Near-Optimal Reinforcement Learning with Self-Play under Adaptivity Constraints

Dan Qiao [1] **Yu-Xiang Wang** [2]

Abstract

We study the problem of multi-agent reinforcement learning (MARL) with adaptivity constraints — a new problem motivated by real-world applications where deployments of new policies are costly and the number of policy updates must be minimized. For two-player zero-sum Markov Games, we design a (policy) elimination based algorithm that achieves a regret of $\widetilde{O}(\sqrt{H^3S^2ABK})$, while the batch complexity is only $O(H + \log\log K)$. In the above, S denotes the number of states, A, B are the number of actions for the two players respectively, H is the horizon and K is the number of episodes. Furthermore, we prove a batch complexity lower bound $\Omega(\frac{H}{\log_A K} + \log\log K)$ for all algorithms with $\widetilde{O}(\sqrt{K})$ regret bound, which matches our upper bound up to logarithmic factors. As a byproduct, our techniques naturally extend to learning bandit games and reward-free MARL within near optimal batch complexity. To the best of our knowledge, these are the first line of results towards understanding MARL with low adaptivity.

1. Introduction

This paper considers the problem of multi-agent reinforcement learning (multi-agent RL), where multiple agents aim to make decisions simultaneously in an unfamiliar environment to maximize their individual cumulative rewards. Multi-agent RL is used not only in large-scale strategy games like Go (Silver et al., 2017), Poker (Brown and Sandholm, 2019) and MOBA games (Ye et al., 2020), but also in various real-world applications such as autonomous driving (Shalev-Shwartz et al., 2016), household power management (Chung et al., 2020), and computer networking (Bhattacharyya et al., 2019).

The sheer amount of computation needed for self-play-based learning in these applications often demands the algorithm to run in a distributed fashion where the communication cost becomes a bottleneck. In such circumstances, instead of syncing up after each single trajectory, a more practical alternative is to assign a larger batch of work for each machine to perform independently and sync up only sporadically. The need for infrequent communication could be hard constraints in applications such as autonomous driving (Shalev-Shwartz et al., 2016). Deploying new policies to vehicle firmware takes weeks, while new data are being collected in millions of cars every second. These constraints render standard multi-agent RL algorithms that require altering the policy after each new data point impractical.

In the scenarios discussed above, the agent needs to minimize the number of policy deployments while learning a good policy using (nearly) the same number of samples as its fully-adaptive counterparts. Previous works brought up different notions to measure the adaptivity of an online RL algorithm, including switching cost (Bai et al., 2019; Zhang et al., 2020c; Qiao et al., 2022; Gao et al., 2021; Wang et al., 2021; He et al., 2023; Qiao et al., 2023; Kong et al., 2021; Velegkas et al., 2022; Zhao et al., 2023; Xiong et al., 2023), batch complexity (Perchet et al., 2016; Gao et al., 2019; Qiao et al., 2022; Zhang et al., 2022; Wang et al., 2021; Johnson et al., 2023; Xiong et al., 2023) and deployment efficiency (Matsushima et al., 2020; Huang et al., 2022b; Qiao and Wang, 2023; Modi et al., 2021). Although algorithms with low adaptivity have been designed for various Markov decision process (MDP) settings, all of the previous works focused on the single-agent setting while the solution to multi-agent RL with low adaptivity is still unknown. Therefore it is natural to question:

Question 1.1. Is it possible to design a self-play algorithm to solve Markov games while achieving near optimal adaptivity and sample complexity simultaneously?

Our contributions. In this paper, we answer the above question affirmatively by proposing a new low-adaptive

[1]Department of Computer Science, University of California, Santa Barbara (UCSB), USA [2]Halıcıoğlu Data Science Institute, University of California, San Diego (UCSD), USA. Correspondence to: Dan Qiao <danqiao@ucsb.edu>, Yu-Xiang Wang <yuxiangw@ucsd.edu>.

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

Algorithms for Markov games	*Single-agent (B=1)?*	*Regret*	*Batch complexity*
VI-ULCB (Bai and Jin, 2020)	No	$\widetilde{O}(\sqrt{H^3S^2ABT})$	K
Nash VI (Liu et al., 2021)	No	$\widetilde{O}(\sqrt{H^2SABT})$	K
Algorithm 1 in Zhang et al. (2022) *	Yes	$\widetilde{O}(\sqrt{H^2SAT})$	$O(H+\log\log K)$
Our Algorithm 1 (Theorem 4.1)	No	$\widetilde{O}(\sqrt{H^2S^2ABT})$	$O(H+\log\log K)$
Lower bound (Bai and Jin, 2020)	No	$\Omega(\sqrt{H^2S(A+B)T})$	No constraints.
Lower bound (Theorem 4.2)	No	if $\widetilde{O}(\sqrt{T})$ ("Optimal regret")	$\Omega(\frac{H}{\log_A K}+\log\log K)$
Algorithms for bandit games	*Single-agent (B=1)?*	*Regret*	*Batch complexity*
BaSE (Gao et al., 2019) †	Yes	$\widetilde{O}(\sqrt{AK})$	$O(\log\log K)$
Our Algorithm 6 (Theorem 5.1)	No	$\widetilde{O}(\sqrt{ABK})$	$O(\log\log K)$
Algorithms for reward-free exploration	*Single-agent (B=1)?*	*Sample (episode) complexity*	*Batch complexity*
VI-Explore (Bai and Jin, 2020)	No	$\widetilde{O}(\frac{H^5S^2AB}{\epsilon^2})$	$\widetilde{O}(\frac{H^7S^4AB}{\epsilon})^\ddagger$
LARFE (Qiao et al., 2022) *	Yes	$\widetilde{O}(\frac{H^5S^2A}{\epsilon^2})$	$O(H)$
Our Algorithm 4 (Theorem 5.2)	No	$\widetilde{O}(\frac{H^3S^2AB}{\epsilon^2})$	$O(H)$

Table 1. Comparison of our results (in blue) to existing work regarding problem type, regret/sample complexity, and batch complexity. A "Yes" in the column "Single-agent (B=1)?" means that the work is specific to the single-agent case ($B=1$) and cannot be directly applied under the two-player game setting ($B>1$) as in this paper. We list such works here for comparison. In the above, S denotes the number of states, A, B are the number of actions for the two players respectively, H is the horizon and K is the number of episodes ($T=HK$ is the number of steps). $*$: This result is derived under the special case of single-agent MDP (Markov game with $B=1$). In this case, our Algorithm 1 achieves the same batch complexity and a regret bound sub-optimal by $\sqrt{S}$. †: The result is derived under the batched multi-armed bandits setting (bandit games with $B=1$). In this case, our Algorithm 6 achieves the same guarantees as BaSE. ‡: For the first part of the algorithm, there are $\widetilde{O}(\frac{H^7S^4AB}{\epsilon})$ episodes of data collected using EULER, which can lead to the same number of batch complexity in the worst case. $\star$: This result is derived under the special case of single-agent MDP. Directly applying our Algorithm 4 to the setting will lead to a significant improvement of H^2 in sample complexity.

algorithm (Algorithm 1). Furthermore, the framework of Algorithm 1 naturally adapts to the bandit game setting and the more challenging low adaptive reward-free setting. Our concrete contributions are summarized as follows.

- We design a new policy elimination based algorithm (Algorithm 1) that achieves $O(H+\log\log K)$ batch complexity and $\widetilde{O}(\sqrt{H^2S^2ABT})$ regret bound (Theorem 4.1). To our knowledge, this provides the first result under multi-agent RL with low adaptivity. Moreover, the regret bound has optimal dependence on T while the batch complexity is optimal up to logarithmic factors among all algorithms with $\widetilde{O}(\sqrt{T})$ regret bound (due to our Theorem 4.2).
- Under the bandit game setting (a special case of Markov game with $H=S=1$), Algorithm 6 achieves $O(\log\log K)$ batch complexity and $\widetilde{O}(\sqrt{ABK})$ regret (Theorem 5.1). The batch complexity is optimal and the result strictly generalizes the best known result of batched multi-armed bandits (Gao et al., 2019).
- We also propose a new low-adaptive algorithm (Algorithm 4) for reward-free exploration in Markov games. It comes with an optimal batch complexity of $O(H)$ and provably identifies an ϵ-approximate Nash policy simultaneously for all possible reward functions (Theorem 5.2). The result improves over previous results in both sample complexity and batch complexity.

Related work. There is a large and growing body of literature on the statistical theory of reinforcement learning that we will not attempt to thoroughly review. Detailed comparisons with existing work on multi-agent RL (Bai and Jin, 2020; Liu et al., 2021), batched RL (Zhang et al., 2022), batched multi-armed bandits (Gao et al., 2019) and reward-free exploration (Bai and Jin, 2020; Qiao et al., 2022) are given in Table 1. For more details about related works, please refer to Appendix A and the references therein. Notably, all existing algorithms with low adaptivity focus on the single-agent case. In comparison, our results work for the more general multi-player setting, and thus can be considered as generalization of previous results.

