OFFLINE EQUILIBRIUM FINDING IN EXTENSIVE-FORM GAMES: DATASETS, METHODS, AND ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline reinforcement learning (Offline RL) has emerged as a promising paradigm for solving real-world decision-making problems using pre-collected datasets. However, its counterpart for extensive-form games (EFGs) has received little attention. To fill this gap, we introduce the *offline equilibrium finding (Offline EF)* paradigm, which aims to compute equilibrium strategies of an EFG from a fixed offline dataset without interacting with the game. Offline EF is harder than offline RL, and the difference lies in the solution concept: offline RL seeks an optimal policy, whereas offline EF seeks an equilibrium. In single-agent offline RL, deciding whether one action is better than another only requires comparing the two in the data. An equilibrium, on the other hand, is a claim about every player's best response: no one can gain by switching to any other strategy. A dataset that records only how the players actually played says nothing about these deviations, and so cannot certify an equilibrium. Existing equilibrium-finding algorithms sidestep this by querying the game for best responses, which a fixed dataset cannot answer. Additionally, there are no public datasets or evaluation protocols for this problem, which leaves methods incomparable and progress hard to measure. To address these issues, we first construct a suite of offline datasets over a range of EFGs, spanning random, expert, learning, and hybrid data-generating strategies. We then propose BOMB, a framework that combines behavior cloning with a model-based method and allows any online equilibrium-finding algorithm (e.g., CFR, PSRO) to be used in the offline setting. We identify the sufficient conditions under which each component of BOMB is guaranteed to converge to an equilibrium, and show that BOMB, with its optimal mixing weight, is at least as good as the better of its two components. Extensive experiments show that BOMB consistently outperforms offline RL baselines and computes approximate Nash and coarse correlated equilibria in both two-player and multi-player games.

1 INTRODUCTION

Extensive-form games (EFGs) offer a powerful framework for modeling sequential interactions among multiple players under imperfect information and chance (Nisan et al., 2007). The canonical solution concept is the Nash Equilibrium (NE), in which no player can increase their own utility by unilaterally deviating. A range of methods have been developed for solving EFGs, including sequence-form linear programming (Shoham & Leyton-Brown, 2008), double-oracle algorithms (McMahan et al., 2003), counterfactual regret minimization (CFR) (Zinkevich et al., 2007), and policy-space response oracles (PSRO) (Lanctot et al., 2017). These techniques have been successfully applied to large-scale EFGs, such as pursuit-evasion games (Xue et al., 2021; Li et al., 2023; 2024), poker (Brown & Sandholm, 2019; Zha et al., 2021; Brown & Sandholm, 2018), and Stratego (Perolat et al., 2022). Despite their successes, existing equilibrium-finding algorithms rely on interactions with the game or access to an accurate simulator of it. For example, CFR-based algorithms require traversing the game tree to compute regret values, while PSRO and its variants depend on interactions with the game environment to train best response strategy and estimate entries in the meta-game. We refer to this paradigm as "*online equilibrium finding*". In many applications, however, such as sports games (Liu et al., 2022), network intrusion detection (Khraisat et al., 2019), and automated negotiations (Kiruthika et al., 2020), interacting with the environment is costly or unsafe, and building a faithful simulator is infeasible. What is usually available instead is a record

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

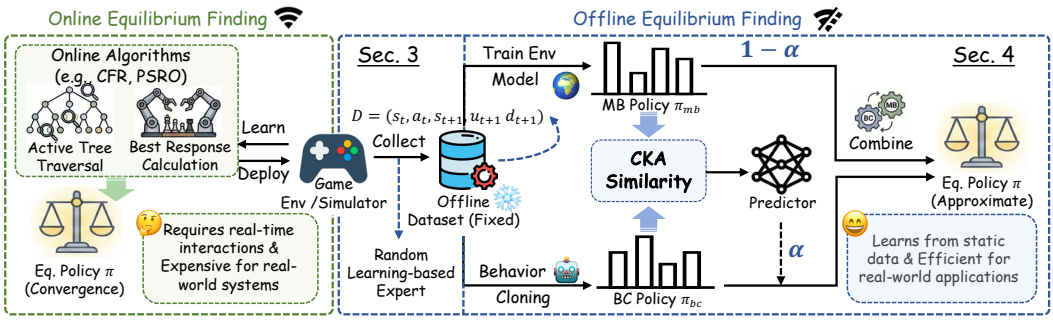


Figure 1: Comparison between online and offline equilibrium finding

of past play, such as match logs, network traces, and negotiation transcripts. Learning equilibrium strategies from such data offline is therefore a natural alternative (Fig. 1).

Offline learning in games has begun to receive attention. StarCraft II Unplugged (Mathieu et al., 2021) provides a dataset of human play for a two-player zero-sum symmetric game, and a line of theoretical work (Cui & Du, 2022; Zhong et al., 2022) studies which dataset properties allow an NE of a two-player zero-sum Markov game to be recovered from an offline dataset. This work, however, focuses on Markov games, where the state is fully observed. It does not address the imperfect-information structure of EFGs, and provides neither practical algorithms nor datasets against which algorithms could be compared. To the best of our knowledge, the offline setting for EFGs has not received much attention. To bridge this gap, we introduce the ***offline equilibrium finding*** (Offline EF) paradigm, which aims to compute equilibrium strategies of an EFG from a fixed dataset without interacting with the game. Offline EF is harder than offline RL, and the difference lies in the solution concept. In single-agent offline RL, deciding whether one action is better than another requires only comparing the two in the data. The target of offline EF is an equilibrium, which holds when no player has a profitable deviation. Certifying one requires the counterfactual payoffs of deviations, and these are exactly what a record of actual play leaves out. Online algorithms such as CFR and PSRO avoid this by querying the game, while a fixed dataset cannot be queried. One might instead borrow the two standard recipes of offline RL, imitation learning and model-based learning, but each has a blind spot here. Cloning the behavior in the data inherits its quality, so it fails on data from unskilled play. Fitting a model of the game to the data and solving it inherits the data's coverage, so it fails when play was concentrated on a few lines of the tree.

This paper provides a first systematic treatment of offline EF, with four contributions. i) We construct a suite of offline datasets for a range of EFGs, generated by random, expert, and learning strategies, together with an evaluation protocol that varies the data composition to test how each method depends on it; ii) we propose BOMB, a framework that combines behavior cloning with a model-based method, which allows any online equilibrium-finding algorithm to be used offline; iii) we identify the dataset coverage conditions under which each component can converge to an equilibrium, and show that BOMB, with its mixing weight selected by grid search, is at least as good as the better of its two components, even when the weight is selected entirely offline in the learned game; and iv) experimental results show that BOMB consistently outperforms offline RL baselines and computes approximate Nash and coarse correlated equilibria in both two-player and multi-player games.

2 Preliminaries

2.1 Imperfect-Information Extensive-Form Games(IIEFGs)

An IIEFG can be represented as a tuple, formally, $\mathcal{G} = (N, \mathcal{H}, A, P, \mathcal{I}, u)$ (Shoham & Leyton-Brown, 2008). $N = \{1, ..., n\}$ is the set of players, and $\mathcal{H}$ is the set of histories, representing all possible action sequences, including the root node of the game tree (empty sequence $\emptyset$) and all prefixes of sequences in $\mathcal{H}$. The set of terminal histories, represented by Z, represents the set of game outcomes, $Z \subseteq \mathcal{H}$. $A(h) = \{a : (h, a) \in \mathcal{H}\}$ is the set of available actions at any non-terminal history $h \in \mathcal{H} \setminus Z$. P is the player function mapping non-terminal histories to players, i.e., $P(h) \mapsto N \cup \{c\}, \forall h \in \mathcal{H} \setminus Z$, where c represents the "chance player" for stochastic events outside players' control. $\mathcal{I}$ is the set of information sets, forming a partition over histories where player i

takes actions, such that player i cannot distinguish these histories within the same information set I_i. Each information set $I_i \in \mathcal{I}_i$ represents one decision point which means that $P(h_1) = P(h_2)$ and $A(h_1) = A(h_2)$ for any $h_1, h_2 \in I_i$. For convenience, $A(I_i)$ and $P(I_i)$ denote the action set $A(h)$ and player $P(h)$ for any $h \in I_i$. For each player i, a utility function $u_i : Z \to \mathbb{R}$ defines the payoff for every terminal history. The behavior strategy of player i, σ_i, maps each information set of i to a probability distribution over $A(I_i)$, with Σ_i denoting the set of all strategies for player i. A strategy σ_i is *pure* if it always selects a single action (i.e., $\sigma_i(I_i, a) \in \{0, 1\}$), *mixed* if it assigns probabilities in $[0, 1]$, and *fully mixed* if all probabilities are strictly positive. A strategy profile $\sigma = (\sigma_1, \sigma_2, ..., \sigma_n)$ is a tuple of strategies for all players, with σ_{-i} referring to all strategies except σ_i. The reaching probability of a history h under σ is $\pi^\sigma(h) = \prod_{i \in N \cup \{c\}} \pi_i^\sigma(h)$. Given σ, the expected payoff for player i is $u_i(\sigma) = \sum_{z \in Z} \pi^\sigma(z) u_i(z)$, summing over the terminal nodes.

The common solution concept for an IIEFG is Nash equilibrium (NE) (Nash, 1950), in which no player can increase their utility by unilaterally deviating. Formally, a strategy profile σ^* forms an NE if it satisfies $\forall i \in N, u_i(\sigma^*) = \max_{\sigma_i' \in \Sigma_i} u_i(\sigma_i', \sigma_{-i}^*)$. To measure the distance from NE in N-player games, we use $\text{NASHCONV}(\sigma)$ as the metric, where $\text{NASHCONV}(\sigma) = \sum_{i \in N} \text{NASHCONV}_i(\sigma) = \sum_{i \in N} (\max_{\sigma_i'} u_i(\sigma_i', \sigma_{-i}) - u_i(\sigma))$. When $\text{NASHCONV}(\sigma) = 0$, then the strategy profile σ is an NE. Apart from the NE solution, in N-player general-sum games, (Coarse) Correlated Equilibrium ((C)CE) is also a common solution concept (Aumann, 1987). A (C)CE is a joint mixed strategy in which no player has an incentive to deviate. Let S_i be the strategy space for player i. A strategy profile σ^* forms a CCE if $\forall i \in N, s_i \in S_i, u_i(\sigma^*) \geq u_i(s_i, \sigma_{-i}^*)$. where σ_{-i}^* is the marginal distribution of σ^* on strategy space S_{-i}. To measure the distance from (C)CE, the (C)CE Gap Sum is adopted as the metric (Marris et al., 2021).

2.2 WHY EXISTING METHODS FAIL?

Offline reinforcement learning (Offline RL) primarily targets single-agent settings (Levine et al., 2020) and is inherently incapable of computing equilibrium strategies in games. Opponent modeling (OM) focuses on predicting the behavioral strategies of opponents (He et al., 2016), but it is also designed to compute the best response strategy for a single player rather than equilibrium strategies. Furthermore, OM requires access to the game environment, making it unsuitable for offline EF. Existing equilibrium-finding algorithms, such as no-regret methods (e.g., CFR (Zinkevich et al., 2007)) and empirical game-theoretic analysis methods (e.g., PSRO (Lanctot et al., 2017)), fundamentally rely on repeated interactions with the game environments or access to an accurate game simulator. These methods, referred to as "Online Equilibrium Finding (Online EF)", are therefore not directly applicable to the offline setting. A comparison of existing methods is provided in Tab. 1, with further discussion in Appendix B.

Methods	Work w/o env	Converge to equilibrium
Offline RL	✓	✗
OM	✗	✗
Online EF	✗	✓

Table 1: Issues of Existing Methods.

2.3 PROBLEM STATEMENT

Definition 2.1 (Offline EF)**.** Let $\mathcal{D}$ be an offline dataset of an IIEFG $\mathcal{G}$, generated by an unknown strategy $\sigma_{\mathcal{D}}$. An *offline equilibrium finding* algorithm maps $\mathcal{D}$ to a strategy profile $\widehat{\sigma}$ without interacting with $\mathcal{G}$. Its output is evaluated by the equilibrium gap of $\widehat{\sigma}$ in $\mathcal{G}$, $\text{GAP}_{\mathcal{G}}(\widehat{\sigma})$. The goal of offline EF is to output $\widehat{\sigma}$ with $\text{GAP}_{\mathcal{G}}(\widehat{\sigma}) \leq \epsilon$, i.e., an ϵ-equilibrium of $\mathcal{G}$, with ϵ as small as $\mathcal{D}$ allows.

Building on the definition of the offline EF paradigm, we can instantiate it by defining $\text{GAP}_{\mathcal{G}}$, such as the NASHCONV for NE and the (C)CE Gap Sum for (C)CE. While offline EF shares some similarities with offline RL, it introduces unique challenges. First, unlike offline RL, which aims to compute an optimal strategy (Levine et al., 2020), offline EF seeks to achieve an equilibrium strategy. This requires iterative processes for best response calculations, adding significant complexities. Second, offline EF inherently involves at least two players, making the game dynamics highly sensitive to distribution shifts and uncertainties. Third, in offline RL, the data from two actions may suffice to determine which action is better. However, in offline EF, simply comparing data from two action tuples is insufficient to identify which action tuple is closer to an equilibrium strategy, as equilibrium identification requires additional action tuples for references (Cui & Du, 2022).

3 DATASETS

3.1 FORMATS

The dataset is represented as $\mathcal{D} = (s_t, a_t, s_{t+1}, u_{t+1}, d_{t+1})$. s_t and s_{t+1} represent the game states at time step t and $t+1$ from a game-level perspective. We slightly overload the term "state" here, using it in a sense distinct from the previously defined notions of history and information set. Specifically, s_t includes all information at t, including the information sets for each player $(I_1^t, I_2^t, ..., I_n^t)$, additional game-level information outside the players' control (GI), the acting player at t (p^t), and the available action set for the acting player $(A(I_{p^t}^t))$. Formally, $s_t = (I_1^t, I_2^t, ..., I_n^t, GI, p^t, A(I_{p^t}^t))$, where p^t may also be a chance player c. a_t is the action taken at t. $u_{t+1} = (u_1^{t+1}, u_2^{t+1}, ..., u_n^{t+1})$ represents the utilities for all players at time step $t+1$. Finally, d_{t+1} indicates whether the game ends at state s_{t+1}, with a value of 1 if the game ends and 0 otherwise. Fig. 2 presents an example of a two-player IIEFG $\mathcal{G}$, where I_1 and I_2 denote the information sets for Player 1 and Player 2, respectively. If the offline dataset $\mathcal{D}$ covers gameplay data of $\mathcal{G}$, it contains the following data points:

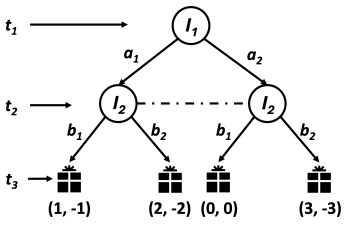


Figure 2: An example game.

$$((I_1^{t_1} = I_1, I_2^{t_1} = \emptyset, \emptyset, 1, \{a_1, a_2\}), a_1, (I_1^{t_2} = I_1 a_1, I_2^{t_2} = I_2, \emptyset, 2, \{b_1, b_2\}), (0, 0), 0),$$

.....

$$((I_1^{t_2} = I_1 a_2, I_2^{t_2} = I_2, \emptyset, 2, \{b_1, b_2\}), b_2, (I_1^{t_3} = I_1 a_2, I_2^{t_3} = I_2 b_2, \emptyset, -1, \emptyset), (3, -3), 1).$$

Note that in states $(I_1^{t_2} = I_1 a_1, I_2^{t_2} = I_2, \emptyset, 2, \{b_1, b_2\})$ and $(I_1^{t_2} = I_1 a_2, I_2^{t_2} = I_2, \emptyset, 2, \{b_1, b_2\})$, the information set for Player 2, $I_2^{t_2}$, is identical, as shown in Fig. 2. However, since our dataset is constructed from the perspective of the entire game, these states can still be distinguished by incorporating the game information of other players together with the game information GI. In this example, there is no chance node, so GI is empty. If the game were to include a chance node, its outcomes would be recorded in GI. This additional information allows us to differentiate game states that may otherwise appear indistinguishable from the perspective of a single player. In this way, the dataset provides sufficient detail to differentiate all relevant states in the game.

