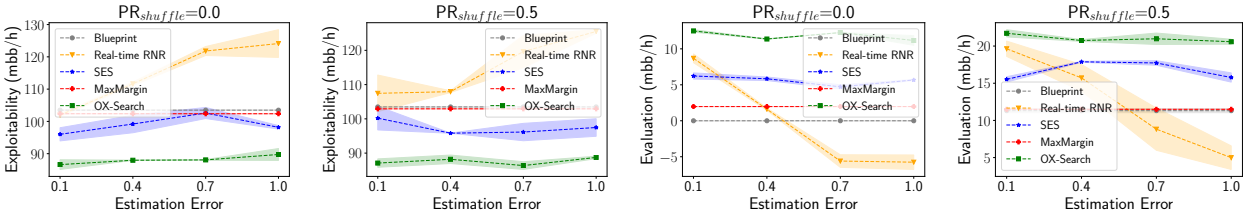
We also carried out ablation studies on the parameter β in Flop Hold'em Poker. Preliminary findings suggest that a smaller β enhances exploitation by directing OX-Search to prioritize exploitation more intensely. Conversely, a larger β does not significantly alter the outcomes, as β essentially serves as an upper bound and OX-Search will automatically adjust its focus between exploitation and safety based.



(a) Exploitability in Liar's Dice. (b) Exploitability in Liar's Dice. (c) Exploitability in Goofspiel. (d) Exploitability in Goofspiel.

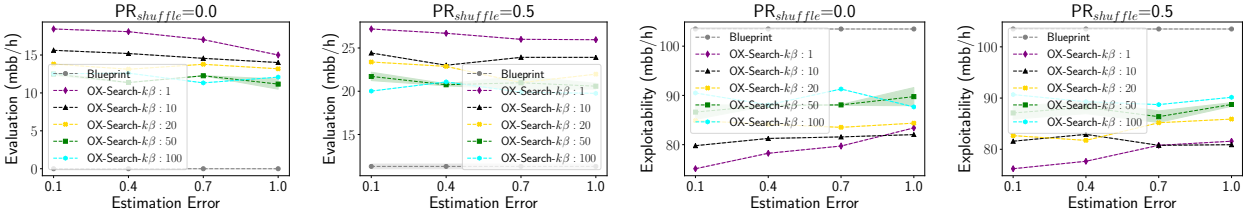
(e) Evaluation in Liar's Dice. (f) Evaluation in Liar's Dice. (g) Evaluation in Goofspiel. (h) Evaluation in Goofspiel.

Figure 9. Performance of OX-Search and other methods in Liar's Dice and 5-card Goofspiel with 3 round ascending order.



(a) Exploitability. (b) Exploitability. (c) Evaluation. (d) Evaluation.

Figure 10. Performance of OX-Search and other methods in FHP.



(a) Evaluation. (b) Evaluation. (c) Exploitability (d) Exploitability

Figure 11. Performance of OX-Search with different β in FHP.