A recent line of works provide non-asymptotic guarantees for learning Markov Games. Bai and Jin (2020) developed the first provably-efficient algorithms in MGs based on optimistic value iteration, whose result is improved by Liu et al. (2021) using model-based approach. Meanwhile, model-free approaches are shown to break the curse of multiagency and improve the dependence on action space (Bai et al., 2020; Jin et al., 2021; Mao et al., 2022; Wang et al., 2023; Cui et al., 2023). Following works also extended the results to MGs with function approximation (Xie et al.,

2020; Huang et al., 2022a; Jin et al., 2022; Li et al., 2022). However, all these works applied fully adaptive algorithms, which can be difficult to implement in practical scenarios. In comparison, our algorithms achieve near optimal adaptivity.

In this paper, we measure the adaptivity by batch complexity (Zhang et al., 2022) which requires decisions about policy updates to be made at only a few predefined checkpoints.[1] Besides, there are other measurements of adaptivity. The most prevalent measurement is *switching cost*, which measures the number of policy switches. However, previous works minimizing switching cost only allow usage of deterministic policies (Bai et al., 2019; Zhang et al., 2020c; Qiao et al., 2022). It is known that in many Markov games like Rock paper scissors, the Nash policy can only be stochastic and running deterministic policies will lead to a linear regret. Therefore, switching cost is not an appropriate measurement for the multi-agent case. [2] Meanwhile, Matsushima et al. (2020) proposed the notion of *deployment efficiency*, which is similar to batched RL with additional requirement that each policy deployment should have similar size. Deployment efficient RL is studied by some following works (Huang et al., 2022b; Qiao and Wang, 2023; Modi et al., 2021). However, as pointed out by Qiao and Wang (2023), deployment complexity is not a good measurement of adaptivity when studying regret minimization.

2. Problem Setup

Notations. Throughout the paper, for $n \in \mathbb{Z}^+$, $[n] = \{1, 2, \cdots, n\}$. For any set U, $\Delta(U)$ denotes the set of all possible distributions over U. In addition, we use standard notations such as O and Ω to absorb constants while $\widetilde{O}$ and $\widetilde{\Omega}$ suppress logarithmic factors.

Markov Games. Markov Games (MGs) generalize the standard Markov Decision Processes (MDPs) into the multi-player setting, where each player aims to maximize her own reward. We consider *two-player zero-sum* episodic Markov Games, denoted by a tuple $\mathcal{MG} = (H, \mathcal{S}, \mathcal{A}, \mathcal{B}, P_h, r_h)$, where H is the horizon, $\mathcal{S}$ is the state space with $S := |\mathcal{S}|$. $\mathcal{A}$ and $\mathcal{B}$ are the action space for the max-player (who aims to maximize the total reward) and the min-player (who aims to minimize the total reward) respectively, where $A := |\mathcal{A}|, B := |\mathcal{B}|$ are finite. The non-stationary transition kernel has the form $P_h : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times \mathcal{S} \mapsto [0, 1]$ with $P_h(s'|s, a, b)$ representing the probability of transition from state s, action (a, b) to next state s' at time step h. In addition, $r_h(s, a, b)$ denotes the known[3] expected (immediate) reward function. Without loss of generality, we assume each episode starts from a fixed initial state s_1.[4] At time step $h \in [H]$, two players observe s_h and choose their actions $a_h \in \mathcal{A}$ and $b_h \in \mathcal{B}$ at the same time. Then both players observe the action of their opponent and receive reward $r_h(s_h, a_h, b_h)$, the environment will transit to $s_{h+1} \sim P_h(\cdot|s_h, a_h, b_h)$.

Markov policy, value function. A (Markov) policy μ of the max-player can be seen as a series of mappings $\mu = (\mu_1, \cdots, \mu_H)$, where each μ_h maps each state $s \in \mathcal{S}$ to a probability distribution over actions $\mathcal{A}$, *i.e.* $\mu_h : \mathcal{S} \to \Delta(\mathcal{A})$. A Markov policy ν for the min-player is defined similarly.

Given a pair of policies (μ, ν) and $h \in [H]$, the value function $V_h^{\mu,\nu}(\cdot)$ and Q-value function $Q_h^{\mu,\nu}(\cdot, \cdot, \cdot)$ are defined as: $V_h^{\mu,\nu}(s) = \mathbb{E}_{\mu\times\nu}[\sum_{t=h}^{H} r_t | s_h = s], Q_h^{\mu,\nu}(s, a, b) = \mathbb{E}_{\mu\times\nu}[\sum_{t=h}^{H} r_t | s_h, a_h, b_h = s, a, b], \ \forall\, s, a, b \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}$. Then the Bellman equation follows $\forall\, h \in [H]$:

$$Q_h^{\mu,\nu}(s, a, b) = [r_h + P_h V_{h+1}^{\mu,\nu}](s, a, b),$$
$$V_h^{\mu,\nu}(s) = [\mathbb{E}_{\mu\times\nu} Q_h^{\mu,\nu}](s).$$

In this work, we will consider different MGs with respective transition kernels and reward functions. We define the value function for policy $\pi = (\mu, \nu)$ under MG $(\widetilde{r}, \widetilde{P})$ as below

$$V^{\pi}(\widetilde{r}, \widetilde{P}) = \mathbb{E}_{\pi}\left[\sum_{h=1}^{H} \widetilde{r}_h \mid \widetilde{P}\right].$$

Best responses, Nash equilibrium. For any policy μ of the max-player, there exists a best response policy $\nu^\dagger(\mu)$ of the min-player such that $V_h^{\mu,\nu^\dagger(\mu)}(s) = \inf_\nu V_h^{\mu,\nu}(s)$ for all $(s, h) \in \mathcal{S} \times [H]$. For simplicity, we denote $V_h^{\mu,\dagger} := V_h^{\mu,\nu^\dagger(\mu)}$. Also, $\mu^\dagger(\nu)$ and $V_h^{\dagger,\nu}$ can be defined similarly. It is shown (Filar and Vrieze, 2012) that there exists a pair of policies $(\mu^\star, \nu^\star)$ that are best responses against each other, *i.e.* for all $(s, h) \in \mathcal{S} \times [H]$,

$$V_h^{\mu^\star,\dagger}(s) = V_h^{\mu^\star,\nu^\star}(s) = V_h^{\dagger,\nu^\star}(s).$$

We call the pair of policies Nash equilibrium of the Markov game, which further satisfies the following minimax property: for all $(s, h) \in \mathcal{S} \times [H]$,

$$\sup_\mu \inf_\nu V_h^{\mu,\nu}(s) = V_h^{\mu^\star,\nu^\star}(s) = \inf_\nu \sup_\mu V_h^{\mu,\nu}(s).$$

The value functions of $(\mu^\star, \nu^\star)$ are called Nash value functions and we denote $V_h^\star = V_h^{\mu^\star,\nu^\star}, Q_h^\star = Q_h^{\mu^\star,\nu^\star}$ for simplicity. Intuitively speaking, Nash equilibrium means that no player could benefit from switching her own policy.

[1] The formal def. of batch complexity is deferred to Section 2.

[2] While it is true that we can generalize the definition of switching cost to support general (stochastic) policies, in this case the guarantee of batched RL is stronger due to its nature that the timestep to switch policy is predetermined.

[3] Our results easily generalize to the case with stochastic reward.

[4] The generalized case where the initial distribution is an arbitrary distribution can be recovered from this setting by adding one layer to the MG.

Non-Markov policies. In this work, we consider general, history-dependent policies that may not be Markov policies. A general policy μ of the max-player is a set of mappings $\mu = \{\mu_h : \Omega \times (\mathcal{S} \times \mathcal{A} \times \mathcal{B} \times \mathbb{R})^{h-1} \times \mathcal{S} \to \Delta(\mathcal{A})\}$. The choice of the action at time step $h \in [H]$ depends on the history and a random sample $w \in \Omega$ that is shared among all time steps. A special case of general policies is a mixture of Markov policies, which will be used in this work. General policies for the min-player, the best response of a general policy can be defined similarly as Markov policies. Note that the best response to a non-Markov policy may be non-Markov.

Learning objective: regret. Suppose K is the number of episodes the agent plan to play and μ^k is the policy executed by the max-player in the k-th episode. Then the regret of the max-player is defined as

$$\text{Regret}(K) := \sum_{k=1}^{K} [V_1^\star(s_1) - V_1^{\mu^k,\dagger}(s_1)].$$

Same as Jin et al. (2022), our goal is to minimize the regret of the max-player, which focuses on learning the Nash policy of the max-player. By symmetry, the techniques readily extend to learning the Nash policy of the min-player.

Batched reinforcement learning. We measure the adaptivity of an online RL algorithm by batch complexity.

Definition 2.1. We say an algorithm has batch complexity M, if the algorithm pre-determines a group of lengths $\{T_i\}_{i\in[M]}$ such that $\sum_{i=1}^{M} T_i = K$. At the beginning of the i-th batch, the agent determines a general policy $\pi^i = (\mu^i, \nu^i)$ and follows π^i for T_i episodes.