3.2 COLLECTING METHODS

In practice, an offline dataset is generated by whatever strategies were deployed, and these are unknown to the learner. Thus, a benchmark built on a single dataset cannot evaluate a method comprehensively. What matters for offline EF is the quality of the behavior that generated the data and the portion of the game the data cover, and the two tend to pull against each other: near-equilibrium play data cover few lines of the game tree, while play data that covers the full tree is far from equilibrium. A useful benchmark must span this trade-off. Offline RL benchmarks span an analogous range by mixing sources, from exploratory agents and intermediate checkpoints of online training to expert demonstrations (Fujimoto et al., 2019; Gulcehre et al., 2020; Fu et al., 2020). We follow the same principle and collect data with three methods, plus a hybrid that interpolates between the extremes.

The first one, *random method*, assumes that players follow uniform strategies and repeatedly play the game to generate gameplay data, thereby producing behavior that resembles the exploratory patterns of novices. This method is motivated by the natural tendency toward exploration and is intended to mimic the experience of a beginner becoming familiar with the game for the first time. The second method, *learning-based method*, collects intermediate interactions produced by equilibrium-finding algorithms, such as CFR (Zinkevich et al., 2007) or PSRO (Lanctot et al., 2017), reflecting gradual skill improvement. This method draws its inspiration from the process of a player learning towards an equilibrium strategy. The third, *expert method*, generates data by having players follow equilibrium strategies, thereby capturing expert-level gameplay. This inspiration comes from the notion that, when learning a game, we often benefit from observing the strategies of more experienced players. In addition, we also consider a *hybrid method* that mixes datasets generated using random and expert methods in different proportions, improving diversity and thus yielding more comprehensive benchmarks. For convenience, we name these datasets according to their collection methods, namely the *random dataset*, *learning dataset*, *expert dataset*, and *hybrid dataset*.

3.3 STATISTICS OF DATASETS

We developed a benchmark dataset for offline EF using these outlined collection methods across **eight** commonly used IIEFGs, as depicted in Fig. 3. The dataset contains approximately **3.8 million** data points, occupying about **11GB** of memory. For each game, we generated three distinct types of datasets: expert dataset, random dataset, and learning dataset. The proportions of each dataset are presented visually, and comprehensive statistics on their distributions are provided in Appendix C.1.

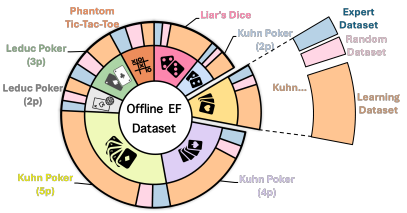


Figure 3: Dataset of Offline EF.

4 BOMB

Inspired by offline RL algorithms, there are two main directions to develop the algorithmic framework for offline EF: i) behavior cloning (BC) (Fujimoto & Gu, 2021), which imitates the strategies used to generate the offline data with additional exploration, and ii) model-based methods (Yu et al., 2020; Kidambi et al., 2020), which learn a world model from the offline dataset and then derive strategies from the world model. However, BC may fail in offline EF when random policies generate the dataset, as such strategies are often far from equilibrium. Similarly, model-based methods struggle with offline EF when the dataset is generated from equilibrium strategies, which often cover only a limited portion of the game states. To mitigate these issues, we propose BOMB, which combines **B**ehavior cl**O**ning and **M**odel-**B**ased methods for the offline EF paradigm.

4.1 BOMB FRAMEWORK

BOMB. Algorithm 1 demonstrates the BOMB framework. Given an offline dataset $\mathcal{D}$, it starts by training the σ_{BC} policy using the behavior cloning method (BC) (Line 2). Note that the dataset $\mathcal{D}$ contains game state $s_t = (I_1^t, I_2^t, ..., I_n^t, GI, p^t, A(I_{p^t}^t))$ and σ_{BC} mimics the behavior strategy, only $I_{p^t}^t$ and the corresponding action a_t are used for training. The training loss is the cross-entropy loss: $\mathcal{L}_{bc} = -\mathbb{E}_{(I_{p^t}^t, a_t)\sim\mathcal{D}}[a_t \cdot \log(\sigma(I_{p^t}^t; \theta))]$. Additionally, inspired by model-based offline RL algorithms (Kidambi et al., 2020; Yu et al., 2020), a dynamic environment model E_{θ_e} is trained based on dataset $\mathcal{D}$ (Lines 3). The model simulates the real environment and is used to derive the model-based (MB) policy σ_{MB} using any online EF algorithm, such as PSRO (Lanctot et al., 2017) and CFR (Zinkevich et al., 2007) (Line 4). Specifically, we use the game state s_t and action a_t as inputs, with the subsequent game state s_{t+1}, reward u_{t+1}, and termination variable d_{t+1} serving as labels. The model is trained using stochastic gradient descent (SGD) with the mean squared error loss: $\mathcal{L}_{env} = \mathbb{E}_{\mathcal{D}}[\textbf{MSE}((s_{t+1}, u_{t+1}, d_{t+1}), E(s_t, a_t; \theta_e))]$. The final policy combines the BC policy σ_{BC} and MB policy σ_{MB} using a weighted combination: $\sigma = \alpha\sigma_{BC} + (1-\alpha)\sigma_{MB}$, where α is the weight of the BC policy (Lines 5-6). The method for estimating the parameter α is described below.

Algorithm 1 BOMB Framework

1: **Input:** an offline dataset $\mathcal{D}$
2: Train σ_{BC} based on $\mathcal{D}$ using BC;
3: Train an env. model E_{θ_e} based on $\mathcal{D}$;
4: Learn σ_{MB} using any EF algorithm on E_{θ_e};
5: Select α using parameter estimation method;
6: $\sigma = \alpha \cdot \sigma_{BC} + (1-\alpha) \cdot \sigma_{MB}$;
7: **Output:** Policy σ

Estimation of Parameter α. We propose three methods of estimating the parameter α. The simplest one is the *random selection method*, in which a value is drawn at random from the interval $[0, 1]$. It is fully offline and easy to implement, but lacks optimal guarantees. The second method is *grid search method*, where we predefine a set of 11 values for α, i.e., $\{0, 0.1, ..., 1\}$. These values are used to configure combined policies, which are then tested in a real game. The value of α that results in the smallest gap from the equilibrium is selected as optimal. This method achieves the best performance by systematically exploring possible values and is commonly used in offline RL for fine-tuning parameters through online interactions (Kalashnikov et al., 2018; Lee et al., 2022). However, it requires a few online interactions, making it less practical in fully offline settings.

To enable fully offline learning while retaining good performance, we further propose a *learning-based method*, as depicted in Fig. 4. This method involves training a predictor to estimate α based on

the relationship between BC and MB policies. Specifically, we first use the grid search method to get optimal α values as labels. The predictor takes a similarity vector derived from the centered kernel alignment (CKA) (Kornblith et al., 2019) between BC and MB policies and outputs the estimated α value. The predictor is trained using the data collected from one game where online interactions are feasible and can then be reused in similar games. Although the predictor provides only an approximate optimal parameter value, it eliminates the need for further online interactions once trained, offering a practical and efficient solution for fully offline settings.

Advantages of BOMB. First, by combining BC and MB methods, BOMB can work on datasets collected by any strategies, whether random or equilibrium strategies. This versatility ensures that BOMB can adapt to diverse offline data sources. Second, with the learned environment model for games, BOMB seamlessly integrates any online EF algorithm, enabling it to generalize across different equilibrium types. For example, we adapt PSRO (Lanctot et al., 2017) and Deep CFR (Brown et al., 2019) methods to compute NE, referred to as MB-PSRO and MB-CFR, respectively. Additionally, we adapt the JPSRO method (Marris et al., 2021) (MB-JPSRO) for computing (C)CE. This flexibility allows BOMB to address a wide range of equilibrium-finding tasks. Third, BOMB is game-agnostic, meaning it can learn game rules directly from offline datasets without requiring prior knowledge of the game itself. This characteristic is similar to the advantage of MuZero (Schrittwieser et al., 2020). This makes BOMB applicable to a broad array of games.

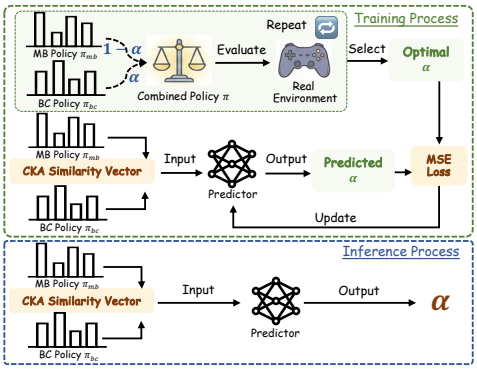


Figure 4: Learning-based estimation method.

4.2 Theoretical Analysis

In the offline RL area, dataset coverage over the optimal policy is typically sufficient for offline learning (Rashidinejad et al., 2021; Xie et al., 2021). However, in offline EF, the assumption that the equilibrium strategy generates the dataset is not sufficient. This insufficiency can be illustrated by the counterexample in Fig. 5. In this example, we can easily get the NE strategy, $\sigma^* = (\sigma_1^*, \sigma_2^*) = (\{I_1 : a_1\}, \{I_2 : b_2\})$. If σ^* is used to generate the offline dataset $\mathcal{D}$, $\mathcal{D}$ would only contain the single data point $((I_1^{t_1} = I_1, I_2^{t_1} = \emptyset, \emptyset, 1, \{a_1, a_2\}), a_1, (I_1^{t_2} = I_1 a_1, I_2^{t_2} = \emptyset, \emptyset, -1, \emptyset), (0, 0), 1)$. Clearly, the dataset $\mathcal{D}$ lacks any information about Player 2, making it insufficient for computing the NE strategy. Similarly, the assumption that the offline dataset covers the equilibrium strategy is also insufficient for the offline EF paradigm, as we prove in Appendix D. In this section, we outline the conditions for dataset coverage to guarantee the convergence of our methods in IIEFGs with perfect recall. To achieve this, we first introduce two key concepts of dataset coverage: uniform coverage and equilibrium coverage.

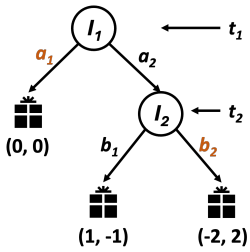


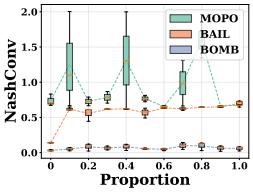
Figure 5: Game Tree.

Definition 4.1. An offline dataset $\mathcal{D}$ is said to be a **uniform coverage** of an IIEFG $\mathcal{G}$ if and only if it covers all possible state-action pairs in the game. Formally, it covers $(s_t, a_t, s_{t+1}, u_{t+1}, d_{t+1}), \forall s_t, a_t \in A(s_t)$ and $s_{t+1} \in T(s_t, a_t)$ where T is the transition function.

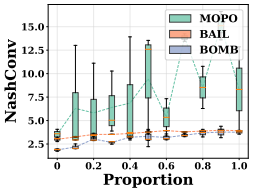
Definition 4.2. An offline dataset $\mathcal{D}$ is said to be an ϵ-**equilibrium coverage** of an IIEFG $\mathcal{G}$ if and only if its underlying behavior strategy $\sigma_{\mathcal{D}}$ satisfies: $\text{GAP}_{\mathcal{G}}(\sigma_{\mathcal{D}}) \leq \epsilon$, where $\sigma_{\mathcal{D}}$ is defined as: $\sigma_{\mathcal{D}}(s_t, a_t) = \frac{C(s_t, a_t)}{C(s_t)}$ and $\sigma_D(s_t, a_t) > 0$ for all s_t and $a_t \in A(s_t)$, with $C(s_t, a_t)$ and $C(s_t)$ denoting the counts of occurrences of state-action pair (s_t, a_t) and the state s_t in $\mathcal{D}$, respectively.

Building on these two dataset coverage definitions, we discuss the conditions required for our method to achieve convergence. Throughout, n is the number of players, $U = \max_{i \in N,\, z \in Z} |u_i(z)|$ is the largest absolute terminal utility, and H is the maximum number of player decision nodes on any root-to-leaf path. To facilitate analysis, we first introduce an assumption on the training error.

Assumption 4.3 (Training Error)**.** The trained BC policy σ_{BC} satisfies $\max_i \max_{I \in \mathcal{I}_i} \|\sigma_{BC,i}(I) - \sigma_{\mathcal{D},i}(I)\|_1 \leq \eta$, and the trained environment model, viewed as a game $\widehat{\mathcal{G}}$ whose state-action pairs and transitions are those recorded in $\mathcal{D}$, satisfies $\max_i \sup_\sigma |u_i^{\widehat{\mathcal{G}}}(\sigma) - u_i^{\mathcal{G}}(\sigma)| \leq \delta$.

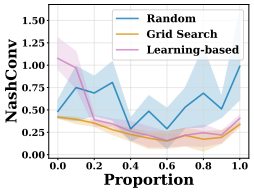


(a) Kuhn poker (2p)

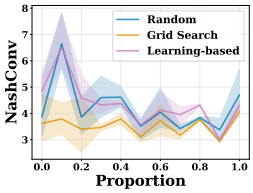


(b) Leduc poker (2p)

Figure 6: Comparison with offline RL.



(a) Kuhn poker (3p)



(b) Leduc poker (2p)

Figure 7: Results of different estimation methods.

We now bound the equilibrium gap of each component of BOMB, and of BOMB itself, in terms of the coverage of $\mathcal{D}$ and the training errors η and δ. Full proofs are given in Appendix D.

Theorem 4.4 (Model-based Method)**.** *Under Assumption 4.3, if $\mathcal{D}$ is a uniform coverage of $\mathcal{G}$ and σ_{MB} is an ϵ-equilibrium of a trained game model $\widehat{\mathcal{G}}$, then* $\text{GAP}_{\mathcal{G}}(\sigma_{MB}) \leq \epsilon + 2n\delta$.

Theorem 4.5 (Behavior Cloning)**.** *Under Assumption 4.3, if $\mathcal{D}$ is an ϵ-equilibrium coverage of $\mathcal{G}$ and σ_{BC} is the strategy returned by behavior cloning algorithm, then* $\text{GAP}_{\mathcal{G}}(\sigma_{BC}) \leq \epsilon + 2nUH\eta$.

Theorem 4.6. *Let $\sigma_i^{\alpha}(I) = \alpha\,\sigma_{BC,i}(I) + (1-\alpha)\,\sigma_{MB,i}(I)$ and let $\mathcal{A} \subset [0,1]$ be a finite grid containing 0 and 1. Under Assumption 4.3: (i) If $\alpha^\star$ minimizes* $\text{GAP}_{\mathcal{G}}(\sigma^{\alpha})$ *over $\mathcal{A}$, then* $\text{GAP}_{\mathcal{G}}(\sigma^{\alpha^\star}) \leq \min\{\text{GAP}_{\mathcal{G}}(\sigma_{BC}),\ \text{GAP}_{\mathcal{G}}(\sigma_{MB})\}$. *In particular,* $\text{GAP}_{\mathcal{G}}(\sigma_{MB}) \leq \text{GAP}_{\widehat{\mathcal{G}}}(\sigma_{MB}) + 2n\delta$ *whenever $\mathcal{D}$ is a uniform coverage of $\mathcal{G}$, and* $\text{GAP}_{\mathcal{G}}(\sigma_{BC}) \leq \text{GAP}_{\mathcal{G}}(\sigma_{\mathcal{D}}) + 2nUH\eta$ *whenever $\sigma_{\mathcal{D}}(s_t, a_t) > 0$ for all s_t and $a_t \in A(s_t)$. (ii) If $\widehat{\alpha}$ minimizes* $\text{GAP}_{\widehat{\mathcal{G}}}(\sigma^{\alpha})$ *over $\mathcal{A}$, which requires no interaction with $\mathcal{G}$, and $\mathcal{D}$ is a uniform coverage of $\mathcal{G}$, then* $\text{GAP}_{\mathcal{G}}(\sigma^{\widehat{\alpha}}) \leq \text{GAP}_{\mathcal{G}}(\sigma^{\alpha^\star}) + 4n\delta$, *with $\alpha^\star$ as in (i).*

5 Experimental Results

To evaluate the performance of BOMB, we conduct the following experiments: i) we compare BOMB with two offline RL algorithms; ii) we evaluate the performance of different methods for estimating the parameter α; and iii) we apply BOMB to various offline datasets to assess its effectiveness in computing different equilibrium strategies, including NE and CCE. We adopt OpenSpiel[1] (Lanctot et al., 2019) as the experimental platform, which provides a well-established collection of games and algorithms, thereby facilitating future replicability. Here, we consider several widely used games, including poker games, Liar's Dice, and Phantom Tic-Tac-Toe (Lisý et al., 2015; Brown et al., 2019). All experiments are conducted on a workstation equipped with a ten-core 3.3GHz Intel i9-9820X CPU and an NVIDIA RTX 2080Ti GPU. Unless otherwise specified, all reported results are averaged over three random seeds, and error bars are included. In some figures, the x-axis denotes the proportion of data from the random dataset. When this proportion is zero, the hybrid dataset reduces to the expert dataset, whereas when it is one, it becomes the random dataset. To facilitate analysis, we organize the experimental results around the following research questions.

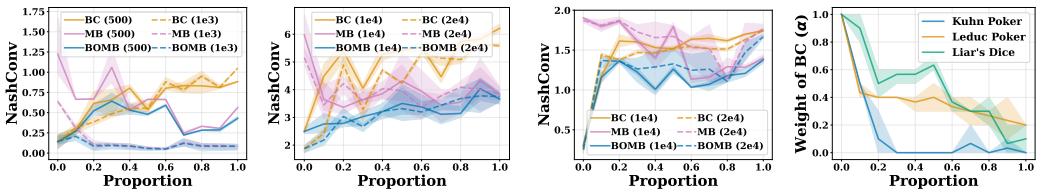
RQ1: Can BOMB outperform offline RL methods?

To support our claim that offline RL algorithms are insufficient for the offline EF paradigm, we consider two representative algorithms: the model-free algorithm–Best-Action Imitation Learning (BAIL) (Chen et al., 2020), and the model-based algorithm–Model-based Offline Policy Optimization (MOPO) (Yu et al., 2020). The comparison results, shown in Fig. 6, are obtained on two-player Kuhn and Leduc games under hybrid datasets. The results show that BOMB consistently outperforms both offline RL algorithms across all settings, regardless of the dataset composition. These findings demonstrate the limitations of traditional offline RL algorithms for offline EF and further highlight the effectiveness of BOMB in computing equilibrium strategies offline.

RQ2: How do different methods for estimating the parameter α perform?

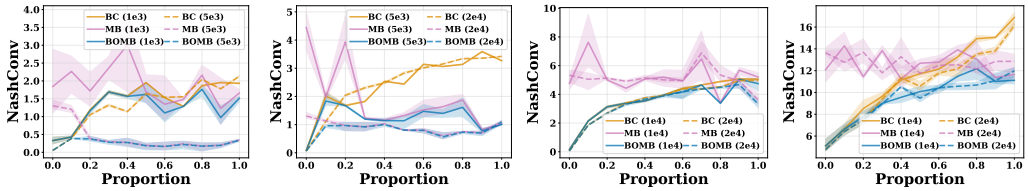
We evaluate the three proposed parameter estimation methods through experiments on poker games. For the learning-based method, the parameter predictor is trained on the two-player Kuhn poker game and then tested on other poker games. The results are presented in Fig. 7, which show that the grid search method consistently performs best among all cases. The learning-based method performs

[1] https://github.com/deepmind/open_spiel



(a) Kuhn poker (2p) (b) Leduc poker (2p) (c) Liar's dice (d) Proper weight

Figure 8: Experimental results on computing NE in two-player games.



(a) Kuhn poker (3p) (b) Kuhn poker (4p) (c) Kuhn poker (5p) (d) Leduc poker (3p)

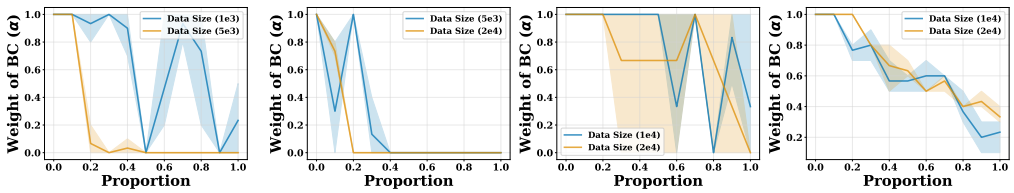
Figure 9: Experimental results on computing NE in multi-player games.

comparably to the grid search method on three-player Kuhn poker, but is slightly worse on the two-player Leduc game, suggesting that its effectiveness depends on the similarity between the test game and the game used for training. Interestingly, the random method also performs surprisingly well in some cases, indicating that even simple combinations can be effective. We adopt the grid search method as the default parameter estimation method in the remaining experiments.

RQ3: Can the BOMB framework compute NE?

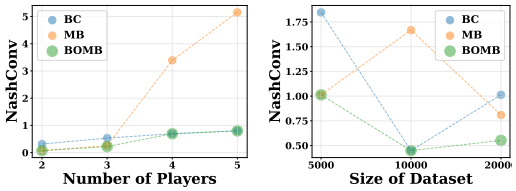
Two-Player Cases. We first evaluate the performance of BOMB in computing NE for two-player games. In addition to BOMB, we also examine the performance of its two individual components: the behavior cloning (BC) method and the model-based (MB) method. This design helps reveal the respective strengths and limitations of each component. For the model-based method component, we use either MB-CFR or MB-PSRO, and the MB framework's performance is shown to be independent of the specific algorithm used (details in Appendix F). Figs. 8(a)-8(c) report the results on two-player games under offline datasets of different sizes. As the proportion of the random dataset increases, the performance of BC declines, whereas the performance of MB improves slightly. We also observe that increasing the size of the offline dataset leads to only limited improvement for BC, but yields more noticeable gains for MB. These results suggest that the performance of BC mainly depends on the quality of the behavior policy underlying the dataset, while the performance of MB relies on how accurately the learned game model matches the true game. Across all cases, BOMB consistently outperforms both BC and MB, demonstrating its effectiveness in computing NE for two-player games. We also plot the parameter α under different cases, as shown in Fig. 8(d). The results reveal that, as the proportion of the random dataset increases, the weight assigned to the BC policy decreases. This indicates that BC is more effective on expert datasets, whereas MB performs better on random datasets, demonstrating the complementary strengths of BC and MB.

Multi-Player Cases. We conduct experiments on several multi-player games to evaluate the performance of BOMB in multi-player cases, as shown in Fig. 9. Similar to the results in the two-player setting, BOMB consistently performs as well as or better than both the BC and MB methods. However, as the proportion of the random dataset in the hybrid dataset increases, the performance of BC declines, while the performance of MB becomes unstable and shows a slight downward trend. Notably, computing NEs in multi-player games is PPAD-hard. Furthermore, CFR-based algorithms are only guaranteed to converge to coarse correlated equilibria in multi-player games, not to NE. Consequently, the weak performance of MB may arise either from inaccuracies in the learned game model or from the intrinsic difficulty of NE computation in multi-player games. These observations highlight critical challenges for offline EF in multi-player games, particularly the need for more effective equilibrium-finding algorithms and training a more accurate environment game model. Addressing these challenges is essential for advancing offline EF in multi-player games. Fig. 10 shows the value of parameter α across different hybrid datasets. In three-player Kuhn poker, the value quickly drops to zero as the proportion of the random dataset increases, indicating that MB generally outperforms BC except when the random dataset proportion is low. In contrast, in three-player Leduc poker, the BC policy retains a relatively high weight in most cases, except when the proportion of the



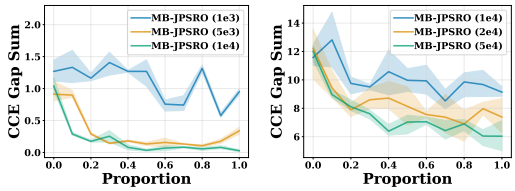
(a) Kuhn poker (3p) (b) Kuhn poker (4p) (c) Kuhn poker (5p) (d) Leduc poker (3p)

Figure 10: Proper weight of BC policy in multi-player games.



(a) Kuhn poker (b) Phantom Tic-Tac-Toe

Figure 11: Results on the learning dataset.



(a) Kuhn poker (3p) (b) Leduc poker (3p)

Figure 12: Results on computing CCE.

random dataset becomes large. This suggests that the MB method struggles in these games due to the difficulty of learning an approximate equilibrium strategy. More broadly, this difficulty reflects the inherent complexity of multi-player games, where both the development of equilibrium-finding algorithms and the training of accurate game models remain challenging.

Simulating Real-World Cases. We further conduct experiments on the learning datasets, which closely approximate real-world conditions. Fig. 11(a) presents the results of Kuhn poker with different numbers of players, and Fig. 11(b) shows the results of Phantom Tic-Tac-Toe under datasets of different sizes. These results indicate that, even when the offline dataset is generated by an unknown strategy, BOMB still consistently outperforms both BC and MB in approximating the NE strategy.

RQ4: Can the BOMB framework compute CCE?

We evaluate the performance of our model-based method in computing the CCE strategies. The BC technique and the BOMB framework are not considered for CCE computation, because the offline dataset is collected from independent strategies for each player rather than a joint strategy. The results are shown in Fig. 12. As the size of the offline dataset increases, the performance of MB-JPSRO improves accordingly. This result suggests that the effectiveness of the model-based method depends strongly on the quality of the learned environment model, and further highlights the importance of accurate environment modeling for robust offline equilibrium computation.

6 Conclusion

We investigated the paradigm of offline equilibrium finding, Offline EF, in extensive-form games, with the goal of computing equilibrium strategies from offline datasets. To address the lack of comprehensive benchmark datasets, we first construct offline EF datasets using established data collection methods. We then propose a novel algorithm, **BOMB**, which combines behavior cloning with a model-based approach. To better understand offline EF and the proposed framework, we conducted both theoretical and empirical analyses, offering performance guarantees for BOMB. Finally, extensive experimental results demonstrate that BOMB consistently outperforms existing offline RL algorithms, confirming its effectiveness in computing equilibrium strategies in the offline setting. Although this paper represents an important step toward offline learning in extensive-form games, several limitations remain. First, the games considered in this paper are relatively small-scale. Extending our study to large-scale games, such as Texas Hold'em poker (Brown & Sandholm, 2018) and football games (Liu et al., 2022), is an important future direction. Second, this paper primarily focuses on NE and CCE. In the future, we plan to consider more solution concepts, such as quantal response equilibrium (QRE) (McKelvey & Palfrey, 1995). This will allow us to examine the generalizability of BOMB to new solution concepts. Third, the relationship between dataset properties and offline EF algorithms deserves further investigation. In particular, rather than relying only on datasets collected by researchers, future work may explore the application of offline EF to human-generated datasets toward real-world deployment (Wang et al., 2024).

AI USE STATEMENT

In this work, we use generative AI tools only for polishing the writing of the paper, i.e., improving the presentation of the paper. We did not use generative AI to generate synthetic datasets (all datasets were generated by the collection strategies described in the main paper), to implement methods or write code, to propose or refine hypotheses, to design experiments, or to interpret experimental results; the remaining tasks requiring disclosure (translation, qualitative or thematic data analysis) do not apply to this work. We have reviewed all AI-assisted content. All AI-edited text was reviewed by the authors for factual accuracy. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies equilibrium computation in extensive-form games from offline data. All datasets used in the paper are synthetic: they are generated by collection strategies (uniform random play, intermediate iterates of equilibrium-finding algorithms, and equilibrium strategies) in games implemented in OpenSpiel (Lanctot et al., 2019). No human subjects, human-generated data, or personal information are involved, and the released datasets raise no privacy concerns. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

We have taken the following steps to make our results reproducible. The source code of BOMB, including MB-PSRO, MB-CFR, and MB-JPSRO, the three parameter-estimation methods, and the scripts that generate all datasets, is submitted as supplementary material. The datasets themselves (about 11 GB) will be released upon acceptance. For the theoretical results, all assumptions and complete proofs are provided in Appendix.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

Maruan Al-Shedivat, Trapit Bansal, Yuri Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. *arXiv preprint arXiv:1710.03641*, 2017.

Robert J Aumann. Correlated equilibrium as an expression of Bayesian rationality. *Econometrica: Journal of the Econometric Society*, pp. 1–18, 1987.

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

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

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *ICML*, pp. 793–802, 2019.

Xinyue Chen, Zijian Zhou, Zheng Wang, Che Wang, Yanqiu Wu, and Keith Ross. Bail: Best-action imitation learning for batch deep reinforcement learning. In *NeurIPS*, pp. 18353–18363, 2020.

Qiwen Cui and Simon S. Du. When is offline two-player zero-sum markov game solvable? *arXiv preprint arXiv:2201.03522*, 2022.

Jakob N Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*, 2017.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto and Shixiang Gu. A minimalist approach to offline reinforcement learning. In *NeurIPS*, pp. 20132–20145, 2021.

Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708*, 2019.

Richard Gibson, Marc Lanctot, Neil Burch, Duane Szafron, and Michael Bowling. Generalized sampling and variance in counterfactual regret minimization. In *AAAI*, pp. 1355–1361, 2012.

Aditya Grover, Maruan Al-Shedivat, Jayesh Gupta, Yuri Burda, and Harrison Edwards. Learning policy representations in multiagent systems. In *ICML*, pp. 1802–1811, 2018.

Caglar Gulcehre, Ziyu Wang, Alexander Novikov, Thomas Paine, Sergio Gómez, Konrad Zolna, Rishabh Agarwal, Josh S Merel, Daniel J Mankowitz, Cosmin Paduraru, et al. Rl unplugged: A suite of benchmarks for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:7248–7259, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, pp. 1861–1870, 2018.

He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. Opponent modeling in deep reinforcement learning. In *ICML*, pp. 1804–1813, 2016.

Zhang-Wei Hong, Shih-Yang Su, Tzu-Yun Shann, Yi-Hsiang Chang, and Chun-Yi Lee. A deep policy inference q-network for multi-agent systems. *arXiv preprint arXiv:1712.07893*, 2017.

Patrick R Jordan, L Julian Schvartzman, and Michael P Wellman. Strategy exploration in empirical games. In *AAMAS*, pp. 1131–1138, 2010.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, pp. 651–673. PMLR, 2018.

Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity*, 2(1):1–22, 2019.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In *NeurIPS*, pp. 21810–21823, 2020.

Dong Ki Kim, Miao Liu, Matthew D Riemer, Chuangchuang Sun, Marwa Abdulhai, Golnaz Habibi, Sebastian Lopez-Cot, Gerald Tesauro, and Jonathan How. A policy gradient algorithm for learning to learn in multiagent reinforcement learning. In *ICML*, pp. 5541–5550, 2021.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, pp. 4909–4926, 2022.

Usha Kiruthika, Thamarai Selvi Somasundaram, and S Kanaga Suba Raja. Lifecycle model of a negotiation agent: A survey of automated negotiation techniques. *Group Decision and Negotiation*, 29:1239–1262, 2020.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, pp. 3519–3529, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *NeurIPS*, pp. 1179–1191, 2020.

Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte Carlo sampling for regret minimization in extensive games. In *NeurIPS*, pp. 1078–1086, 2009.

Marc Lanctot, Vinicius Zambaldi, Audrūnas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *NeurIPS*, pp. 4193–4206, 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.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*, pp. 1702–1712. PMLR, 2022.

David A Levin and Yuval Peres. *Markov chains and mixing times*. American Mathematical Society, second edition, 2017.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Hui Li, Kailiang Hu, Shaohua Zhang, Yuan Qi, and Le Song. Double neural counterfactual regret minimization. In *ICLR*, 2019.

Pengdeng Li, Jakub Černỳ, Shuxin Li, Youzhi Zhang, Xinrun Wang, Stephen McAleer, Hau Chan, and Bo An. Grasper: A generalist pursuer for pursuit-evasion problems. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, volume 2024, pp. 1147–1155, 2024.

Shuxin Li, Xinrun Wang, Youzhi Zhang, Wanqi Xue, Jakub Černỳ, and Bo An. Solving large-scale pursuit-evasion games using pre-trained strategies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11586–11594, 2023.

Viliam Lisý, Marc Lanctot, and Michael Bowling. Online Monte Carlo counterfactual regret minimization for search in imperfect information games. In *AAMAS*, pp. 27–36, 2015.

Siqi Liu, Kay Choong See, Kee Yuan Ngiam, Leo Anthony Celi, Xingzhi Sun, Mengling Feng, et al. Reinforcement learning for clinical decision support in critical care: Comprehensive review. *Journal of Medical Internet Research*, 22(7):e18477, 2020.