Our goal is to minimize the batch complexity of our algorithm while achieving near-optimal regret.

Remark 2.2. We highlight that our setting strictly generalizes the batched (single-agent) RL setting in Qiao et al. (2022); Zhang et al. (2022). This is because when the min-player plays a fixed and known policy, the definitions of MG, regret and batch complexity will reduce to the single-agent RL setting. Therefore, our setting is more complex and technically demanding by incorporating the min-player.

3. Main algorithms

In order to attain a batch complexity that is near-optimal, we extend the batch schedule derived from the arm-elimination algorithm for bandits (Cesa-Bianchi et al., 2013) to an elimination based algorithm for RL. The core concept behind our policy elimination approach revolves around maintaining a *version space* Π_A that encompasses the remaining policies for the max-player. In each batch, we enhance the estimated values of all policies within Π_A and utilize these values to eliminate policies that cannot possibly be Nash policies. The overarching goal is that as the algorithm progresses, the policies remaining after elimination are already in approximate alignment with Nash equilibrium.

Algorithm 1 Main algorithm

Require: Number of episodes K. The known reward r. Universal constants C, N. Failure probability δ.
Initialize: $T^{(k)} = K^{1-\frac{1}{2^k}}$, $k \le K_0 = O(\log\log K)$, $\Pi_A^1 :=$ {All Markov policies for the max-player}, $\Pi_B :=$ {All Markov policies for the min-player}, $\iota = \log(2HSABK/\delta)$.
for $k = 1, 2, \cdots, K_0$ **do**
  ◇ Number of episodes in k-th stage:
  if $(N+1)T^{(1)} + (1+(k-1)N)T^{(2)} + \sum_{i=1}^{k} T^{(i)} \ge K$ **then**
    $T^{(k)} = K - (N+1)T^{(1)} - (1+(k-1)N)T^{(2)} - \sum_{i=1}^{k-1} T^{(i)}$ (o.w. $T^{(k)} = K^{1-\frac{1}{2^k}}$).
  end if
end for
for $k = 1, 2$ **do**
  ◇ Update the infrequent set $\mathcal{F}$ and construct an empirical estimate $\widehat{P}$ of the absorbing MG:
  $\mathcal{F}^k, P^{int,k}, \pi_k$ = Crude Exploration$(\Pi_A^k, \Pi_B, T^{(k)})$.
  $\widehat{P}^k$ = Fine Exploration$(\mathcal{F}^k, P^{int,k}, \Pi_A^k, \Pi_B, T^{(k)}, \pi_k, NT^{(k)})$.
  ◇ Policy elimination for the max-player:
  $\Pi_A^{k+1} \leftarrow \{\mu \in \Pi_A^k \mid \inf_{\nu\in\Pi_B} V^{\mu,\nu}(r, \widehat{P}^k) \ge \sup_{\mu\in\Pi_A^k} \inf_{\nu\in\Pi_B} V^{\mu,\nu}(r, \widehat{P}^k) - 2C(\sqrt{\frac{H^3S^2AB\iota}{T^{(k)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(k)}})\}$.
end for
for $k = 3, 4, \cdots, K_0$ **do**
  ◇ Only update the empirical transition kernel:
  $\widehat{P}^k$ = Fine Exploration$(\mathcal{F}^2, P^{int,2}, \Pi_A^k, \Pi_B, T^{(k)}, \pi_2, NT^{(2)})$.
  $\Pi_A^{k+1} \leftarrow \{\mu \in \Pi_A^k \mid \inf_{\nu\in\Pi_B} V^{\mu,\nu}(r, \widehat{P}^k) \ge \sup_{\mu\in\Pi_A^k} \inf_{\nu\in\Pi_B} V^{\mu,\nu}(r, \widehat{P}^k) - 2C(\sqrt{\frac{H^3S^2AB\iota}{T^{(k)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(2)}})\}$.
end for

The primary hurdle lies in efficiently estimating the value function for all policy pairs with minimal sample usage. The challenge of uniform convergence typically involves estimating transition kernels, compounded by the necessity to address an exploration problem to visit specific state-action pairs at least once. Additionally, certain states may not be frequently visited by any policy pair. To tackle these challenges, we construct a surrogate Markov Game termed an "absorbing MG" featuring an absorbing state. This absorbing MG replaces problematic states with an absorbing state, denoted as $s^\dagger$. Consequently, all remaining states can be adequately visited by some policy in $\Pi_A \times \Pi_B$. Furthermore, its value function uniformly approximates the original MG

Algorithm 2 Crude Exploration

Input: Policy sets Π_A (max) and Π_B (min). Number of episodes T. Universal constant C_1.
Initialize: $T_0 = \frac{T}{H}$, $\mathcal{F} = \emptyset$, $\mathcal{D} = \emptyset$, $\iota = \log(2HSABK/\delta)$. $1_{h,s,a,b}$ is a reward function r' where $r'_{h'}(s', a', b') = \mathbb{1}[(h', s', a', b') = (h, s, a, b)]$. $s^\dagger$ is an additional absorbing state. P^{int} is a transition kernel over the extended space $\mathcal{S} \cup \{s^\dagger\} \times \mathcal{A} \times \mathcal{B}$, initialized *arbitrarily*.
Output: Infrequent tuples $\mathcal{F}$. Intermediate transition kernel P^{int}. A uniformly explorative policy π.
for $h = 1, 2, \cdots, H$ **do**
  ◇ Construct and run policies to visit each state-action:
  for $(s, a, b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}$ **do**
    $\pi_{h,s,a,b} = \text{argmax}_{(\mu,\nu)\in\Pi_A\times\Pi_B} V^{\mu,\nu}(1_{h,s,a,b}, P^{int})$.
  end for
  Run $\pi_h =$ uniform mixture of $\{\pi_{h,s,a,b}\}_{(s,a,b)}$ for T_0 episodes, and add the trajectories into data set $\mathcal{D}$.
  for $(s, a, b, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times \mathcal{S}$ **do**
    $N_h(s, a, b, s') =$ count of (h, s, a, b, s') in $\mathcal{D}$.
  end for
  ◇ Use $\mathcal{F}$ to store the infrequent tuples:
  $\mathcal{F} = \mathcal{F} \cup \{(h, s, a, b, s') | N_h(s, a, b, s') \le C_1 H^2 \iota\}$.
  ◇ Update the intermediate transition kernel using Algorithm 5:
  $P^{int} = \text{EstimateTransition}(\mathcal{D}, \mathcal{F}, s^\dagger, h, P^{int})$
  Reset data set $\mathcal{D} = \emptyset$.
end for
◇ Construct a uniformly explorative policy:
Policy $\pi \leftarrow$ uniform mixture of $\{\pi_{h,s,a,b}\}_{(h,s,a,b)}$.
Return: $\{\mathcal{F}, P^{int}, \pi\}$.

Algorithm 3 Fine Exploration

Input: Infrequent tuples $\mathcal{F}$. Intermediate transition kernel P^{int}. Policy sets Π_A and Π_B. Number of episodes T. Auxiliary policy π' with number of episodes T'.
Initialize: $\mathcal{D} = \emptyset$. $\widehat{P} = P^{int}$. $1_{h,s,a,b}$ is a reward function r' where $r'_{h'}(s', a', b') = \mathbb{1}[(h', s', a', b') = (h, s, a, b)]$.
Output: Empirical estimate $\widehat{P}$.
Construct explorative policy π according to (1).
Run π for T episodes and run π' for T' episodes, add all the trajectories into data set $\mathcal{D}$.
◇ Construct an empirical estimate for $\widetilde{P}$ using Algorithm 5:
for $h \in [H]$ **do**
  $\widehat{P} = \text{EstimateTransition}(\mathcal{D}, \mathcal{F}, s^\dagger, h, \widehat{P})$.
end for
Return $\widehat{P}$.

for all relevant policies, thereby simplifying the problem to estimating the transition kernel of the absorbing MG.

Main algorithm. Given a budget of K episodes, Algorithm 1 divides it into multiple stages with increasing length $T^{(k)} := K^{1-1/2^k}$ for $k = 1, 2, 3, ..., K_0$. It is known (Cesa-Bianchi et al., 2013) that the total number of stages K_0 is bounded by $O(\log \log K)$.

In Algorithm 1, the first two stages involve three steps while all the following stages only contain the last two steps. Before we introduce the steps, we remark that such design is due to technical reasons. First, two implementations of Crude exploration could give an "absorbing" MG model that is sufficient for our purpose. In addition, to achieve the optimal batch complexity, the number of Crude exploraion's can only be a constant. Therefore, we give up the cleaner schedule where each stage contains all the three steps and choose the current schedule. Below is the three steps:

Step 1. Crude exploration Explore each (h, s, a, b) pair layer-by-layer using policies from the current version-space Π_A^k (and Π_B). Establish an absorbing MG $\widetilde{P}$ and a *crude* intermediate estimate (P^{int}) of $\widetilde{P}$.