Siqi Liu, Guy Lever, Zhe Wang, Josh Merel, SM Ali Eslami, Daniel Hennes, Wojciech M Czarnecki, Yuval Tassa, Shayegan Omidshafiei, Abbas Abdolmaleki, et al. From motor control to team play in simulated humanoid football. *Science Robotics*, 7(69):eabo0235, 2022.

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*, pp. 7480–7491. PMLR, 2021.

Michael Mathieu, Sherjil Ozair, Srivatsan Srinivasan, Caglar Gulcehre, Shangtong Zhang, Ray Jiang, Tom Le Paine, Konrad Zolna, Richard Powell, Julian Schrittwieser, et al. StarCraft II unplugged: Large scale offline reinforcement learning. In *Deep RL Workshop NeurIPS 2021*, 2021.

Tatsuya Matsushima, Hiroki Furuta, Yutaka Matsuo, Ofir Nachum, and Shixiang Gu. Deployment-efficient reinforcement learning via model-based offline optimization. *arXiv preprint arXiv:2006.03647*, 2020.

Stephen McAleer, John Lanier, Roy Fox, and Pierre Baldi. Pipeline psro: A scalable approach for finding approximate nash equilibria in large games. In *NeurIPS*, pp. 20238–20248, 2020.

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.

Stephen Marcus McAleer, John Banister Lanier, Kevin Wang, Pierre Baldi, and Roy Fox. Xdo: A double oracle algorithm for extensive-form games. In *NeurIPS*, pp. 23128–23139, 2021.

Richard D McKelvey and Thomas R Palfrey. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38, 1995.

H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *ICML*, pp. 536–543, 2003.

John F Nash. Equilibrium points in n-person games. *PNAS*, 36(1):48–49, 1950.

Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

Shayegan Omidshafiei, Christos Papadimitriou, Georgios Piliouras, Karl Tuyls, Mark Rowland, Jean-Baptiste Lespiau, Wojciech M Czarnecki, Marc Lanctot, Julien Perolat, and Remi Munos. α-rank: Multi-agent evaluation by evolution. *Scientific Reports*, 9(1):1–29, 2019.

Julien Perolat, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T Connor, Neil Burch, Thomas Anthony, et al. Mastering the game of stratego with model-free multiagent reinforcement learning. *Science*, 378(6623):990–996, 2022.

Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *arXiv preprint arXiv:2203.01387*, 2022.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.

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 *AAAI*, pp. 2157–2164, 2019.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

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

L Julian Schvartzman and Michael P Wellman. Stronger cda strategies through empirical game-theoretic analysis and reinforcement learning. In *AAMAS*, pp. 249–256, 2009b.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008.

Noah Y Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, and Martin Riedmiller. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *arXiv preprint arXiv:2002.08396*, 2020.

Bharat Singh, Rajesh Kumar, and Vinay Pratap Singh. Reinforcement learning in robotic applications: A comprehensive survey. *Artificial Intelligence Review*, pp. 1–46, 2021.

Adish Singla, Anna N Rafferty, Goran Radanovic, and Neil T Heffernan. Reinforcement learning for education: Opportunities and challenges. *arXiv preprint arXiv:2107.08828*, 2021.

Eric Steinberger. Single deep counterfactual regret minimization. *arXiv preprint arXiv:1901.07621*, 2019.

Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2): 146–160, 1972.

Zhe Wang, Petar Veličković, Daniel Hennes, Nenad Tomašev, Laurel Prince, Michael Kaisers, Yoram Bachrach, Romuald Elie, Li Kevin Wenliang, Federico Piccinini, et al. TacticAI: an AI assistant for football tactics. *Nature communications*, 15(1):1906, 2024.

Kevin Waugh, David Schnizlein, Michael H Bowling, and Duane Szafron. Abstraction pathologies in extensive games. In *AAMAS*, pp. 781–788, 2009.

Michael P Wellman. Methods for empirical game-theoretic analysis. In *AAAI*, pp. 1552–1556, 2006.

Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34:6683–6694, 2021.

Wanqi Xue, Youzhi Zhang, Shuxin Li, Xinrun Wang, Bo An, and Chai Kiat Yeo. Solving large-scale extensive-form network security games via neural fictitious self-play. In *IJCAI*, pp. 3713–3720, 2021.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *NeurIPS*, pp. 14129–14142, 2020.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in Neural Information Processing Systems*, 34, 2021a.

Xiaopeng Yu, Jiechuan Jiang, Haobin Jiang, and Zongqing Lu. Model-based opponent modeling. *arXiv preprint arXiv:2108.01843*, 2021b.

Daochen Zha, Jingru Xie, Wenye Ma, Sheng Zhang, Xiangru Lian, Xia Hu, and Ji Liu. Douzero: Mastering doudizhu with self-play deep reinforcement learning. In *ICML*, pp. 12333–12344, 2021.

Yan Zheng, Zhaopeng Meng, Jianye Hao, Zongzhang Zhang, Tianpei Yang, and Changjie Fan. A deep bayesian policy reuse approach against non-stationary agents. In *NeurIPS*, pp. 962–972, 2018.

Han Zhong, Wei Xiong, Jiyuan Tan, Liwei Wang, Tong Zhang, Zhaoran Wang, and Zhuoran Yang. Pessimistic minimax value iteration: Provably efficient equilibrium learning from offline datasets. In *ICML*, pp. 27117–27142, 2022.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *NeurIPS*, pp. 1729–1736, 2007.

A FREQUENTLY ASKED QUESTIONS (FAQS)

Q1: What are the potential impacts of this work?

This work addresses a critical gap in offline learning within the domain of game theory. By introducing the offline learning framework, we demonstrate its potential to advance the application of game-theoretic methods to real-world problems and to stimulate new research directions in equilibrium computation. Notably, equilibrium strategies offer greater robustness than purely optimal strategies, particularly in security-sensitive and adversarial contexts. Consequently, the offline EF paradigm provides a valuable foundation for deriving robust strategies capable of effectively managing competitive and adversarial real-world challenges.

Q2: Why is offline EF important and needed?

Offline EF algorithms designed for adversarial environments are crucial in strictly competitive games, such as security games. This setting fundamentally differs from offline multi-agent reinforcement learning, which often emphasizes cooperation among agents rather than strict competition. Consider, for example, pursuit-evasion games, where the pursuer (defender) attempts to capture the evader (attacker). In such contexts, it is unrealistic to assume that the attacker follows a fixed strategy, since attackers are strategic and capable of adapting their behavior. If one were to apply a standard offline RL algorithm to learn the defender's optimal strategy solely from historical data, the resulting strategy could suffer significant utility loss, as it may be highly exploitable. Specifically, the attacker could deviate from past behaviors and instead adopt a best response to the defender's learned strategy. By contrast, computing a Nash Equilibrium (NE) provides a more robust solution, since equilibrium strategies are, by definition, non-exploitable.

Q3: Is offline EF more difficult than offline RL or offline cooperative MARL?

Traditional offline RL focuses on learning an optimal strategy for a single agent, to maximize utility in a dynamic environment modeled as a Markov Decision Process (MDP). In this setting, the environment dynamics are independent of the actions of other agents. Offline cooperative multi-agent RL (MARL) extends this paradigm to multiple agents, but typically assumes that the agents work together toward a shared objective. Although the dynamics depend on the joint actions of all agents, the cooperative setting aligns their incentives, making it possible to optimize for collective performance without the concern of strategic exploitation.

By contrast, offline EF arises in strictly competitive games, where each player faces a strategic opponent. The effective dynamics for one player are determined jointly by the game structure and the opponent's strategy, and any change in the opponent's behavior alters the induced MDP. This makes the problem significantly more challenging than both offline RL and cooperative MARL, since a strategy learned from historical data may be highly exploitable if the opponent adapts. The offline EF framework addresses this difficulty by enabling the computation of Nash equilibrium (NE) strategies, which provide robustness against fully strategic and adversarial opponents.

Q4: What are the differences between offline EF and Empirical Game-Theoretic Analysis?

1) As described in (Wellman, 2006), Empirical Game-Theoretic Analysis (EGTA) requires access to a game simulator and conducts strategic reasoning through an iterative process of simulation and game-theoretic analysis. In this sense, the availability of a simulator is essential for EGTA. By contrast, the offline EF paradigm does not rely on a simulator. Instead, it operates under the offline learning setting, where only a fixed dataset of past interactions is available.

2) In EGTA, the empirical game is constructed from simulation outcomes obtained by executing a set of known strategies on the simulator. In contrast, in the offline EF paradigm, the offline dataset is generated using unknown strategies. While we use various behavior strategies to create multiple offline datasets, these strategies are not utilized directly within the offline EF paradigm. These distinctions highlight the fundamental differences between EGTA and offline EF.

Q5: What are the novelties of the proposed offline EF algorithm – BOMB?

To the best of our knowledge, we are the first to propose an empirical algorithm for computing equilibrium strategies from offline datasets, i.e., the offline EF paradigm. Unlike traditional offline RL algorithms, which are typically categorized as either model-based or model-free, our algorithm combines the advantages of both to efficiently compute equilibrium strategies in the offline setting. The

proposed BOMB framework integrates behavior cloning (BC) with a model-based (MB) component, supported by novel parameter estimation techniques. By incorporating an environment model into the MB component, we enable the adaptation of existing online equilibrium-finding algorithms for use in offline settings. Furthermore, we design multiple methods for selecting the combination parameter, providing flexibility to accommodate different scenarios depending on whether limited online interaction is available. Experimental results show that neither the BC nor the MB approach consistently achieves strong performance in isolation, whereas BOMB outperforms both across all tested cases. These results demonstrate that the BOMB framework successfully unifies the strengths of BC and MB, yielding a highly effective solution for offline equilibrium computation.

Q6: What are the limitations and future works for this paper?

This work has several limitations that we aim to address in the future. First, the games considered in this study are relatively small-scale. Expanding to large-scale games, such as Texas Hold'em poker (Brown & Sandholm, 2018) and football games (Liu et al., 2022), will be a focus of future work. Second, this work primarily targets NE and CCE as solution conceptions. In future research, we plan to incorporate additional solution concepts, such as quantal response equilibrium (QRE) (McKelvey & Palfrey, 1995) and α-rank (Omidshafiei et al., 2019). Investigating the generalizability of both the datasets and the BOMB framework to these novel solution concepts will further broaden the scope of offline EF. Third, the relationships between the datasets and offline EF algorithms require further exploration. Instead of relying on datasets curated by researchers, we aim to apply offline EF to human-play datasets, moving toward real-world deployment (Wang et al., 2024). This will help bridge the gap between theoretical research and practical applications.

B Related Work

B.1 Equilibrium Finding Algorithms.

Contemporary state-of-the-art algorithms for solving IIEFGs can be broadly categorized into two groups: no-regret methods derived from CFR (Zinkevich et al., 2007), and incremental strategy-space generation methods of the PSRO framework (Lanctot et al., 2017).

For the first group, CFR is a family of iterative methods for approximately solving IIEFGs. Let σ_i^t be the strategy used by player i in iteration t. We use $u_i(\sigma, h)$ to define the expected utility of player i given that the history h is reached and all players act according to strategy σ from that point on. Accordingly, $u_i(\sigma, h \cdot a)$ is used to define the expected utility of player i given that the history h is reached and all players play according to strategy σ except player i selects action a in history h. Formally, $u_i(\sigma, h) = \sum_{z \in Z} \pi^\sigma(h, z) u_i(z)$ and $u_i(\sigma, h \cdot a) = \sum_{z \in Z} \pi^\sigma(h \cdot a, z) u_i(z)$. The *counterfactual value* of the information set I, $v_i^\sigma(I)$, is the expected value of information set I given that player i attempts to reach it. This value is the weighted average of the expected utility of each history in the information set. The weight is proportional to the contribution of all players except player i to reach each history. Thus, $v_i^\sigma(I) = \sum_{h \in I} \pi_{-i}^\sigma(h) u_i(\sigma, h)$. For any action $a \in A(I)$, the counterfactual value of action a is $v_i^\sigma(I, a) = \sum_{h \in I} \pi_{-i}^\sigma(h) u_i(\sigma, h \cdot a)$. The *instantaneous counterfactual regret* for an action a in information set I during iteration t is $r^t(I, a) = v_{P(I)}^{\sigma^t}(I, a) - v_{P(I)}^{\sigma^t}(I)$. Therefore, the counterfactual regret for an action a in information set I on iteration T is $R^T(I, a) = \sum_{t=1}^T r^t(I, a)$. In vanilla CFR, players use *Regret Matching* to pick a distribution over available actions in an information set proportional to the cumulative regret of those actions. Formally, in iteration $T + 1$, player i selects action $a \in A(I)$ according to probabilities

$$\sigma^{T+1}(I, a) = \begin{cases} \frac{R_+^T(I,a)}{\sum_{b \in A(I)} R_+^T(I,b)} & \text{if } \sum_{b \in A(I)} R_+^T(I, b) > 0, \\ \frac{1}{|A(I)|} & \text{otherwise,} \end{cases}$$

where $R_+^T(I, a) = \max\{R^T(I, a), 0\}$ is the positive portion of the regret value, since we are often most concerned about the cumulative regret when it is positive. If a player acts according to regret matching in the information set I on every iteration, then in iteration T, $R^T(I) \leq \Delta_i \sqrt{|A_i|}\sqrt{T}$ where $\Delta_i = \max_z u_i(z) - \min_z u_i(z)$ is the range of utilities of player i. Moreover, $R_i^T \leq \sum_{I \in \mathcal{I}_i} R^T(I) \leq |\mathcal{I}_i| \Delta_i \sqrt{|A_i|}\sqrt{T}$. Therefore, $\lim_{T \to \infty} \frac{R_i^T}{T} = 0$. In two-player zero-sum games,

if both players' average regret $\frac{R_i^T}{T} \leq \epsilon$, their average strategies $(\overline{\sigma}_1^T, \overline{\sigma}_2^T)$ over all iterations form a 2ϵ-equilibrium (Waugh et al., 2009). Some CFR-based variants are proposed to solve large-scale imperfect-information extensive-form games. Some sampling-based CFR (Lanctot et al., 2009; Gibson et al., 2012; Schmid et al., 2019) are proposed to effectively solve large-scale games by traversing a subset of the game tree instead of the whole game tree. With the development of deep learning techniques, neural network function approximation can be applied to the CFR algorithm. Deep CFR (Brown et al., 2019), Single Deep CFR (Steinberger, 2019), and Double Neural CFR (Li et al., 2019) are algorithms using deep neural networks to replace the tabular representation.

For the second group, PSRO is a general framework that scales Double Oracle (DO) (McMahan et al., 2003) to large extensive-form games via using RL to compute the best response approximately. To make PSRO more effective in solving large-scale games, Pipeline PSRO (P2SRO) (McAleer et al., 2020) is proposed by parallelizing PSRO with convergence guarantees. Extensive-Form Double Oracle (XDO) (McAleer et al., 2021) is a version of PSRO where the restricted game allows mixing population strategies not only at the root of the game but every information set. It can guarantee to converge to an approximate NE in several iterations that are linear in the number of information sets, while PSRO may require a number of iterations exponential in the number of information sets. Neural XDO (NXDO), as a neural version of XDO, learns approximate best response strategies through any deep RL algorithm. Recently, Anytime Double Oracle (ADO) (McAleer et al., 2022), a tabular double oracle algorithm for two-player zero-sum games, was proposed to converge to an NE while decreasing exploitability from one iteration to the next. Anytime PSRO (APSRO), as a version of ADO, calculates best responses via RL algorithms. Except for NEs, we also consider (Coarse) Correlated equilibrium ((C)CE). Joint Policy Space Response Oracles (JPSRO) (Marris et al., 2021) is proposed for training agents in n-player, general-sum extensive-form games, which provably converges to (C)CEs. The excellent performance of these equilibrium-finding algorithms depends on the interactions with the actual game environment or a precise simulator. Therefore, these algorithms cannot directly be applied to the offline EF paradigm.

B.2 Opponent Modeling.

Opponent modeling algorithm is necessary for multi-agent settings where secondary agents with competing goals also adapt their strategies, yet it remains challenging because policies interact with each other and change (He et al., 2016). One simple idea of opponent modeling is to build a model each time a new opponent or group of opponents is encountered (Zheng et al., 2018). However, it is infeasible to learn a model every time. A better approach is to represent an opponent's policy with an embedding vector. (Grover et al., 2018) uses a neural network as an encoder, taking the trajectory of one agent as input. Imitation learning and contrastive learning are also used to train the encoder. Then, the learned encoder can be combined with reinforcement learning algorithms by feeding the generated representation into the policy and/or value network. DRON (He et al., 2016) and DPIQN (Hong et al., 2017) are two algorithms based on DQN, which use a secondary network that takes observations as input and predicts opponents' actions. However, if the opponents can also learn, these methods become unstable. Therefore, it is necessary to take the learning process of opponents into account. (Foerster et al., 2017) propose a method named Learning with Opponent-Learning Awareness (LOLA), in which each agent shapes the anticipated learning of the other agents in the environment. Further, the opponents may still be learning continuously during execution. Therefore, (Al-Shedivat et al., 2017) proposes a method based on a meta-policy gradient named Meta-MPG. It uses trajectories from current opponents to perform multiple meta-gradient steps and constructs a policy that favors updating the opponents. Meta-MAPG (Kim et al., 2021) extends Meta-MPG by including an additional term that accounts for the impact of the agent's current policy on the future policies of opponents, similar to LOLA. (Yu et al., 2021b) proposes model-based opponent modeling (MBOM), which employs the environment model to adapt to various opponents. In our offline EF paradigm, our goal is to compute the equilibrium strategy based on the offline dataset. Applying opponent modeling is not enough for the offline EF paradigm since it only aims at computing the best response strategy instead of the equilibrium strategy.

B.3 Empirical Game Theoretic Analysis.

Empirical game theoretic analysis (EGTA) is an empirical methodology that bridges the gap between game theory and simulation for practical strategic reasoning (Wellman, 2006). In EGTA, game mod-

els are iteratively extended through a process of generating new strategies based on learning from experience with prior strategies. The strategy exploration problem (Jordan et al., 2010) that how to efficiently assemble an efficient portfolio of policies for EGTA, is the most challenging problem. (Schvartzman & Wellman, 2009b) deploy tabular RL as a best-response oracle in EGTA for strategy generation. They also build the general problem of strategy exploration in EGTA and investigate whether better options exist beyond best-responding to an equilibrium (Schvartzman & Wellman, 2009a). Investigation of strategy exploration was advanced significantly by the introduction of the Policy Space Response Oracle (PSRO) framework (Lanctot et al., 2017), which is a flexible framework for iterative EGTA, where at each iteration, new strategies are generated through reinforcement learning. Note that when employing NE as the meta-strategy solver, PSRO reduces to the double oracle (DO) algorithm (McMahan et al., 2003). In EGTA, a space of strategies is examined through simulation, which means that it needs a simulator, and the policies are known in advance. However, in the offline EF paradigm, only an offline dataset is provided. Therefore, techniques in EGTA cannot be directly applied to the offline EF paradigm.

B.4 Offline Reinforcement Learning.

Offline reinforcement learning (offline RL) is a *data-driven* paradigm that learns exclusively from static datasets of previously collected interactions, making it feasible to extract policies from large and diverse training datasets (Levine et al., 2020). This paradigm can be extremely valuable in settings where online interaction is impractical, either because data collection is expensive or dangerous (e.g., in robotics (Singh et al., 2021), education (Singla et al., 2021), healthcare (Liu et al., 2020), and autonomous driving (Kiran et al., 2022)). Therefore, efficient offline RL algorithms have a much broader range of applications than online RL and are particularly appealing for real-world applications (Prudencio et al., 2022). Due to its attractive characteristics, there have been a lot of recent studies. Here, we can divide the research of offline RL into two categories: model-based algorithms and model-free algorithms.

Model-free offline RL algorithms learn a good policy directly from the offline dataset. To do this, there are two types of algorithms: actor-critic and imitation learning methods. These actor-critic algorithms focus on implementing policy regularization and value regularization based on existing reinforcement learning algorithms. (Haarnoja et al., 2018) propose soft actor-critic (SAC) by adding an entropy regularization term to the policy gradient objective. This work mainly focuses on policy regularization. For the research on value regularization, an offline RL method named Conservative Q-Learning (CQL) (Kumar et al., 2020) learns a lower bound of the true Q-function by adding value regularization terms to its objective. Another line of model-free offline RL research is imitation learning, which mimics the behavior policy based on the offline dataset. (Chen et al., 2020) propose a method named Best-Action Imitation Learning (BAIL), which fits a value function, then uses it to select the best actions. Meanwhile, (Siegel et al., 2020) proposes a method that learns an Advantage-weighted Behavior Model (ABM) and uses it as a prior in performing Maximum a posteriori Policy Optimization (MPO) (Abdolmaleki et al., 2018). It consists of multiple iterations of policy evaluation and prior learning until they finally perform a policy improvement step using their learned before extracting the best possible policy.

Model-based algorithms rely on the offline dataset to learn a dynamics model or a trajectory distribution used for planning. The trajectory distribution induced by models is used to determine the best set of actions to take at each given time step. (Kidambi et al., 2020) propose a method named Model-based Offline Reinforcement Learning (MOReL), which measures their model's epistemic uncertainty through an ensemble of dynamics models. Meanwhile, (Yu et al., 2020) proposes another method named Model-based Offline Policy Optimization (MOPO), which uses the maximum prediction uncertainty from an ensemble of models. Concurrently, (Matsushima et al., 2020) proposes the BehaviorREgularized Model-ENsemble (BREMEN) method, which learns an ensemble of models of the behavior MDP, as opposed to a pessimistic MDP. In addition, it implicitly constrains the policy to be close to the behavior policy through trust-region policy updates. More recently, (Yu et al., 2021a) proposed a method named Conservative Offline Model-Based Policy Optimization (COMBO), a model-based version of CQL. The main advantage of COMBO concerning MOReL and MOPO is that it removes the need for uncertainty quantification in model-based offline RL approaches, which is challenging and often unreliable. However, the above offline RL algorithms cannot be directly applied to the offline EF paradigm, and experimental results also empirically verify this claim.

C DATASETS

C.1 VISUALIZATION

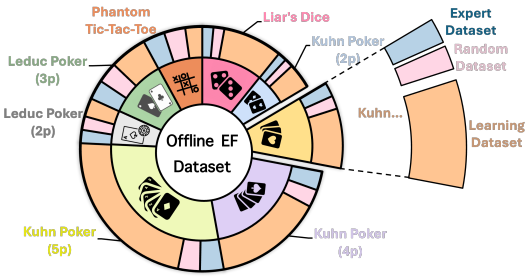
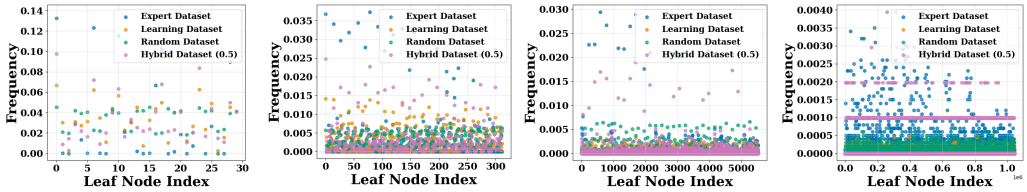


Figure 13: Visualization of the offline EF dataset

Fig. 13 provides a comprehensive view of our offline EF dataset, which includes data collected for eight games: two-player Kuhn poker, three-player Kuhn poker, four-player Kuhn poker, five-player Kuhn poker, two-player Leduc poker, three-player Leduc poker, Phantom Tic-Tac-Toe, and Liar's Dice. For each game, we generated three types of datasets: a random dataset, an expert dataset, and a learning dataset, following our data collection methods. To validate the diversity of these collected offline datasets and gain insights into them, we developed a visualization method for comparison. First, we generate the game tree for each game. Subsequently, we traverse the game tree using depth-first search (DFS) (Tarjan, 1972) and assign an index to each leaf node based on the DFS results. Then, we count the frequency of each leaf node within the dataset. The reason why we do this is that each leaf node represents a unique sampled trajectory originating from the root node of the game tree. As a result, the frequency of leaf nodes can effectively capture the distribution of the dataset. Finally, these frequency data can be plotted to visualize. Fig. 14 shows the visualized distributions for some datasets. From these figures, we can find that in the random dataset, the frequency of leaf nodes is nearly uniform, whereas in the expert dataset, the frequency of leaf nodes is uneven. The distribution of the learning dataset and the hybrid dataset falls between that of the expert dataset and the random dataset. These observations confirm that the distribution of these datasets differs, thus validating the diversity of our proposed offline datasets.



(a) Kuhn poker (2p) (b) Kuhn poker (3p) (c) Leduc poker (2p) (d) Leduc poker (3p)

Figure 14: Frequency of leaf nodes in different offline datasets.

D THEORETICAL ANALYSIS

D.1 GENERALIZATION BOUND FOR TRAINING MODEL

As described in the main paper, to conduct the BOMB framework, we need to train one behavior cloning policy and an environment model, which are both neural network models. Furthermore, these two models are trained in a supervised learning manner with different loss functions based on the offline EF dataset. Here, we provide a generalization bound for training such neural network models, facilitating the following analysis of the BOMB framework.

As we know, the supervised learning framework includes a data-generation distribution σ, a hypothesis class $\mathcal{H}$ of the neural network approximator, a training dataset $\mathcal{D}$, and evaluation metrics to

evaluate the performance of any approximator. Here, we can use the loss function l to evaluate the performance of any approximation. The learning framework aims to minimize the true risk function $L_\sigma(h)$ which is the expected loss function of $h \in \mathcal{H}$ under the distribution σ,

$$L_\sigma(h) = \mathbb{E}_{d\sim\sigma}[l(h(d), d)].$$

Accordingly, the empirical risk function $L_\mathcal{D}(h)$ on the training dataset $\mathcal{D}$ can be defined as:

$$L_\mathcal{D} = \frac{1}{|\mathcal{D}|}\sum_{d\sim\mathcal{D}}[l(h(d), d)].$$

For a hypothesis h and a sample $d = (x, y)$, we write $\ell(h, d) = l(h(x), y)$ for the loss of h on d, and $\ell \circ \mathcal{H} = \{d \mapsto \ell(h, d) : h \in \mathcal{H}\}$, following Shalev-Shwartz & Ben-David (2014). To get a generalization bound, we use an auxiliary lemma from (Shalev-Shwartz & Ben-David, 2014). Therefore, we can measure the capacity of the composition function class $\ell \circ \mathcal{H}$ using the empirical Rademacher complexity on the training set $\mathcal{D}$ with size m, which is defined as:

$$\mathcal{R}_\mathcal{D}(\ell \circ \mathcal{H}) = \frac{1}{m}\mathbb{E}_{\mathbf{x}\sim\{+1,-1\}^m}[\sup_{h\in\mathcal{H}}\sum_{i=1}^{m} x_i \cdot l(h(d_i), d_i)]$$

where **x** is distributed i.i.d. according to uniform distribution in $\{+1, -1\}$. Before providing the generalization bound, we first provide the distance between two different approximators and a common theorem to facilitate the proof of the generalization bound.

Definition D.1. (r-cover) We say function class $\mathcal{H}_r$ r-cover $\mathcal{H}$ under $\ell_{\infty,1}$-distance if $\forall h, h \in \mathcal{H}$, there exists h_r in $\mathcal{H}_r$ such that $||h - h_r||_{\infty,1} = \max_{x\in\mathcal{D}} ||h(x) - h_r(x)||_1 \leq r$.

Definition D.2. (r-covering number) The r-covering number of $\mathcal{H}$, $\mathcal{N}_{\infty,1}(\mathcal{H}, r)$, is the cardinality of the smallest function class H_r that r-covers $\mathcal{H}$ under $\ell_{\infty,1}$-distance.

Theorem D.3 (Shalev-Shwartz & Ben-David, 2014, Theorem 26.5)**.** *Assume that $|\ell(h, d)| \leq B$ for all $h \in \mathcal{H}$ and all d. Let $\mathcal{D}$ be a training set of size m drawn i.i.d. from σ. Then with probability at least $1 - \rho$ over the draw of $\mathcal{D}$, for all $h \in \mathcal{H}$,*

$$L_\sigma(h) - L_\mathcal{D}(h) \leq 2\mathcal{R}_\mathcal{D}(l \circ \mathcal{H}) + 4B\sqrt{2\ln(4/\rho)/m}.$$

We provide a bound to measure the generalizability of the trained approximator in a training dataset of size m using the following theorem.

Theorem D.4 (Generalization bound)**.** *Assume that $\ell(h, d) = l(h(x), y)$ takes values in $[0, B]$ and that $l(x, y)$ is T-Lipschitz continuous, then for hypothesis class $\mathcal{H}$ of approximator and distribution σ, with probability at least $1 - \rho$ over draw of the training set $\mathcal{D}$ with size m from σ, for all $h \in \mathcal{H}$, we have*

$$L_\sigma(h) - L_\mathcal{D}(h) \leq 2 \cdot \inf_{r>0}\left[B\sqrt{\frac{2\log\mathcal{N}_{\infty,1}(\mathcal{H}, r)}{m}} + Tr\right] + 4B\sqrt{\frac{2\ln(4/\rho)}{m}}.$$

Proof. According to Theorem D.3, we have

$$L_\sigma(h) - L_\mathcal{D}(h) \leq 2\mathcal{R}_\mathcal{D}(\ell \circ \mathcal{H}) + 4B\sqrt{\frac{2\ln(4/\rho)}{m}}.$$

According to the assumption, the loss function $l(x, y)$ is T-Lipschitz continuous under ℓ_k-distance, i.e., $|l(x, y) - l(x', y)| \leq T||x - x'||_k$, where $||\cdot||_k$ is the k-norm. Let $\mathcal{H}_r$ be the function class that r-cover $\mathcal{H}$ for some $r > 0$ and $|\mathcal{H}_r| = \mathcal{N}_{\infty,1}(\mathcal{H}, r)$ be the r-covering number of $\mathcal{H}_r$. For all $h \in \mathcal{H}$, $h_r \in \mathcal{H}_r$ is denoted to be the function approximator that r-covers h. Based on above equations, we have

$$|l(h(x), y) - l(h_r(x), y)| \leq T||h(x) - h_r(x)||_k \leq Tr.$$

Then we have

$$\mathcal{R}_{\mathcal{D}}(\ell \circ \mathcal{H}) = \frac{1}{m}\mathbb{E}_{\mathbf{x}\sim\{+1,-1\}^m}[\sup_{h\in\mathcal{H}} \sum_{i=1}^{m} x_i \cdot l(h(d_i), d_i)] \tag{1}$$

$$= \frac{1}{m}\mathbb{E}_{\mathbf{x}\sim\{+1,-1\}^m}[\sup_{h\in\mathcal{H}} \sum_{i=1}^{m} x_i \cdot (l(h_r(d_i), d_i) + l(h(d_i), d_i) - l(h_r(d_i), d_i))] \tag{2}$$

$$\leq \frac{1}{m}\mathbb{E}_{\mathbf{x}\sim\{+1,-1\}^m}[\sup_{h_r\in\mathcal{H}_r} \sum_{i=1}^{m} x_i \cdot l(h_r(d_i), d_i)] + \frac{1}{m}\mathbb{E}_{\mathbf{x}\sim\{+1,-1\}^m}[\sup_{h\in\mathcal{H}} \sum_{i=1}^{m} |x_i \cdot Tr|] \tag{3}$$

$$\leq \sup_{h_r\in\mathcal{H}_r} \sqrt{\sum_{i=1}^{m}(\ell(h_r, d_i))^2} \cdot \frac{\sqrt{2\log \mathcal{N}_{\infty,1}(\mathcal{H}, r)}}{m} + \frac{Tr}{m}\mathbb{E}_{\mathbf{x}}||\mathbf{x}||_1 \tag{4}$$

$$\leq B\sqrt{m}\frac{\sqrt{2\log \mathcal{N}_{\infty,1}(\mathcal{H}, r)}}{m} + Tr = B\sqrt{\frac{2\log \mathcal{N}_{\infty,1}(\mathcal{H}, r)}{m}} + Tr \tag{5}$$

The reduction from Eq. 3 to Eq. 4 is based on Massart's lemma (Shalev-Shwartz & Ben-David, 2014). Using $\sum_{i=1}^{m}(\ell(h_r, d_i))^2 \leq mB^2$, we can get the reduction from Eq. 4 to Eq. 5. Finally,

$$L_\sigma(h) - L_{\mathbf{D}}(h) \leq 2\mathcal{R}_{\mathcal{D}}(\ell \circ \mathcal{H}) + 4B\sqrt{\frac{2\ln(4/\rho)}{m}} \tag{6}$$

$$\leq 2 \cdot \inf_{r>0}\left[B\sqrt{\frac{2\log \mathcal{N}_{\infty,1}(\mathcal{H}, r)}{m}} + Tr\right] + 4B\sqrt{\frac{2\ln(4/\rho)}{m}}$$

□

Theorem D.4 controls, for a training set of size m, the gap between the empirical risk of the trained network and its true risk under the data distribution. The two quantities in the following Assumption (Assumption 4.3) are controlled by this true risk: η is the ℓ_1 error of the BC policy at the information sets, and δ is the error of the environment model in expected utility. Both are therefore bounded, with probability $1-\rho$, by the training loss plus a term of order $\sqrt{\log \mathcal{N}_{\infty,1}(\mathcal{H}, r)/m} + \sqrt{\ln(4/\rho)/m}$, and both decrease to the training loss as m grows. The theorems of Section 4.2 take η and δ as given and are otherwise independent of how the networks are trained.