Step 2. Fine exploration Solve for a uniformly explorative policy to explore all tuples based on the crude estimate (P^{int}) of the absorbing MG. Construct a more accurate estimate ($\widehat{P}$) of the absorbing MG $\widetilde{P}$.

Step 3. Policy elimination Evaluate all policies in Π_A using $\inf_{\nu\in\Pi_B} V^{\mu,\nu}(r, \widehat{P})$ as an estimate of $V^{\mu,\dagger}(r, P)$. Update the version space Π_A by eliminating all policies whose empirical value $\inf_{\nu\in\Pi_B} V^{\mu,\nu}(r, \widehat{P})$ is suboptimal by a gap depending on the batch size.

As the algorithm advances, assuming the high probability event that our confidence bounds hold, the Nash policy of the max-player is unlikely to be eliminated. At each stage, the remaining policies consistently outperform the LCB of the Nash policy, which will converge to the Nash value.

Next, we explain the first two steps of Algorithm 1: Crude Exploration (Alg. 2) and Fine Exploration (Alg. 3).

Layer-by-layer Exploration in Algorithm 2. Our Algorithm 2 is a generalization of Algorithm 2 of Qiao et al. (2022) to the multi-agent case. The motivation is that if we visit some tuple (h, s, a, b, s') for $O(H^2\iota)$ times, the empirical estimate of $P_h(s'|s, a, b)$ would be multiplicatively accurate. Therefore, the challenge would be to visit each tuple as much as possible using policies from the input policy sets Π_A and Π_B. However, some (h, s, a, b, s') tuples may be hard to reach by any policy in the policy set. To tackle this issue, we construct the set $\mathcal{F}$ to be all the tuples that do not have enough visitations. Consequently, for the tuples not in $\mathcal{F}$, the empirical estimate of transition is accurate enough, while for the tuples in $\mathcal{F}$, we will prove that no policy in the policy set could visit the tuple frequently thus

they have little influence on the value function.

More specifically, we explore the MG layer-by-layer. When exploring the h-th layer, we construct $\pi_{h,s,a,b}$'s to be the greedy policies under P^{int}, *i.e.* $\pi_{h,s,a,b}$ could visit (h,s,a,b) with the largest probability under P^{int}. Then we run a uniform mixture of $\{\pi_{h,s,a,b}\}_{(s,a,b)}$ (which is a general policy) for several episodes to collect the samples $\mathcal{D}$, after which we update the h-th layer of P^{int} (i.e. P^{int}_h) using Algorithm 5 and $\mathcal{D}$. Finally, a uniformly explorative policy π that can visit all tuples is stored for future use.

Based on infrequent tuples $\mathcal{F}$, the absorbing MG is established according to Definition 3.1. The construction is equivalent to first letting $\widetilde{P} = P$, and then moving the probability of $\widetilde{P}_h(s'|s,a,b)$ to $\widetilde{P}_h(s^\dagger|s,a,b)$ for $(h,s,a,b,s') \in \mathcal{F}$.

Definition 3.1 (The absorbing MG $\widetilde{P}$)**.** Given $\mathcal{F}$ and P, $\forall\,(h,s,a,b,s') \notin \mathcal{F}$, let $\widetilde{P}_h(s'|s,a,b) = P_h(s'|s,a,b)$. For any $(h,s,a,b,s') \in \mathcal{F}$, $\widetilde{P}_h(s'|s,a,b) = 0$. For any $(h,s,a,b) \in [H]\times\mathcal{S}\times\mathcal{A}\times\mathcal{B}$, $\widetilde{P}_h(s^\dagger|s^\dagger,a,b) = 1$ and

$$\widetilde{P}_h(s^\dagger|s,a,b) = 1 - \sum_{s'\in\mathcal{S}:(h,s,a,b,s')\notin\mathcal{F}} \widetilde{P}_h(s'|s,a,b).$$

Remark 3.2. Such absorbing structure has been applied under the single-agent case (Qiao et al., 2022; Zhang et al., 2022; Zhang and Zanette, 2023), and here we generalize the definition to the multi-agent case. It will be shown that with high probability, for (h,s,a,b,s'), either $(1-\frac{1}{H})P^{int}_h(s'|s,a,b) \le \widetilde{P}_h(s'|s,a,b) \le (1+\frac{1}{H})P^{int}_h(s'|s,a,b)$ or $P^{int}_h(s'|s,a,b) = \widetilde{P}_h(s'|s,a,b) = 0$. Therefore, $\pi_{h,s,a,b}$'s are efficient in exploration.

Algorithm 5 and corresponding explanations are deferred to Appendix B due to space limit. Here we remark that P^{int} is the empirical estimate of $\widetilde{P}$ using the samples in $\mathcal{D}$. Besides, we discuss about the transition between the original MG and the absorbing MG in Appendix C.

Fine exploration by Algorithm 3. The key behind Algorithm 3 is that with high probability, $V^{\mu,\nu}(1_{h,s,a,b},P^{int})$ is multiplicatively close to $V^{\mu,\nu}(1_{h,s,a,b},\widetilde{P})$ for all (h,s,a,b) and remaining (μ,ν). Therefore, P^{int} can be used to guide the exploration. We construct the exploration policy π as:

$$\pi = \mathrm{argmin}_{\pi\in\Delta(\Pi_A\times\Pi_B)} \sup_{(\mu',\nu')\in\Pi_A\times\Pi_B} \sum_{h=1}^{H}\sum_{s,a,b} \frac{V^{\mu',\nu'}(1_{h,s,a,b},P^{int})}{V^{\pi}(1_{h,s,a,b},P^{int})}, \tag{1}$$

which is a mixture of several policies from the current version space that could provide uniform coverage of all remaining policies. Then we run both π (which is a general policy) and an auxiliary policy π'[5] to collect samples. At last, $\widehat{P}$ is calculated as an empirical estimate of $\widetilde{P}$ by using Algorithm 5 and the data set $\mathcal{D}$.

Details about the implementation and computational efficiency of the algorithms are deferred to Section 4.

4. Main results

In this section, we will state our main results, which formalize the algorithmic ideas explained in the previous section.

Theorem 4.1 (Regret and batch complexity of Algorithm 1)**.** *With probability $1-\delta$, Algorithm 1 will have regret bounded by $\widetilde{O}(\sqrt{H^2S^2AB\mathcal{T}})$, where $\mathcal{T} := KH$ is the number of steps. Furthermore, the batch complexity of Algorithm 1 is bounded by $O(H+\log\log K)$.*

Theorem 4.1 says that Algorithm 1 is able to achieve a regret bound with optimal dependence on K while using only $O(\log\log K)$ batches. Due to space limit, the proof is sketched in Section 6 with details in the Appendix. Now we discuss a few interesting aspects of the result.

Near optimal batch complexity. We state the following lower bound of batch complexity for all algorithms with $\widetilde{O}(\sqrt{K})$ regret bound, which implies that the batch complexity of our Algorithm 1 is nearly optimal.

Theorem 4.2 (Lower bound)**.** *For any algorithm with $O(\mathrm{poly}(S,A,B,H)\sqrt{K})$ regret bound, the batch complexity is at least $\Omega(H/\log_A K + \log\log K)$.*

Due to space limit, the proof is deferred to Appendix F.

Transformation to PAC guarantee. Various MARL applications require outputting a near optimal policy at the end of the algorithm. Below we provide a sample complexity upper bound for finding an ϵ-**approximate Nash** policy for the max-player. We highlight that although we run general policies in the algorithm, the output policy can be a single Markov policy that is convenient to store and execute.

Theorem 4.3 (Sample complexity)**.** *For any $\epsilon>0$ and $\delta>0$, with probability $1-\delta$, Algorithm 1 could output a **Markov** policy μ of the max-player in $\widetilde{O}\left(\frac{H^3S^2AB}{\epsilon^2}\right)$ episodes such that μ is ϵ-approximate Nash,* i.e.

$$V^\star(r,P) - V^{\mu,\dagger}(r,P) \le \epsilon.$$

The proof is deferred to Appendix F. By symmetry, an ϵ-approximate Markov Nash policy for the min-player can be found using the same number of episodes.

Dependence on H,S,A,B in the regret. Our regret bound is optimal in T. However, there is a gap of $\sqrt{S\min\{A,B\}}$ when compared to the information-theoretic limit of

[5] Running a uniformly explorative policy is for technical reason.

$\Omega(\sqrt{H^2S(A+B)T})$ that covers all algorithms (including those without adaptivity constraints) (Bai and Jin, 2020). When applying our Algorithm 1 to the single-agent MDP setting where the min-player plays a fixed and known policy ($B = 1$), the regret bound and batch complexity will be $\widetilde{O}(\sqrt{H^2S^2AT})$ and $O(H + \log\log K)$, respectively. The batch complexity is known to be optimal (Zhang et al., 2022) while the regret is suboptimal by $\sqrt{S}$ when compared to the lower bound of $\Omega(\sqrt{H^2SAT})$ (Jin et al., 2018). We believe our analysis is tight and further improvements on S will require new algorithmic ideas to handle the larger policy space compared to the single agent case. It is an intriguing open problem whether we can design an algorithm with both optimal batch complexity and optimal regret bound.