Assumption D.5. (Assumption 4.3) [Training Error] The trained BC policy σ_{BC} satisfies $\max_i \max_{I\in\mathcal{I}_i} \|\sigma_{BC,i}(I) - \sigma_{\mathcal{D},i}(I)\|_1 \leq \eta$, and the trained environment model, viewed as a game $\widehat{\mathcal{G}}$ on the state-action pairs in $\mathcal{D}$ and reproduces the transitions recorded in $\mathcal{D}$, satisfies $\max_i \sup_\sigma |u_i^{\widehat{\mathcal{G}}}(\sigma) - u_i^{\mathcal{G}}(\sigma)| \leq \delta$.

D.2 Insufficiency of Equilibrium Coverage

In the offline RL area, dataset coverage over the optimal policy is typically sufficient for offline learning (Rashidinejad et al., 2021; Xie et al., 2021). However, in the offline EF paradigm, the assumption that the equilibrium strategy generates the dataset is not sufficient for computing equilibrium strategies offline. This insufficiency is illustrated by the counterexample in Fig. 15. In this example, we can easily get the NE strategy, $\sigma^* = (\sigma_1^*, \sigma_2^*) = (\{I_1 : a_1\}, \{I_2 : b_2\})$. If this equilibrium strategy is used to generate the offline dataset $\mathcal{D}$, the dataset $\mathcal{D}$ would only contain the single data point $((I_1^{t_1} = I_1, I_2^{t_1} = \emptyset, \emptyset, 1, \{a_1, a_2\}), a_1, (I_1^{t_2} = I_1 a_1, I_2^{t_2} = \emptyset, \emptyset, -1, \emptyset), (0, 0), 1)$. Clearly, the dataset $\mathcal{D}$ lacks any information about Player 2, making it insufficient for computing the NE strategy. Apart from the above assumption, we also provide another dataset assumption related to the equilibrium strategy.

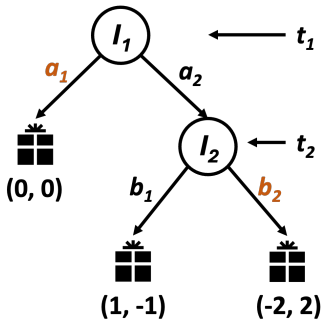
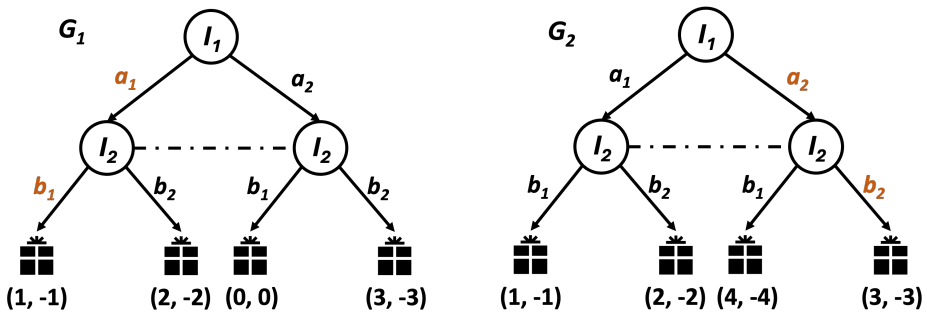


Figure 15: Game example.



(a) G_1 game. (b) G_2 game.

Figure 16: Counterexample for proving Theorem D.7.

Assumption D.6. (Single Strategy Coverage) The offline dataset $\mathcal{D}$ is said to be a *single strategy coverage* if the equilibrium strategy profile σ^* is covered by $\mathcal{D}$, i.e., for each player i, each information set I_i and action a_i with $\sigma_i^*(I_i, a_i) > 0$, there is a state-action pair (s_t, a_i) in $\mathcal{D}$.

The single strategy coverage assumption regarding the optimal strategy is sufficient for computing the optimal strategy in the offline RL setting. Subsequently, a question arises: *Is the single strategy coverage assumption also sufficient for computing the equilibrium strategy in the offline setting?* We employ the following theorem to answer this question and elucidate the rationale behind this.

Theorem D.7. *The single strategy coverage assumption over the offline dataset $\mathcal{D}$ is not sufficient for computing an ϵ-equilibrium for an arbitrarily small ϵ in the offline EF setting.*

Proof. We prove this theorem by providing a counterexample. First, we consider two two-player IIEFGs G_1 and G_2, as shown in Fig. 16. We can easily find that the NE of the game G_1 is strategy profile $\sigma^1 = (\sigma_1^1, \sigma_2^1) = (\{I_1 : a_1\}, \{I_2 : b_1\})$, i.e., Player 1 plays a_1 at information set I_1 and Player 2 plays b_1 at information set I_2. The NE of the game G_2 is strategy profile $\sigma^2 = (\sigma_1^2, \sigma_2^2) = (\{I_1 : a_2\}, \{I_2 : b_2\})$. Next, we consider an offline dataset $\mathcal{D}$ which is generated using a strategy profile $\sigma_{\mathcal{D}}$. The strategy profile $\sigma_{\mathcal{D}}$ is set to be the uniform distribution over the strategy profiles σ^1 and σ^2, which means that the dataset $\mathcal{D}$ covers both σ^1 and σ^2. Therefore, the offline dataset $\mathcal{D}$ satisfies the single strategy coverage assumption for these two games G_1 and G_2. However, no algorithm can distinguish these two games only based on the dataset $\mathcal{D}$ since these two games are both consistent on the dataset $\mathcal{D}$. In conclusion, the single strategy coverage assumption is not sufficient for computing an ϵ-equilibrium in the offline EF setting. □

From the above proof, we understand that the single strategy coverage assumption is sufficient for computing the optimal strategy in the offline RL setting, but is inadequate for computing an NE strategy in the offline EF setting. The intuition behind this difference lies in the two problems: in offline RL, data from only two actions is sufficient to determine which action is better; however, in offline EF, data from only two action pairs cannot determine which action pair is closer to NE. Identifying an NE strategy requires referencing additional action pairs for comparison and inference. Based on this analysis, Cui et al. (Cui & Du, 2022) provided a minimal coverage assumption that is sufficient for computing NE in two-player zero-sum Markov games, which is defined as follows:

Assumption D.8. (Deterministic Unilateral Coverage) For all deterministic strategy σ_i for player i, $(\sigma_i, \sigma_{-i}^*)$ are covered by the dataset, where $(\sigma_1^*, ..., \sigma_n^*)$ is one NE strategy.

Assumption D.9. (Unilateral Coverage) For all (possible stochastic) strategy σ_i for all player i, $(\sigma_i, \sigma_{-i}^*)$ are covered by the dataset, where $(\sigma_1^*, ..., \sigma_n^*)$ is one NE strategy.

Note that the deterministic unilateral coverage assumption is equivalent to the unilateral coverage assumption. The intuition behind this is that any mixed strategy can be represented by a combination of deterministic strategies. Therefore, if all deterministic strategies are covered by the dataset, then all mixed strategies are also covered. Based on this finding, in the following proof, we only consider all deterministic strategies. Previously, Cui et al. (Cui & Du, 2022) established that the unilateral coverage assumption is the minimal sufficient condition for computing an NE strategy

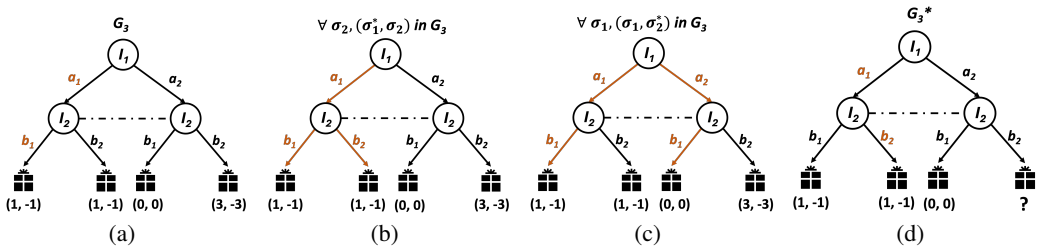


Figure 17: Counter-example for proving Theorem D.10.

in the two-player zero-sum Markov games. However, this unilateral coverage assumption over the offline dataset is **not sufficient for our model-based method** to compute the equilibrium strategy in the offline setting. We formally proved this limitation through the following theorem.

Theorem D.10. *The unilateral coverage assumption over the offline dataset $\mathcal{D}$ is not sufficient for our model-based method to converge to an ϵ-equilibrium for an arbitrarily small ϵ in the offline equilibrium finding setting.*

Proof. We prove this theorem by providing a counterexample. First, we consider an IIEFG G_3, which is represented in Fig. 17(a). We can easily find that the NE strategy of game G_3 is strategy profile $\sigma^* = (\sigma_1, \sigma_2) = (\{I_1 : a_1\}, \{I_2 : b_1\})$. To build a dataset $\mathcal{D}$ satisfying the unilateral coverage assumption, the dataset needs to cover (σ_1^*, σ_2) for all deterministic strategies σ_2 and (σ_1, σ_2^*) for all deterministic strategies σ_1. We show the state-action pairs covered by these strategy profiles in Figs. 17(b)-17(c). It means that if the dataset $\mathcal{D}$ satisfies the unilateral coverage assumption, then the dataset $\mathcal{D}$ would cover these state-action pairs marked by all these orange lines.

When applying our model-based method to compute the NE strategy only based on the dataset $\mathcal{D}$, the first step is to train an environment model using the dataset $\mathcal{D}$. Assume that the environment model can be trained well, which means that the environment model can precisely represent all game information in the dataset. Therefore, the game represented by the trained environment model would be G_3^* in Fig. 17(d). Note that there is some missing game data in the dataset $\mathcal{D}$ (i.e., marked by a question mark in Fig. 17(d)). Although our trained environment model can yield approximate results for these missing parts, it may lead to a different equilibrium strategy. For example, if the missing value in G_3^* is $(0, 0)$ or $(-1, 1)$, then the strategy profile $\sigma = (\sigma_1, \sigma_2') = (\{I_1 : a_1\}, \{I_2 : b_2\})$ would be the NE strategy of game G_3^*. However, the strategy profile σ is not an NE of the original game G_3.

In conclusion, through the above counterexample, we prove that the unilateral coverage assumption over the offline dataset is not sufficient for our model-based method to converge to an ϵ-equilibrium for an arbitrarily small ϵ. □

D.3 Convergence Guarantee Results of Our Method

In this section, we provide some theoretical results about the performance of our method under different dataset coverage assumptions. Before that, we provide some preliminaries for the proofs.

Notation. $\mathcal{G}$ is a finite IIEFG with perfect recall. N is its set of players, i.e., $n = |N|$. Z is its terminal histories, and $\sigma_c(h) \in \Delta(A(h))$ is the fixed chance distribution at a chance node h. For a history h, we write $I(h)$ for the information set containing it when $P(h) \in N$. For a behavior strategy profile σ, $\pi^\sigma(z) = \prod_{ha \sqsubseteq z} \sigma_{P(h)}(h)(a)$ is the reach probability of z, where $\sigma_{P(h)}(h)$ stands for $\sigma_{P(h)}(I(h))$ at player nodes and for $\sigma_c(h)$ at chance nodes, and $u_i(\sigma) = \sum_{z \in Z} \pi^\sigma(z) u_i(z)$. As in Sec. 4.2, $U = \max_{i \in N, z \in Z} |u_i(z)|$ and H is the maximum number of player (non-chance) nodes on a root-to-leaf path. We also let H_c be the maximum number of chance nodes on such a path.

The Equilibrium Gap. For a profile σ, let $\mathrm{BR}_i(\sigma_{-i})$ denote the largest expected utility player i can obtain by a unilateral deviation from σ. Over behavior strategies τ_i when the solution concept is NE, and over pure strategies s_i played against the marginal σ_{-i} of a joint distribution σ when the

solution concept is CCE. In both cases

$$\textsc{Gap}_{\mathcal{G}}(\sigma) = \sum_{i \in N} \big[\mathrm{BR}_i(\sigma_{-i}) - u_i(\sigma)\big], \tag{7}$$

which is NASHCONV for NE and the (C)CE gap sum of Marris et al. (2021) for (C)CE. The maxima defining BR_i exist because the strategy sets are compact and u_i is continuous.

The Behavior Strategy of the Dataset. Since a behavior strategy must assign the same distribution to all histories in an information set, we take $\sigma_{\mathcal{D}}$ to be the behavior strategy profile induced by the counts aggregated over each information set,

$$\sigma_{\mathcal{D},i}(I)(a) = \frac{\sum_{h \in I} C(h, a)}{\sum_{h \in I} C(h)}, \qquad I \in \mathcal{I}_i,\ a \in A(I),$$

which is exactly the empirical conditional distribution that the cross-entropy loss $\mathcal{L}_{bc}$ of the behavior cloning method is minimized by. The positivity condition $\sigma_{\mathcal{D}}(s_t, a_t) > 0$ for all s_t and $a_t \in A(s_t)$ implies $C(h) > 0$ for every history, hence every information set of every player appears in $\mathcal{D}$ and $\sigma_{\mathcal{D}}$ is defined everywhere. This is the only consequence of positivity we use.

The Learned Game by Model-based Method. The trained environment model E_{θ_e} is viewed as a game $\widehat{\mathcal{G}}$ as follows. Its histories are the states appearing in $\mathcal{D}$, its player and chance nodes and their action sets are those recorded in the states, and its transitions are the transitions recorded in $\mathcal{D}$: these are deterministic given the state and the action (chance outcomes are recorded in GI). The model is trained on all of them, and we assume it reproduces them. Its terminal utilities are those output by the model. Under uniform coverage (Definition 4.1), every state-action pair of $\mathcal{G}$ appears in $\mathcal{D}$, so $\widehat{\mathcal{G}}$ has the same tree, information sets, and action sets as $\mathcal{G}$ and differs from it only in chance probabilities and terminal utilities. Assumption 4.3 bounds the resulting discrepancy in expected utility by δ.