Computational efficiency. Our Main Algorithm (Algorithm 1) is not computationally efficient in general, since the algorithm needs to explicitly go over the remaining policy set to implement policy elimination, construct policy $\pi_{h,s,a,b}$'s in Crude exploration and π in Fine exploration. These steps can be solved approximately in exponential time by enumerating over a tight covering set of Markov policies. For efficient surrogate of Algorithm 1, we remark that a possible method is to apply softmax (or other differentiable) representation of the policy space and use gradient-based optimization techniques to find approximate solutions.

5. Some discussions

5.1. Application to bandit games ($H = S = 1$)

If for a two-player zero-sum Markov game, there is no multiple time steps and no states ($H = S = 1$), then the Markov game reduces to a two-player zero-sum bandit game, which has wide real-world applications. According to Theorem 4.1, a direct application of Algorithm 1 under the bandit setting will lead to a regret bound of $\widetilde{O}(\sqrt{ABK})$ and a batch complexity bound of $O(\log\log K)$. Moreover, with some mild revision to Algorithm 1, the implementation can be computationally efficient. The result is summarized in Theorem 5.1 below. Note that the definition of batch complexity and regret is identical to the Markov game case.

Theorem 5.1. *If we run Algorithm 6 (adapted from Algorithm 1) under a two-player zero-sum bandit game for K episodes, the batch complexity is bounded by $O(\log\log K)$ while the regret is $\widetilde{O}(\sqrt{ABK})$ with high probability. Furthermore, the algorithm is computationally efficient.*

Due to space limit, Algorithm 6 and proof of Theorem 5.1 are deferred to Appendix G.1. To the best of our knowledge, Theorem 5.1 is the first result for learning multi-player bandit games with low adaptivity, and the batch complexity is optimal while the regret has optimal dependence on K. The special case of bandit games where $B = 1$ is the well-studied multi-armed bandits setting (Auer et al., 2002). Previous works (Perchet et al., 2016; Gao et al., 2019) designed algorithms for the MAB setting which achieve $O(\log\log K)$ batch complexity and $\widetilde{O}(\sqrt{AK})$ regret simultaneously, where both bounds are shown to be minimax optimal. In comparison, the above upper bounds can be achieved by directly plugging in $B = 1$ to Theorem 5.1. Therefore, our result can be considered as a provably efficient generalization of Gao et al. (2019) to the game setting.

5.2. Extension to the reward-free case

In this part, we further consider the setting of low adaptive reward-free exploration (Jin et al., 2020), where the goal is to output a near Nash policy for any possible reward function. Specifically, due to its nature that Crude Exploration (Algorithm 2) and Fine Exploration (Algorithm 3) do not use any information about the reward function r, these two algorithms can be leveraged in reward-free setting. Algorithm 4 uses Crude Exploration to construct the infrequent tuples $\mathcal{F}$ and the intermediate MG P^{int}. Then the algorithm uses Fine Exploration to get an empirical estimate $\widehat{P}$ of the absorbing MG $\widetilde{P}$. At last, for any reward function r, the algorithm outputs the Nash policy under the empirical MG.

Algorithm 4 Algorithm for reward-free case

1: **Input**: Episodes for crude exploration N_0, episodes for fine exploration N_1. Universal constant N. Failure probability δ.
2: **Initialize**: $\iota = \log(2HSAB(N_0+N_1)/\delta)$, $\Pi_A := \{\text{All Markov policies for the max-player}\}$, $\Pi_B := \{\text{All Markov policies for the min-player}\}$.
3: **Output**: $\pi^r = (\mu^r, \nu^r)$ for any reward function r.
4: $\mathcal{F}$, P^{int}, $\overline{\pi}$ = Crude Exploration(Π_A, Π_B, N_0).
5: $\widehat{P}$ = Fine Exploration$(\mathcal{F}, P^{int}, \Pi_A, \Pi_B, N_1, \overline{\pi}, NN_0)$.
6: $\mu^r = \mathrm{argmax}_{\mu\in\Pi_A} \inf_{\nu\in\Pi_B} V^{\mu,\nu}(r, \widehat{P})$ for any r.
7: $\nu^r = \mathrm{argmin}_{\nu\in\Pi_B} \sup_{\mu\in\Pi_A} V^{\mu,\nu}(r, \widehat{P})$ for any r.
8: **Return** $\{\pi^r = (\mu^r, \nu^r)\}_r$.

The batch complexity and sample complexity of Algorithm 4 is summarized in Theorem 5.2 below (whose proof is deferred to Appendix G.2 due to space limit).

Theorem 5.2. *The batch complexity of Algorithm 4 is bounded by $H + 2$. There exists a constant $c > 0$ such that, for any $\epsilon > 0$ and any $\delta > 0$, if the number of total episodes K satisfies that*

$$K > c\left(\frac{H^3S^2AB\iota'}{\epsilon^2} + \frac{H^5S^3A^2B^2\iota'}{\epsilon}\right),$$

where $\iota' = \log(\frac{HSAB}{\epsilon\delta})$, then there exists a choice of N_0 and N_1 such that $N_0 + N_1 = K$ and with probability $1-\delta$, for any reward function r, Algorithm 4 will output a policy pair $\pi^r = (\mu^r, \nu^r)$ that is ϵ-approximate Nash.

The batch complexity of Algorithm 4 is known to be optimal up to logarithmic factors (Huang et al., 2022b) while the sample complexity is optimal in ϵ (Jin et al., 2020). Moreover, when applied to the special case of single-agent MDP, our sample complexity is $\widetilde{O}(\frac{H^3S^2A}{\epsilon^2})$, which matches the best known results even without adaptivity constraints (Ménard et al., 2021; Zhang et al., 2020b). We highlight that our sample complexity improves significantly over previous algorithms with near optimal adaptivity (Qiao et al., 2022; Qiao and Wang, 2023), whose sample complexities are both $\widetilde{O}(\frac{H^5S^2A}{\epsilon^2})$. Last but not least, compared to Algorithm 1, our Algorithm 4 could output near Nash policies for both players simultaneously by running a single algorithm.

6. Proof overview

In this section, we summarize the proof of Theorem 4.1. The analysis contains two main parts: the batch complexity bound and the regret bound. The batch complexity bound can be directly derived from the schedule of Algorithm 1.

Upper bound for batch complexity. Crude exploration (Algorithm 2) can be run in H batches while Fine exploration (Algorithm 3) can be run in 2 batches. Since Algorithm 1 contains 2 Crude exploration's and $O(\log\log K)$ Fine exploration's, we have the conclusion that the batch complexity of Algorithm 1 is bounded by $O(H+\log\log K)$.

However, such an elimination schedule requires the algorithm to run the same exploration policy for a large number of episodes before being able to switch to another policy, which is the main technical hurdle to the regret analysis.

Regret upper bound. The core of the regret analysis is to construct a uniform off-policy evaluation bound that covers all remaining policies. The remaining policy set (for the max-player) at the beginning of stage k is Π_A^k. Assume with high probability, for all $(\mu,\nu)\in\Pi_A^k\times\Pi_B$, we can estimate $V^{\mu,\nu}(r,P)$ to ϵ_k accuracy using $\widehat{P}^k$, then the difference between $\inf_{\nu\in\Pi_B}V^{\mu,\nu}(r,\widehat{P}^k)$ and $V^{\mu,\dagger}(r,P)$ is bounded by ϵ_k uniformly for all $\mu\in\Pi_A^k$. Therefore if we eliminate all policies μ that are at least $2\epsilon_k$ sub-optimal in the sense of $\inf_{\nu\in\Pi_B}V^{\mu,\nu}(r,\widehat{P}^k)$, the Nash policy will not be eliminated and all the policies remaining will be $4\epsilon_k$-approximate Nash. Summing up the regret of all stages, we have with high probability, the total regret is bounded by

$$\text{Regret}(K)\le O(HT^{(1)})+O\left(\sum_{k=2}^{K_0}T^{(k)}\times\epsilon_{k-1}\right). \quad (2)$$

The following key lemma provides an upper bound of ϵ_{k-1} given that we use the empirical transition kernel $\widehat{P}^k$ in Algorithm 1 to estimate $V^{\mu,\nu}(r,P)$.

Lemma 6.1. *W.h.p, for any k and $(\mu,\nu)\in\Pi_A^k\times\Pi_B$,*

$$\left|V^{\mu,\nu}(r,\widehat{P}^k)-V^{\mu,\nu}(r,P)\right|\le\widetilde{O}\left(\sqrt{\frac{H^3S^2AB}{T^{(k)}}}\right).$$

The proof of Lemma 6.1 requires controlling both the "bias" and "variance" part of estimation error. The "bias" refers to the difference between the true MG and the absorbing MG, while the "variance" refers to the statistical error in estimating the value functions of the absorbing MG using $\widehat{P}^k$. For simplicity, in the following discussion, we omit the stage number k and the discussion holds true for all k.

The "bias": difference between P and $\widetilde{P}$. First, it holds that if the visitation number of a tuple (h,s,a,b,s') is larger than $O(H^2\iota)$, with high probability,

$$\begin{aligned}(1-\frac{1}{H})P_h^{int}(s'|s,a,b)&\le\widetilde{P}_h(s'|s,a,b)\\&\le(1+\frac{1}{H})P_h^{int}(s'|s,a,b).\end{aligned}$$

According to the definition of $\mathcal{F}$ in Algorithm 2, we have the above inequality is true for any (h,s,a,b,s'). Then we have for any (h,s,a,b) and any policy $(\mu,\nu)\in\Pi_A\times\Pi_B$,

$$\begin{aligned}\frac{1}{4}V^{\mu,\nu}(1_{h,s,a,b},P^{int})&\le V^{\mu,\nu}(1_{h,s,a,b},\widetilde{P})\\&\le 3V^{\mu,\nu}(1_{h,s,a,b},P^{int}).\end{aligned}$$

Due to the construction of $\pi_{h,s,a,b}$, we can see that it is efficient in visiting the tuple (h,s,a,b) under the true MG. Therefore, we are able to bound the difference between P and $\widetilde{P}$ in the sense of value function.

Lemma 6.2. *W.h.p, for any policy $(\mu,\nu)\in\Pi_A\times\Pi_B$,*

$$\left|V^{\mu,\nu}(r,\widetilde{P})-V^{\mu,\nu}(r,P)\right|\le\widetilde{O}\left(\frac{H^5S^3A^2B^2}{T}\right).$$

The "variance": difference between $\widehat{P}$ and $\widetilde{P}$. Since P^{int} is close to $\widetilde{P}$, we can prove the following lemma.

Lemma 6.3. *W.h.p, for any policy $(\mu,\nu)\in\Pi_A\times\Pi_B$,*

$$\left|V^{\mu,\nu}(r,\widehat{P})-V^{\mu,\nu}(r,\widetilde{P})\right|\le\widetilde{O}\left(\sqrt{\frac{H^3S^2AB}{T}}\right).$$

Put everything together. Combining the bounds of the "bias" term and the "variance" term, because of triangular inequality, we have the conclusion in Lemma 6.1 holds (the "bias" term is lower order). Then the proof of regret bound is completed by plugging in $\epsilon_k=\widetilde{O}\left(\sqrt{\frac{H^3S^2AB}{T^{(k)}}}\right)$ in (2).

7. Conclusion

In this paper, we take the initial step to study the well-motivated problem of low adaptive multi-agent reinforcement learning. We design Algorithm 1 that achieves the optimal batch complexity of $O(H + \log\log K)$ and near optimal regret of $\widetilde{O}(\sqrt{H^2S^2ABT})$. Furthermore, our techniques naturally extend to bandit games and reward-free exploration with low adaptivity, where our results generalize or improve various previous works in the single-agent case. Future extensions are numerous, it remains open to address the computational efficiency, study Markov Games with general function approximation, as well as make the algorithms practical. We leave those as future works.

Acknowledgements

The research is partially supported by NSF Awards #2007117 and #2003257. YW was with the Computer Science Department at UCSB when the work was completed and submitted for peer review at ICML.

Impact Statement

This paper is a theory paper that studies multi-agent RL with adaptivity constraints, where the goal is to advance the field of machine learning. Therefore, there will not be ethical issues or negative potential broader impact.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Yu Bai and Chi Jin. Provable self-play algorithms for competitive reinforcement learning. In *International Conference on Machine Learning*, pages 551–560. PMLR, 2020.

Yu Bai, Tengyang Xie, Nan Jiang, and Yu-Xiang Wang. Provably efficient q-learning with low switching cost. *Advances in Neural Information Processing Systems*, 32, 2019.

Yu Bai, Chi Jin, and Tiancheng Yu. Near-optimal reinforcement learning with self-play. *Advances in neural information processing systems*, 33:2159–2170, 2020.

Rajarshi Bhattacharyya, Archana Bura, Desik Rengarajan, Mason Rumuly, Srinivas Shakkottai, Dileep Kalathil, Ricky KP Mok, and Amogh Dhamdhere. Qflow: A reinforcement learning approach to high qoe video streaming over wireless networks. In *Proceedings of the twentieth ACM international symposium on mobile ad hoc networking and computing*, pages 251–260, 2019.

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

Nicolo Cesa-Bianchi, Ofer Dekel, and Ohad Shamir. Online learning with switching costs and other adaptive adversaries. In *Advances in Neural Information Processing Systems*, pages 1160–1168, 2013.

Jinglin Chen, Aditya Modi, Akshay Krishnamurthy, Nan Jiang, and Alekh Agarwal. On the statistical efficiency of reward-free exploration in non-linear rl. *Advances in Neural Information Processing Systems*, 35:20960–20973, 2022.

Xiaoyu Chen, Jiachen Hu, Lin F Yang, and Liwei Wang. Near-optimal reward-free exploration for linear mixture mdps with plug-in solver. *arXiv preprint arXiv:2110.03244*, 2021.

Herman Chernoff et al. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952.

Hwei-Ming Chung, Sabita Maharjan, Yan Zhang, and Frank Eliassen. Distributed deep reinforcement learning for intelligent load scheduling in residential smart grids. *IEEE Transactions on Industrial Informatics*, 17(4):2752–2763, 2020.

Qiwen Cui, Kaiqing Zhang, and Simon Du. Breaking the curse of multiagents in a large state space: Rl in markov games with independent linear function approximation. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 2651–2652. PMLR, 2023.

Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5713–5723, 2017.

Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes*. Springer Science & Business Media, 2012.

Minbo Gao, Tianle Xie, Simon S Du, and Lin F Yang. A provably efficient algorithm for linear markov decision process with low switching cost. *arXiv preprint arXiv:2101.00494*, 2021.

Zijun Gao, Yanjun Han, Zhimei Ren, and Zhengqing Zhou. Batched multi-armed bandits problem. *Advances in Neural Information Processing Systems*, 32, 2019.

Yanjun Han, Zhengqing Zhou, Zhengyuan Zhou, Jose Blanchet, Peter W Glynn, and Yinyu Ye. Sequential batch learning in finite-action linear contextual bandits. *arXiv preprint arXiv:2004.06321*, 2020.

Jiafan He, Heyang Zhao, Dongruo Zhou, and Quanquan Gu. Nearly minimax optimal reinforcement learning for linear markov decision processes. In *International Conference on Machine Learning*, pages 12790–12822. PMLR, 2023.

Pihe Hu, Yu Chen, and Longbo Huang. Towards minimax optimal reward-free reinforcement learning in linear mdps. In *The Eleventh International Conference on Learning Representations*, 2023.

Baihe Huang, Jason D Lee, Zhaoran Wang, and Zhuoran Yang. Towards general function approximation in zero-sum markov games. In *10th International Conference on Learning Representations, ICLR 2022*, 2022a.

Jiawei Huang, Jinglin Chen, Li Zhao, Tao Qin, Nan Jiang, and Tie-Yan Liu. Towards deployment-efficient reinforcement learning: Lower bound and optimality. In *International Conference on Learning Representations*, 2022b.

Xiang Ji and Gen Li. Regret-optimal model-free reinforcement learning for discounted mdps with short burn-in time. *arXiv preprint arXiv:2305.15546*, 2023.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pages 4863–4873, 2018.

Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR, 2020.

Chi Jin, Qinghua Liu, Yuanhao Wang, and Tiancheng Yu. V-learning–a simple, efficient, decentralized algorithm for multiagent rl. *arXiv preprint arXiv:2110.14555*, 2021.

Chi Jin, Qinghua Liu, and Tiancheng Yu. The power of exploiter: Provable multi-agent rl in large state spaces. In *International Conference on Machine Learning*, pages 10251–10279. PMLR, 2022.

Emmeran Johnson, Ciara Pike-Burke, and Patrick Rebeschini. Sample-efficiency in multi-batch reinforcement learning: The need for dimension-dependent adaptivity. *arXiv preprint arXiv:2310.01616*, 2023.

Emilie Kaufmann, Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Edouard Leurent, and Michal Valko. Adaptive reward-free exploration. In *Algorithmic Learning Theory*, pages 865–891. PMLR, 2021.

Dingwen Kong, Ruslan Salakhutdinov, Ruosong Wang, and Lin F Yang. Online sub-sampling for reinforcement learning with general function approximation. *arXiv preprint arXiv:2106.07203*, 2021.

Chris Junchi Li, Dongruo Zhou, Quanquan Gu, and Michael Jordan. Learning two-player markov games: Neural function approximation and correlated equilibrium. *Advances in Neural Information Processing Systems*, 35:33262–33274, 2022.