Lemma D.11 (Stability of the GAP under model error)**.** *Let $\mathcal{G}$ and $\widehat{\mathcal{G}}$ share the same tree, information sets, and action sets, and let δ be as in Assumption 4.3. Then for every strategy profile σ and either instantiation of* GAP*,*

$$\big|\textsc{Gap}_{\mathcal{G}}(\sigma) - \textsc{Gap}_{\widehat{\mathcal{G}}}(\sigma)\big| \le 2n\delta.$$

Proof. For every strategy profile $\sigma = (\tau_i, \sigma_{-i})$, $|u_i^{\widehat{\mathcal{G}}}(\tau_i, \sigma_{-i}) - u_i^{\mathcal{G}}(\tau_i, \sigma_{-i})| \le \delta$ according to Assumption 4.3. Since $|\max_x f(x) - \max_x g(x)| \le \sup_x |f(x) - g(x)|$ for any two functions on the same set, we can get

$$|\max_{\tau_i} u_i^{\widehat{\mathcal{G}}}(\tau_i, \sigma_{-i}) - \max_{\tau_i} u_i^{\mathcal{G}}(\tau_i, \sigma_{-i})| \le \sup_{\tau_i} |u_i^{\widehat{\mathcal{G}}}(\tau_i, \sigma_{-i}) - u_i^{\mathcal{G}}(\tau_i, \sigma_{-i})|$$

$$|\mathrm{BR}_i^{\mathcal{G}}(\sigma_{-i}) - \mathrm{BR}_i^{\widehat{\mathcal{G}}}(\sigma_{-i})| \le \sup_{\tau_i} |u_i^{\widehat{\mathcal{G}}}(\tau_i, \sigma_{-i}) - u_i^{\mathcal{G}}(\tau_i, \sigma_{-i})| \le \delta$$

Next, according to Equation 7, we can compute

$$\begin{aligned}
\big|\textsc{Gap}_{\mathcal{G}}(\sigma) - \textsc{Gap}_{\widehat{\mathcal{G}}}(\sigma)\big| &= \Big|\sum_{i \in N} \big[\mathrm{BR}_i^{\mathcal{G}}(\sigma_{-i}) - u_i^{\mathcal{G}}(\sigma)\big] - \sum_{i \in N} \big[\mathrm{BR}_i^{\widehat{\mathcal{G}}}(\sigma_{-i}) - u_i^{\widehat{\mathcal{G}}}(\sigma)\big]\Big| \\
&= \Big|\sum_{i \in N} \big[(\mathrm{BR}_i^{\mathcal{G}}(\sigma_{-i}) - \mathrm{BR}_i^{\widehat{\mathcal{G}}}(\sigma_{-i})) + (u_i^{\widehat{\mathcal{G}}}(\sigma) - u_i^{\mathcal{G}}(\sigma))\big]\Big| \\
&\le \sum_{i \in N} \Big|\big[(\mathrm{BR}_i^{\mathcal{G}}(\sigma_{-i}) - \mathrm{BR}_i^{\widehat{\mathcal{G}}}(\sigma_{-i})\big) + (u_i^{\widehat{\mathcal{G}}}(\sigma) - u_i^{\mathcal{G}}(\sigma))\big]\Big| \\
&\le n(\delta + \delta) = 2n\delta
\end{aligned}$$

The argument is the same for both instantiations of GAP: for NE, τ_i ranges over behavior strategies and σ is a behavior profile. For CCE, τ_i ranges over pure strategies s_i and σ_{-i} is the marginal of the joint distribution σ, in which case (s_i, σ_{-i}) is again a joint distribution over pure profiles and Assumption 4.3 bounds its utility discrepancy by δ. □

Lemma D.12 (Coupling)**.** *Let σ, σ' be two behavior profiles of $\mathcal{G}$ and let $\eta = \max_{i \in N} \max_{I \in \mathcal{I}_i} \|\sigma_i(I) - \sigma_i'(I)\|_1$. Then $|u_i(\sigma) - u_i(\sigma')| \le UH\eta$ for every player i.*

Proof. We first recall that for two distributions p, q on a finite set, $\|p-q\|_{\mathrm{TV}} = \frac{1}{2}\|p-q\|_1$, and that this quantity equals the minimum of $\Pr(a \neq a')$ over all couplings (a, a') of p and q, the minimum being attained by an *optimal coupling* (Levin & Peres, 2017, Props. 4.2, 4.7 and Remark 4.8).

Based on this fact, we construct a joint distribution over pairs of terminal histories (z, z') whose marginals are π^σ and $\pi^{\sigma'}$, by running the following process on pairs of histories starting from $(\varnothing, \varnothing)$. While the two histories coincide, say both equal a nonterminal history h, proceed as follows. If $P(h) = c$, draw $a \sim \sigma_c(h)$ and move to (ha, ha). Note that σ and σ' share the chance distribution σ_c. If $P(h) = j \in N$, let γ_h^* be an optimal coupling of the two action distributions $p = \sigma_j(I(h))$ and $q = \sigma'_j(I(h))$, i.e., a joint distribution over pairs $(a, a') \in A(I(h))^2$ with marginals p and q that minimizes $\Pr(a \neq a')$. By the fact recalled above, γ_h^* exists and satisfies $\Pr_{\gamma_h^*}(a \neq a') = \frac{1}{2}\|\sigma_j(I(h)) - \sigma'_j(I(h))\|_1 \leq \eta/2$. Draw $(a, a') \sim \gamma_h^*$ and move to (ha, ha'). Once the two histories differ, continue them independently, the first under σ and the second under σ', until both are terminal, and let z and z' denote the terminal histories reached by the first and second coordinate, respectively. At every step, each coordinate is advanced according to the action distribution prescribed by its own profile at its current history, so by induction on the depth $z \sim \pi^\sigma$ and $z' \sim \pi^{\sigma'}$.

Let E be the event $z \neq z'$. The two histories can separate only at a player node of their common prefix, and a common prefix contains at most H player nodes. For $k = 1, \ldots, H$, let E_k be the event that the pair first separates at the k-th player node of the common prefix. Then $E = \bigcup_{k=1}^{H} E_k$, and $\Pr[E_k]$ is at most the probability of separating at that node conditional on having stayed together before it, which is at most $\eta/2$ by the choice of γ_h^*. Hence $\Pr[E] \leq \sum_{k=1}^{H} \Pr[E_k] \leq H\eta/2$. On the complement of E, we have $z = z'$ and hence $u_i(z) - u_i(z') = 0$, while $|u_i(z) - u_i(z')| \leq 2U$ always. Therefore

$$|u_i(\sigma) - u_i(\sigma')| = \big|\mathbb{E}[u_i(z) - u_i(z')]\big| = \big|\mathbb{E}\big[(u_i(z) - u_i(z'))\mathbf{1}_E\big]\big| \leq 2U \Pr[E] \leq UH\eta. \quad \square$$

Lemma D.13. *For behavior profiles σ, σ' and η as in Lemma D.12,*

$$\big|\textsc{Gap}_{\mathcal{G}}(\sigma) - \textsc{Gap}_{\mathcal{G}}(\sigma')\big| \leq 2nUH\eta.$$

Proof. Fix $i \in N$. For any behavior strategy τ_i of player i, the two profiles (τ_i, σ_{-i}) and (τ_i, σ'_{-i}) coincide at every information set of player i and differ by at most η in ℓ_1 at every information set of the other players, so Lemma D.12 applied to this pair gives

$$|u_i(\tau_i, \sigma_{-i}) - u_i(\tau_i, \sigma'_{-i})| \leq UH\eta \qquad \text{for every } \tau_i. \tag{8}$$

Since $|\max_x f(x) - \max_x g(x)| \leq \sup_x |f(x) - g(x)|$ for any two bounded functions f, g on the same set, applying this with $f(\tau_i) = u_i(\tau_i, \sigma_{-i})$ and $g(\tau_i) = u_i(\tau_i, \sigma'_{-i})$, we can get

$$\begin{aligned} |\mathrm{BR}_i(\sigma_{-i}) - \mathrm{BR}_i(\sigma'_{-i})| &= \Big|\max_{\tau_i} u_i(\tau_i, \sigma_{-i}) - \max_{\tau_i} u_i(\tau_i, \sigma'_{-i})\Big| \\ &\leq \sup_{\tau_i} \big|u_i(\tau_i, \sigma_{-i}) - u_i(\tau_i, \sigma'_{-i})\big| \leq UH\eta. \end{aligned}$$

Lemma D.12 applied to σ and σ' directly gives $|u_i(\sigma) - u_i(\sigma')| \leq UH\eta$. Therefore, by Equation7,

$$\begin{aligned} |\textsc{Gap}_{\mathcal{G}}(\sigma) - \textsc{Gap}_{\mathcal{G}}(\sigma')| &\leq \sum_{i \in N} \Big(|\mathrm{BR}_i(\sigma_{-i}) - \mathrm{BR}_i(\sigma'_{-i})| + |u_i(\sigma) - u_i(\sigma')|\Big) \\ &\leq \sum_{i \in N} (UH\eta + UH\eta) = 2nUH\eta. \end{aligned}$$

$\square$

Remark D.14. For a behavior profile, which is a product of independent strategies, the two instantiations of $\textsc{Gap}_{\mathcal{G}}$ coincide: the marginal σ_{-i} of a product distribution is the product of the other players' strategies, and the best response to a fixed opponent profile can be taken pure, so $\max_{s_i} u_i(s_i, \sigma_{-i}) = \max_{\tau_i} u_i(\tau_i, \sigma_{-i})$ and the CCE gap sum equals NashConv. Lemma D.13 and Theorems 4.5 and 4.6 are therefore valid for either instantiation, although BC and BOMB are only used for NE in this paper.

We first recall the assumptions for dataset coverage, as in the main paper.

Definition D.15. An offline dataset $\mathcal{D}$ is said to be a **uniform coverage** of an IIEFG $\mathcal{G}$ if and only if it covers all possible state-action pairs in the game. Formally, it covers $(s_t, a_t, s_{t+1}, u_{t+1}, d_{t+1}), \forall s_t, a_t \in A(s_t)$ and $s_{t+1} \in T(s_t, a_t)$ where T is the transition function.

Definition D.16. An offline dataset $\mathcal{D}$ is said to be an ϵ-**equilibrium coverage** of an IIEFG $\mathcal{G}$ if and only if its underlying behavior strategy $\sigma_{\mathcal{D}}$ satisfies: $\text{GAP}_{\mathcal{G}}(\sigma_{\mathcal{D}}) \leq \epsilon$, where $\sigma_{\mathcal{D}}$ is defined as: $\sigma_{\mathcal{D}}(s_t, a_t) = \frac{C(s_t, a_t)}{C(s_t)}$ and $\sigma_D(s_t, a_t) > 0$ for all s_t and $a_t \in A(s_t)$, with $C(s_t, a_t)$ and $C(s_t)$ denoting the counts of occurrences of state-action pair (s_t, a_t) and the state s_t in $\mathcal{D}$, respectively.

Next, we provide the full proofs of the theorems for our method provided in the main paper.

Theorem D.17. *(Theorem 4.4) Under Assumption 4.3, if $\mathcal{D}$ is a uniform coverage of $\mathcal{G}$ and σ_{MB} is an ϵ-equilibrium of a trained game model $\widehat{\mathcal{G}}$, then* $\text{GAP}_{\mathcal{G}}(\sigma_{MB}) \leq \epsilon + 2n\delta$.

Proof of Theorem 4.4. Recall the introduction of **the learned game by model-based method** in the preliminaries for the proofs. Under uniform coverage, the learned game $\widehat{\mathcal{G}}$ shares the tree, information sets, and action sets of $\mathcal{G}$, and Assumption 4.3 bounds its discrepancy from $\mathcal{G}$ in expected utility by δ. Lemma D.11 applied to σ_{MB} gives $\text{GAP}_{\mathcal{G}}(\sigma_{MB}) \leq \text{GAP}_{\widehat{\mathcal{G}}}(\sigma_{MB}) + 2n\delta$. Since σ_{MB} is an ϵ-equilibrium of $\widehat{\mathcal{G}}$, $\text{GAP}_{\widehat{\mathcal{G}}}(\sigma_{MB}) \leq \epsilon$, and the claim follows. □

Remark D.18 (The first term is computable offline). $\text{GAP}_{\widehat{\mathcal{G}}}(\sigma_{MB})$ is the convergence error of the online algorithm inside the learned game. It requires only best responses in $\widehat{\mathcal{G}}$, which is fully known to the learner, and can therefore be evaluated without any interaction with $\mathcal{G}$.

Theorem D.19. *(Theorem 4.5) Under Assumption 4.3, if $\mathcal{D}$ is an ϵ-equilibrium coverage of $\mathcal{G}$ and σ_{BC} is the strategy returned by behavior cloning algorithm, then* $\text{GAP}_{\mathcal{G}}(\sigma_{BC}) \leq \epsilon + 2nUH\eta$.

Proof of Theorem 4.5. Let $\mathcal{D}$ be an ϵ-equilibrium coverage of $\mathcal{G}$. By the definition of ϵ-equilibrium coverage and the introduction of **the behavior strategy of the dataset** in the preliminaries for the proofs, the positivity condition ensures that every information set appears in $\mathcal{D}$, so $\sigma_{\mathcal{D}}$ is a behavior profile of $\mathcal{G}$ defined at every information set, and $\text{GAP}_{\mathcal{G}}(\sigma_{\mathcal{D}}) \leq \epsilon$. Assumption 4.3 states that $\|\sigma_{BC,i}(I) - \sigma_{\mathcal{D},i}(I)\|_1 \leq \eta$ for every i and $I \in \mathcal{I}_i$. Lemma D.13 applied to σ_{BC} and $\sigma_{\mathcal{D}}$ gives

$$\text{GAP}_{\mathcal{G}}(\sigma_{BC}) \leq \text{GAP}_{\mathcal{G}}(\sigma_{\mathcal{D}}) + 2nUH\eta \leq \epsilon + 2nUH\eta. \qquad \square$$

Remark D.20 (The bound holds for any behavior). The proof only uses $\text{GAP}_{\mathcal{G}}(\sigma_{\mathcal{D}}) \leq \epsilon$ in the last step. Under positivity alone, $\text{GAP}_{\mathcal{G}}(\sigma_{BC}) \leq \text{GAP}_{\mathcal{G}}(\sigma_{\mathcal{D}}) + 2nUH\eta$ for a dataset generated by strategy profiles that are defined at every information set.

Theorem D.21. *Let $\sigma_i^{\alpha}(I) = \alpha\, \sigma_{BC,i}(I) + (1-\alpha)\, \sigma_{MB,i}(I)$ and let $\mathcal{A} \subset [0,1]$ be a finite grid containing 0 and 1. Under Assumption 4.3:*

(i) If $\alpha^\star$ minimizes $\text{GAP}_{\mathcal{G}}(\sigma^\alpha)$ over $\mathcal{A}$, then $\text{GAP}_{\mathcal{G}}(\sigma^{\alpha^\star}) \leq \min\{\text{GAP}_{\mathcal{G}}(\sigma_{BC}),\ \text{GAP}_{\mathcal{G}}(\sigma_{MB})\}$. In particular, $\text{GAP}_{\mathcal{G}}(\sigma_{MB}) \leq \text{GAP}_{\widehat{\mathcal{G}}}(\sigma_{MB}) + 2n\delta$ whenever $\mathcal{D}$ is a uniform coverage of $\mathcal{G}$, and $\text{GAP}_{\mathcal{G}}(\sigma_{BC}) \leq \text{GAP}_{\mathcal{G}}(\sigma_{\mathcal{D}}) + 2nUH\eta$ whenever $\sigma_{\mathcal{D}}(s_t, a_t) > 0$ for all s_t and $a_t \in A(s_t)$.

(ii) If $\widehat{\alpha}$ minimizes $\text{GAP}_{\widehat{\mathcal{G}}}(\sigma^\alpha)$ over $\mathcal{A}$, which requires no interaction with $\mathcal{G}$, and $\mathcal{D}$ is a uniform coverage of $\mathcal{G}$, then $\text{GAP}_{\mathcal{G}}(\sigma^{\widehat{\alpha}}) \leq \text{GAP}_{\mathcal{G}}(\sigma^{\alpha^\star}) + 4n\delta$, with $\alpha^\star$ as in (i).

Proof of Theorem 4.6. By definition of the combined policy, $\sigma^1 = \sigma_{BC}$ and $\sigma^0 = \sigma_{MB}$.

(i) Since $\alpha^\star$ minimizes $\text{GAP}_{\mathcal{G}}(\sigma^\alpha)$ over $\mathcal{A}$ and $\{0,1\} \subseteq \mathcal{A}$,

$$\text{GAP}_{\mathcal{G}}(\sigma^{\alpha^\star}) \leq \min\{\text{GAP}_{\mathcal{G}}(\sigma^1), \text{GAP}_{\mathcal{G}}(\sigma^0)\} = \min\{\text{GAP}_{\mathcal{G}}(\sigma_{BC}), \text{GAP}_{\mathcal{G}}(\sigma_{MB})\}.$$

If $\mathcal{D}$ is a uniform coverage of $\mathcal{G}$, using Theorem 4.4 gives $\text{GAP}_{\mathcal{G}}(\sigma_{MB}) \leq \text{GAP}_{\widehat{\mathcal{G}}}(\sigma_{MB}) + 2n\delta$. If $\sigma_{\mathcal{D}}(s_t, a_t) > 0$ for all s_t and $a_t \in A(s_t)$, then $\sigma_{\mathcal{D}}$ is defined at every information set and Lemma D.13 applied to σ_{BC} and $\sigma_{\mathcal{D}}$, exactly as in the proof of Theorem 4.5, gives $\text{GAP}_{\mathcal{G}}(\sigma_{BC}) \leq \text{GAP}_{\mathcal{G}}(\sigma_{\mathcal{D}}) + 2nUH\eta$.