Gen Li, Yuling Yan, Yuxin Chen, and Jianqing Fan. Minimax-optimal reward-agnostic exploration in reinforcement learning. *arXiv preprint arXiv:2304.07278*, 2023.

Qinghua Liu, Tiancheng Yu, Yu Bai, and Chi Jin. A sharp analysis of model-based reinforcement learning with self-play. In *International Conference on Machine Learning*, pages 7001–7010. PMLR, 2021.

Weichao Mao, Lin Yang, Kaiqing Zhang, and Tamer Basar. On improving model-free algorithms for decentralized multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 15007–15049. PMLR, 2022.

Tatsuya Matsushima, Hiroki Furuta, Yutaka Matsuo, Ofir Nachum, and Shixiang Gu. Deployment-efficient reinforcement learning via model-based offline optimization. In *International Conference on Learning Representations*, 2020.

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.

Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Emilie Kaufmann, Edouard Leurent, and Michal Valko. Fast active learning for pure exploration in reinforcement learning. In *International Conference on Machine Learning*, pages 7599–7608. PMLR, 2021.

Sobhan Miryoosefi and Chi Jin. A simple reward-free approach to constrained reinforcement learning. In *International Conference on Machine Learning*, pages 15666–15698. PMLR, 2022.

Aditya Modi, Jinglin Chen, Akshay Krishnamurthy, Nan Jiang, and Alekh Agarwal. Model-free representation learning and exploration in low-rank mdps. *arXiv preprint arXiv:2102.07035*, 2021.

George Nemhauser and Laurence Wolsey. Polynomial-time algorithms for linear programming. *Integer and Combinatorial Optimization*, pages 146–181, 1988.

Vianney Perchet, Philippe Rigollet, Sylvain Chassang, and Erik Snowberg. Batched bandit problems. *The Annals of Statistics*, 44(2):660–681, 2016.

Dan Qiao and Yu-Xiang Wang. Near-optimal deployment efficiency in reward-free reinforcement learning with linear function approximation. In *The Eleventh International Conference on Learning Representations*, 2023.

Dan Qiao, Ming Yin, Ming Min, and Yu-Xiang Wang. Sample-efficient reinforcement learning with loglog(T) switching cost. In *International Conference on Machine Learning*, pages 18031–18061. PMLR, 2022.

Dan Qiao, Ming Yin, and Yu-Xiang Wang. Logarithmic switching cost in reinforcement learning beyond linear mdps. *arXiv preprint arXiv:2302.12456*, 2023.

Shuang Qiu, Jieping Ye, Zhaoran Wang, and Zhuoran Yang. On reward-free rl with kernel and neural function approximations: Single-agent mdp and markov game. In *International Conference on Machine Learning*, pages 8737–8747. PMLR, 2021.

Yufei Ruan, Jiaqi Yang, and Yuan Zhou. Linear bandits with limited adaptivity and learning distributional optimal design. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 74–87, 2021.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

M Shi, Y Liang, and N Shroff. Near-optimal adversarial reinforcement learning with switching costs. In *International Conference on Learning Representations (ICLR)*, 2023.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

David Simchi-Levi and Yunzong Xu. Phase transitions and cyclic phenomena in bandits with switching constraints. *Advances in Neural Information Processing Systems*, 32, 2019.

Grigoris Velegkas, Zhuoran Yang, and Amin Karbasi. The best of both worlds: Reinforcement learning with logarithmic regret and policy switches. *arXiv preprint arXiv:2203.01491*, 2022.

Andrew J Wagenmaker, Yifang Chen, Max Simchowitz, Simon Du, and Kevin Jamieson. Reward-free rl is no harder than reward-aware rl in linear markov decision processes. In *International Conference on Machine Learning*, pages 22430–22456. PMLR, 2022.

Ruosong Wang, Simon S Du, Lin Yang, and Russ R Salakhutdinov. On reward-free reinforcement learning with linear function approximation. *Advances in neural information processing systems*, 33:17816–17826, 2020.

Tianhao Wang, Dongruo Zhou, and Quanquan Gu. Provably efficient reinforcement learning with linear function approximation under adaptivity constraints. *Advances in Neural Information Processing Systems*, 34, 2021.

Yuanhao Wang, Qinghua Liu, Yu Bai, and Chi Jin. Breaking the curse of multiagency: Provably efficient decentralized multi-agent rl with function approximation. *arXiv preprint arXiv:2302.06606*, 2023.

Qiaomin Xie, Yudong Chen, Zhaoran Wang, and Zhuoran Yang. Learning zero-sum simultaneous-move markov games using function approximation and correlated equilibrium. In *Conference on learning theory*, pages 3674–3682. PMLR, 2020.

Nuoya Xiong, Zhuoran Yang, and Zhaoran Wang. A general framework for sequential decision-making under adaptivity constraints. *arXiv preprint arXiv:2306.14468*, 2023.

Shusheng Xu, Yancheng Liang, Yunfei Li, Simon Shaolei Du, and Yi Wu. Beyond information gain: An empirical benchmark for low-switching-cost reinforcement learning. *Transactions on Machine Learning Research*, 2022.

Yunchang Yang, Han Zhong, Tianhao Wu, Bin Liu, Liwei Wang, and Simon S Du. A reduction-based framework for sequential decision making with delayed feedback. *arXiv preprint arXiv:2302.01477*, 2023.

Deheng Ye, Guibin Chen, Wen Zhang, Sheng Chen, Bo Yuan, Bo Liu, Jia Chen, Zhao Liu, Fuhao Qiu, Hongsheng Yu, et al. Towards playing full moba games with deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33:621–632, 2020.

Andrea Zanette and Emma Brunskill. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In *International Conference on Machine Learning*, pages 7304–7312. PMLR, 2019.

Andrea Zanette, Alessandro Lazaric, Mykel J Kochenderfer, and Emma Brunskill. Provably efficient reward-agnostic navigation with linear value iteration. *Advances in Neural Information Processing Systems*, 33:11756–11766, 2020.

Ruiqi Zhang and Andrea Zanette. Policy finetuning in reinforcement learning via design of experiments using offline data. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Weitong Zhang, Dongruo Zhou, and Quanquan Gu. Reward-free model-based reinforcement learning with linear function approximation. *Advances in Neural Information Processing Systems*, 34:1582–1593, 2021.

Xuezhou Zhang, Adish Singla, et al. Task-agnostic exploration in reinforcement learning. *Advances in Neural Information Processing Systems*, 2020a.

Zihan Zhang, Simon S Du, and Xiangyang Ji. Nearly minimax optimal reward-free reinforcement learning. *arXiv preprint arXiv:2010.05901*, 2020b.

Zihan Zhang, Yuan Zhou, and Xiangyang Ji. Almost optimal model-free reinforcement learning via reference-advantage decomposition. *Advances in Neural Information Processing Systems*, 33:15198–15207, 2020c.

Zihan Zhang, Yuhang Jiang, Yuan Zhou, and Xiangyang Ji. Near-optimal regret bounds for multi-batch reinforcement learning. *Advances in Neural Information Processing Systems*, 35:24586–24596, 2022.

Heyang Zhao, Jiafan He, and Quanquan Gu. A nearly optimal and low-switching algorithm for reinforcement learning with general function approximation. *arXiv preprint arXiv:2311.15238*, 2023.

A. Extended related work

Low switching algorithms for bandits and RL. Auer et al. (2002) first studied multi-armed bandits with low switching cost by proposing the famous UCB2 algorithm. The optimal switching cost bound is later achieved by Cesa-Bianchi et al. (2013). For multi-armed bandits with A arms and T episodes, they designed an algorithm with the optimal $\widetilde{O}(\sqrt{AT})$ regret while the switching cost is only $O(A\log\log T)$. Simchi-Levi and Xu (2019) generalized the result by showing that a switching cost of order $A\log\log T$ is necessary for getting the optimal $\widetilde{O}(\sqrt{T})$ regret bound. For stochastic linear bandits, Abbasi-Yadkori et al. (2011) achieved the optimal regret $\widetilde{O}(d\sqrt{T})$ with $O(d\log T)$ policy switches by applying doubling trick. Under a slightly different setting, Ruan et al. (2021) improved the result by reducing the switching cost to $O(\log\log T)$ while keeping the regret bound. Bai et al. (2019) first studied the problem under tabular MDP. They reached regret bound of $\widetilde{O}(\sqrt{H^3SAT})$ with local switching cost $O(H^3SA\log T)$ by applying doubling trick to Q-learning. Both regret and switching cost are improved by Zhang et al. (2020c) using advantage decomposition. Qiao et al. (2022) established that a global switching cost of order $HSA\log\log T$ is enough and necessary to achieve the optimal $\widetilde{O}(\sqrt{T})$ regret. For linear MDP, Gao et al. (2021) arrived at regret bound $\widetilde{O}(\sqrt{d^3H^3T})$ with global switching cost $O(dH\log T)$ by applying doubling trick to LSVI-UCB. Wang et al. (2021) generalized the above result to work for arbitrary budget of switching cost. Later, the regret bound is improved to the minimax optimal rate $\widetilde{O}(\sqrt{d^2H^2T})$ by He et al. (2023), while the switching cost remains the same. Beyond the linear MDP model, Qiao et al. (2023) designed low switching algorithms for linear Bellman complete MDP and MDP with general linear approximation, while Kong et al. (2021); Velegkas et al. (2022); Zhao et al. (2023); Xiong et al. (2023) considered MDPs with general function approximation. All of these algorithms achieved a switching cost bound depending only logarithmically on T. Shi et al. (2023) considered low switching adversarial reinforcement learning. On the empirical side, Xu et al. (2022) constructed an empirical benchmark for low switching RL. In addition, low switching algorithms can be further applied to deal with RL with delayed feedback (Yang et al., 2023) or to achieve short burn-in time under discounted MDPs (Ji and Li, 2023).