(ii) Let $\alpha^\star$ be as in (i) and let $\widehat{\alpha}$ minimize $\text{GAP}_{\widehat{\mathcal{G}}}(\sigma^\alpha)$ over $\mathcal{A}$. Since $\mathcal{D}$ is a uniform coverage of $\mathcal{G}$, the learned game $\widehat{\mathcal{G}}$ shares the tree, information sets, and action sets of $\mathcal{G}$, so every σ^α is a behavior

profile of both games, $\text{GAP}_{\widehat{\mathcal{G}}}(\sigma^{\alpha})$ is well defined, and Lemma D.11 applies to each σ^{α}. Therefore

$$\text{GAP}_{\mathcal{G}}(\sigma^{\widehat{\alpha}}) \leq \text{GAP}_{\widehat{\mathcal{G}}}(\sigma^{\widehat{\alpha}}) + 2n\delta \leq \text{GAP}_{\widehat{\mathcal{G}}}(\sigma^{\alpha^\star}) + 2n\delta \leq \text{GAP}_{\mathcal{G}}(\sigma^{\alpha^\star}) + 4n\delta,$$

where the first and third inequalities are Lemma D.11, and the second holds because $\widehat{\alpha}$ minimizes $\text{GAP}_{\widehat{\mathcal{G}}}(\sigma^{\alpha})$ over $\mathcal{A}$ and $\alpha^\star \in \mathcal{A}$. □

Remark D.22 (On the mixing parameter). Theorem 4.6 does not claim that intermediate values of α are never worse than the endpoints. With a common α at every information set, the reach probabilities are polynomials in α and $\text{GAP}_{\mathcal{G}}(\sigma^{\alpha})$ need not be convex. What the theorem guarantees is that selecting α by grid search, whether in the true game (i) or entirely offline in the learned game (ii), inherits the better of the two component guarantees, up to $4n\delta$ in the offline case.

E Implementation Details

Here, we provide the details for the model-based method by introducing our instantiable algorithms: MB-PSRO and MB-CFR, which are adaptations of two widely used online equilibrium-finding algorithms, PSRO and Deep CFR.

E.1 MB-PSRO

Algorithm 2 MB-PSRO

Input: Trained environment model E_{θ_e}
Initial policy sets Π for all players;
Compute expected rewards U^{Π} for each strategy $\pi \in \Pi$ *based on the environment model* E_{θ_e};
Initialize meta-strategies $\sigma_i = \text{UNIFORM}(\Pi_i), \forall i$;
repeat
  for each player $i \in [1, .., n]$ **do**
    for best response episodes $t \in [1, ..., T]$ **do**
      Sample $\pi_{-i} \sim \sigma_{-i}$;
      Train best response policy π_i' over $\rho \sim (\pi_i', \pi_{-i})$, which *samples on the environment model* E_{θ_e};
    end for
    add the best response policy π_i' to policy set Π_i;
  end for
  Compute missing entries in U^{Π} *based on the environment model* E_{θ_e};
  Compute the meta-strategy σ using any meta-solver;
until Meet the convergence condition
Output: Policy set Π and meta-strategy σ

We present the whole framework in Alg. 2. In the beginning, we need a well-trained environment model E_{θ_e} as input to replace the function of the actual environment. Firstly, we initialize policy sets Π for all players using random strategies. Then, we estimate the expected utilities for each strategy profile based on the model E_{θ_e} to form the meta-game matrix. In vanilla PSRO, this process needs to interact with the actual game environment. However, in the offline setting, the actual game environment is not available. Therefore, we use the well-trained environment model E_{θ_e} to replace the actual game environment to provide the information needed in the algorithm. After building the meta-game matrix, the meta-strategy is initialized by a uniform strategy. Next, we compute the best response policy for every player and add these trained best response policies to their policy sets. When training the best response policy oracle using DQN or other RL algorithms, we sample the training data based on the environment model E_{θ_e}. After adding trained best response policies, we compute missing entries in the meta-game matrix based on the trained environment model E_{θ_e}. Then, the meta-strategy σ of the meta-game matrix can be computed using any meta-solver, such as the Nash solver or α-rank algorithm. For games with more than two players, the α-rank algorithm is taken as the meta-solver. Finally, we repeat the above processes until meeting the convergence condition and output the policy set and meta-strategy as the approximate equilibrium strategy.

Algorithm 3 MB-CFR

1: **Input:** Trained environment model E_{θ_e}
2: Initialize regret network $R(I, a|\theta_{r,p})$ for all players;
3: Initialize strategy network $S(I|\theta_{\pi,p})$ for all players;
4: Initialize regret memory $M_{r,p}$ and strategy memory $M_{\pi,p}$ for every player p;
5: **for** iteration $t = 1$ to T **do**
6: **for** player $p \in [1, ..., n]$ **do**
7: **for** traverse episodes $k \in [1, ..., K]$ **do**
8: **TRAVERSE**(ϕ, p, $\theta_{r,p}$, $\theta_{\pi,-p}$,$M_{r,p}$, $M_{\pi,-p}$, E_{θ_e});
 # Use sample algorithm to traverse game tree, record regret and strategy training data
9: **end for**
10: Train $\theta_{r,p}$ from scratch based on regret memory $M_{r,p}$ for every player p;
11: **end for**
12: **end for**
13: Train $\theta_{\pi,p}$ based on strategy memory $M_{\pi,p}$ for every player p;
14: **Output:**$\theta_{\pi,p}$ for every player p

To compute the CCE strategy, we also instantiate one algorithm: MB-JPSRO, an adaptation of the JPSRO algorithm. The process of JPSRO is similar to PSRO except for the best response computation and meta solver. Therefore, MB-JPSRO is also similar to MB-PSRO. For this reason, we do not cover MB-JPSRO in detail here.

E.2 MB-CFR

Alg. 3 shows the process of MB-CFR, which is adapted from the Deep CFR algorithm. It also needs the well-trained environment model E_{θ_e} as input for the MB-CFR algorithm. We first initialize regret and strategy networks for each player and then initialize regret and strategy memories for each player (Lines 2-4). Then we need to update the regret network for every player. To do this, we perform a traverse function to collect corresponding training data. The traverse function can be any sampling-based CFR algorithm. Here, we use the external sampling algorithm as the traverse method to collect training data, and the process of external sampling is shown in Alg. 4. In this traverse function, we collect the regret training data of the traveler, and the strategy training data of other players is also gathered. After performing the traverse function several times, the regret network can be updated based on the regret memory using a supervised learning algorithm. The above processes are repeated for T times. Then, the average strategy network for every player is trained based on its corresponding strategy memory. Finally, the trained average strategy networks are output as the approximate equilibrium strategy.

F Additional Experimental Results

In this section, we provide more experimental results and an ablation study. Finally, we provide the main parameters we used in our experiments.

F.1 Experimental Results

Here, we first verify that the performance of the model-based approach is independent of the algorithm used for computing the equilibrium strategy. To this end, we perform both MB-CFR and MB-PSRO algorithms in the two-player Kuhn poker game under different sizes of offline datasets. Fig. 18 shows the results. We can find that under the same size of an offline dataset, MB-PSRO and MB-CFR achieve nearly identical results. When the size of the offline dataset increases, the performance of both algorithms becomes better. It may be caused by the

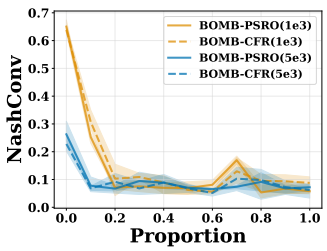
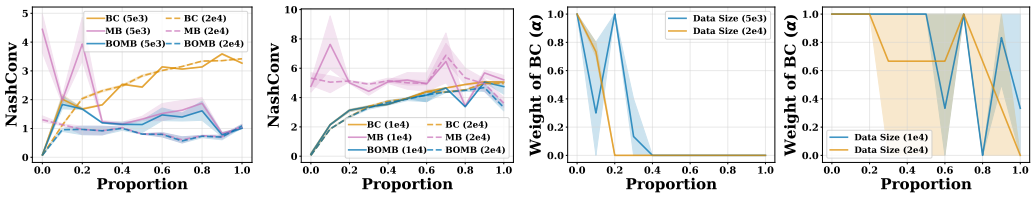


Figure 18: Results of different MB methods

Algorithm 4 TRVERSE($s, p, \theta_{r,p}, \theta_{\pi,-p}, M_{r,p}, M_{\pi,-p}, E_{\theta_e}$)-External Sampling Algorithm

if s is terminal state **then**
  Get the utility $u_p(s)$ from the environment model E_{θ_e};
  Output: $u_p(s)$
else if s is a chance state **then**
  Sample an action a according to the chance distribution over the available actions, which is obtained from model E_{θ_e};
  $s' = E_{\theta_e}(s, a)$;
  Output:TRAVERSE($s', p, \theta_{r,p}, \theta_{\pi,-p}, M_{r,p}, M_{\pi,-p}, E_{\theta_e}$)
else if $P(s) = p$ **then**
  $I \leftarrow s[p]$; # *Get the corresponding information set from the game state*
  $\sigma(I) \leftarrow$ strategy of I computed using regret values $R(I, a|\theta_{r,p})$ based on regret matching;
  for $a \in A(s)$ **do**
    $s' = E_{\theta_e}(s, a)$;
    $u(a) \leftarrow$ TRAVERSE($s', p, \theta_{r,p}, \theta_{\pi,-p}, M_{r,p}, M_{\pi,-p}, E_{\theta_e}$);
  end for
  $u_\sigma \leftarrow \sum_{a \in A(s)} \sigma(I, a)u(a)$;
  for $a \in A(s)$ **do**
    $r(I, a) \leftarrow u(a) - u_\sigma$;
  end for
  Insert the infoset and its action regret values $(I, r(I))$ into regret memory $M_{r,p}$;
  Output: u_σ
else
  $I \leftarrow s[p]$;
  $\sigma(s) \leftarrow$ strategy of I computed using regret value $R(I, a|\theta_{r,-p})$ based on regret matching;
  Insert the infoset and its strategy $(I, \sigma(s))$ into strategy memory $M_{\pi,-p}$;
  Sample an action a from distribution $\sigma(s)$;
  $s' = E_{\theta_e}(s, a)$;
  Output: TRAVERSE($s', p, \theta_{r,p}, \theta_{\pi,-p}, M_{r,p}, M_{\pi,-p}, E_{\theta_e}$);
end if

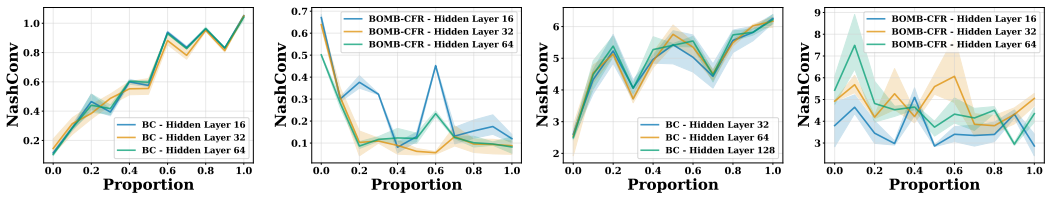
environment model being well-trained with more data. These observations indicate that the performance of the model-based algorithm is independent of the algorithm used to compute the equilibrium strategy and mainly relies on the similarity between the trained environment model and the actual environment.



(a) Kuhn poker (4p) (b) Kuhn poker (5p) (c) Kuhn poker (4p) (d) Kuhn poker (5p)

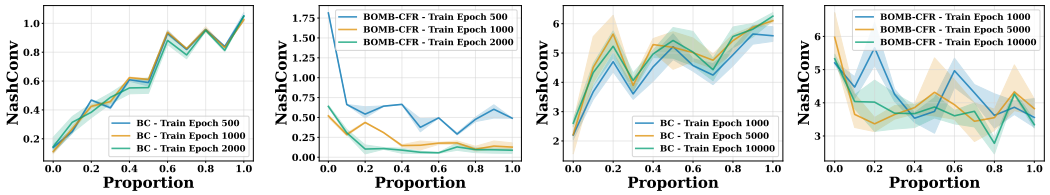
Figure 19: Performance of BOMB on multi-player games.

Then, we provide experimental results for the BOMB framework on multi-player games, including four-player and five-player Kuhn poker games, to compute NE strategies. The results, shown in Fig. 19, are consistent with those in the main paper. Across all cases, our BOMB framework outperforms or matches the performance of both BC and MB algorithms, aligning with the theoretical analysis. As the proportion of the random dataset increases, the performance of BC decreases, while the performance of MB is unstable and shows a slight downward trend. Here, we use OEF-CFR as the model-based algorithm, noting that CFR-based algorithms do not guarantee convergence to NE strategies in multi-player games. Additionally, the performance of the MB framework relies on the quality of the trained environment model. Poor performance in the MB algorithm may stem from either an inadequately trained environment model or the limitations of CFR-based algorithms



(a) Kuhn poker (2p) (b) Kuhn poker (2p) (c) Leduc poker (2p) (d) Leduc poker (2p)

Figure 20: Ablation results for different hidden layer sizes.



(a) Kuhn poker (2p) (b) Kuhn poker (2p) (c) Leduc poker (2p) (d) Leduc poker (2p)

Figure 21: Ablation results for different train epochs.

in multi-player settings. This underscores the challenges of learning a robust strategy and training an accurate environment model in these complex games. Fig. 19 also illustrates the weights of BC in the BOMB algorithm under different datasets. In the four-player Kuhn poker game, the BC weight quickly drops to zero as the proportion of the random dataset increases, indicating that MB outperforms BC in most scenarios except when the random dataset proportion is low. Conversely, in the five-player Kuhn poker game, the BC weight remains high in most cases unless the random dataset proportion is very high. This is because the MB framework performs poorly in these highly dynamic multi-player games, making it a significant challenge to approximate equilibrium strategies with the MB method. These findings highlight the importance of improving both the strategy-learning process and the environment model in multi-player settings.

F.2 Ablation Study

To investigate the influence of hyperparameters, we conduct several ablation experiments on two-player Kuhn poker and Leduc poker games. We consider different model structures with various numbers of hidden layers. Specifically, for the 2-Player Kuhn poker game, we use different environment models with 16, 32, and 64 hidden layers. For the 2-Player Leduc poker game, which is a more complicated game, the numbers of hidden layers for different models are 32, 64, and 128. In addition, we train the environment models for different epochs to evaluate the robustness of our approach. Figs. 20-21 show these ablation results. We find that the number of hidden layers and the number of training epochs have little effect on the performance of the BC algorithm. These results further verify that the performance of the BC algorithm primarily depends on the quality of the dataset. As we know, the performance of the model-based method mainly depends on the trained environment model. Since the number of the hidden layer and the number of training epochs influence the training phase of the environment model, the number of the hidden layer and the number of train epochs have a slight impact on the performance of the model-based method. As long as the size of the hidden layer and the number of training epochs can guarantee that the environment model is trained accurately, the performance of the model-based method will not be affected.

F.3 PARAMETER SETTING

We list the parameters used to train the behavior cloning policy and environment model for all games used in our experiments in Tab. 2.

Methods	Behavior Cloning Algorithm				Environment Model Training			
Games	Kuhn Poker (2p)		Kuhn Poker (3p)		Kuhn Poker (2p)		Kuhn Poker (3p)	
Data size	500	1000	1000	5000	500	1000	1000	5000
Hidden layer	32	32	32	32	32	32	32	32
Batch size	32	32	32	32	32	32	32	32
Train epoch	1000	2000	5000	5000	1000	2000	2000	5000
Games	Kuhn Poker (4p)		Kuhn Poker (5p)		Kuhn Poker (4p)		Kuhn Poker (5p)	
Data size	5000	20000	10000	20000	5000	20000	10000	20000
Hidden layer	64	64	64	64	64	64	64	64
Batch size	64	128	128	128	64	128	128	128
Train epoch	5000	5000	5000	5000	5000	5000	5000	5000
Games	Leduc Poker (2p)		Leduc Poker (3p)		Leduc Poker (2p)		Leduc Poker (3p)	
Data size	10000	20000	10000	20000	10000	20000	10000	20000
Hidden layer	128	128	128	128	64	128	128	128
Batch size	128	128	128	128	64	128	128	128
Train epoch	10000	10000	10000	5000	10000	10000	10000	10000
Games	Liar's Dice		Phantom TTT		Liar's Dice		Phantom TTT	
Data size	10000	20000	10000	20000	10000	20000	10000	20000
Hidden layer	64	64	128	128	64	64	128	128
Batch size	128	128	128	128	64	128	128	128
Train epoch	5000	5000	5000	5000	5000	5000	5000	5000

Table 2: Parameters for Behavior Cloning algorithm and Environment Model Training