Batched bandits and RL. For multi-armed bandits with A arms and T episodes, Cesa-Bianchi et al. (2013) designed an algorithm with $\widetilde{O}(\sqrt{AT})$ regret while the batch complexity is only $O(\log\log T)$. Perchet et al. (2016) proved that $\Omega(\log\log T)$ batches are necessary for a regret bound of $\widetilde{O}(\sqrt{T})$ under 2-armed bandits. The result is generalized to K-armed bandits by Gao et al. (2019). For stochastic linear bandits, Han et al. (2020) achieved a regret bound of $\widetilde{O}(\sqrt{T})$ while the batch complexity is only $O(\log\log T)$. The result is improved by Ruan et al. (2021) via using weaker assumptions. For batched RL setting, Qiao et al. (2022) showed that their algorithm uses the optimal $O(H+\log\log T)$ batches to achieve the optimal $\widetilde{O}(\sqrt{T})$ regret. Later, Zhang et al. (2022) incorporated the idea of optimal experimental design to get near optimal regret bound and computational efficiency. For linear MDP, Wang et al. (2021) first designed an algorithm with low batch complexity. Later, Johnson et al. (2023) proved a dimension-dependent lower bound of batch complexity for any sample-efficient algorithms. The deployment efficient algorithms for pure exploration by Huang et al. (2022b); Qiao and Wang (2023) also satisfy the definition of batched RL. Beyond linear MDPs, Modi et al. (2021) designed deployment efficient algorithms for low rank MDPs while Xiong et al. (2023) proposed batched algorithms for MDPs with general function approximation.

Reward-free exploration. The problem of reward-free exploration is first studied by Jin et al. (2020). They used the EULER algorithm (Zanette and Brunskill, 2019) to visit each state-action pair as much as possible, and their sample complexity is $\widetilde{O}(H^5S^2A/\epsilon^2)$. Kaufmann et al. (2021) designed an algorithm which requires $\widetilde{O}(H^4S^2A/\epsilon^2)$ episodes to output a near-optimal policy for any reward function. Ménard et al. (2021) further improved the sample complexity to $\widetilde{O}(H^3S^2A/\epsilon^2)$. Zhang et al. (2020b) considered a more general horizon-free setting with stationary transition kernel. They constructed a novel condition to achieve the optimal sample complexity $\widetilde{O}(S^2A/\epsilon^2)$ under their specific setting. Also, their result can be transferred to achieve $\widetilde{O}(H^3S^2A/\epsilon^2)$ sample complexity under the standard setting where $r_h\in[0,1]$ and the transition kernel is non-stationary. When considering low adaptivity, Qiao et al. (2022) designed an algorithm with the optimal $O(HSA)$ switching cost and $\widetilde{O}(H^5S^2A/\epsilon^2)$ sample complexity. There is a relevant setting named task-agnostic exploration. Zhang et al. (2020a) designed an algorithm that could find ϵ-optimal policies for N arbitrary tasks within at most $\widetilde{O}(H^5SA\log N/\epsilon^2)$ exploration episodes. Recently, Li et al. (2023) studied reward-agnostic exploration in reinforcement learning, which incorporates the two settings above. For MDP with linear function approximation, a series of papers (Wang et al., 2020; Zanette et al., 2020; Zhang et al., 2021; Chen et al., 2021; Wagenmaker et al., 2022; Huang et al., 2022b; Qiao and Wang, 2023; Hu et al., 2023) analyzed reward-free exploration. Among the works, Huang et al. (2022b); Qiao and Wang (2023) achieved near optimal adaptivity at the same time. Beyond linear MDPs, Qiu et al. (2021); Chen et al. (2022) considered reward-free exploration in MDPs with general function approximation. Reward-free RL is also known to be

helpful to constrained reinforcement learning (Miryoosefi and Jin, 2022).

B. Missing algorithm: EstimateTransition (Algorithm 5) and some explanation

Our Algorithm 5 is a generalization of Algorithm 5 in Qiao et al. (2022) to the multi-agent case. The algorithm takes a data set $\mathcal{D}$, a set of infrequent tuples $\mathcal{F}$, a transition model P and the target layer h to update as inputs. The output is an updated transition model where the h-th layer is derived according to $\mathcal{D}$ while the remaining layers stay the same. The construction of P_h is for the tuples in $\mathcal{F}$, $P_h(s'|s,a,b) = 0$. For the tuples not in $\mathcal{F}$, $P_h(s'|s,a,b)$ is the empirical estimate from $\mathcal{D}$. At last, $P_h(s^\dagger|s,a,b) = 1 - \sum_{s' \in \mathcal{S}:(h,s,a,b,s') \notin \mathcal{F}} P_h(s'|s,a,b)$ holds so that P_h is a valid transition kernel. In other words, the construction is similar to the construction of $\widetilde{P}$. We first let P be the empirical estimate based on $\mathcal{D}$, then for $(h,s,a,b,s') \in \mathcal{F}$, we move the probability of $P_h(s'|s,a,b)$ to $P_h(s^\dagger|s,a,b)$.

Algorithm 5 Compute Transition Kernel (EstimateTransition)

Require: Data set $\mathcal{D}$, infrequent tuples $\mathcal{F}$, absorbing state $s^\dagger$, the target layer h, transition kernel P.
Output: Estimated transition kernel P from data set $\mathcal{D}$.
$\diamond$ Count the visitation number of each state-action pairs from the target layer h:
for $(s,a,b,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times \mathcal{S}$ **do**
  $N_h(s,a,b,s') =$ count of (h,s,a,b,s') in $\mathcal{D}$.
  $N_h(s,a,b) =$ count of (h,s,a,b) in $\mathcal{D}$.
end for
$\diamond$ Update the h-th layer of the transition kernel:
for $(s,a,b,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times \mathcal{S}$ s.t. $(h,s,a,b,s') \in \mathcal{F}$ **do**
  $P_h(s'|s,a,b) = 0$.
end for
for $(s,a,b,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times \mathcal{S}$ s.t. $(h,s,a,b,s') \notin \mathcal{F}$ **do**
  $P_h(s'|s,a,b) = \frac{N_h(s,a,b,s')}{N_h(s,a,b)}$.
end for
for $(s,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}$ **do**
  $P_h(s^\dagger|s,a,b) = 1 - \sum_{s' \in \mathcal{S}:(h,s,a,b,s') \notin \mathcal{F}} P_h(s'|s,a,b)$.
end for
for $(a,b) \in \mathcal{A} \times \mathcal{B}$ **do**
  $P_h(s^\dagger|s^\dagger,a,b) = 1$.
end for
Return P.

C. Transition between original MG and absorbing MG

For any reward function r defined under the original MG P, we overload the notation and also use it under the absorbing MG. The extended definition of r under the absorbing MG is shown below:

$$r(s,a,b) = \begin{cases} r(s,a,b), & s \in \mathcal{S}, \\ 0, & s = s^\dagger. \end{cases}$$

For any policy $\pi = (\mu,\nu)$ defined under the original MG P, we overload the notation and also use it under the absorbing MG. The extended definition of (μ,ν) under the absorbing MG is shown below:

$$\mu(\cdot|s) = \begin{cases} \mu(\cdot|s), & s \in \mathcal{S}, \\ \text{arbitrary distribution}, & s = s^\dagger. \end{cases}$$

$$\nu(\cdot|s) = \begin{cases} \nu(\cdot|s), & s \in \mathcal{S}, \\ \text{arbitrary distribution}, & s = s^\dagger. \end{cases}$$

With the definition of r and π, the value function under the absorbing MG is independent of the *arbitrary distribution* since there will be no more reward once the agent enters the absorbing state $s^\dagger$. The policies defined under the absorbing MG can be directly applied under the real MG because such policies have definition for any $s \in \mathcal{S}$.

In this paper, P is the real MG, which is under original MG. In each stage, $\widetilde{P}$ is an absorbing MG constructed based on infrequent tuples $\mathcal{F}$ and the real MG P. When we run the algorithm, we do not have access to $\widetilde{P}$, but we know exactly the intermediate transition kernel P^{int}, which is also an absorbing MG. In Algorithm 3, the $\widehat{P}$ we construct is the empirical estimate of $\widetilde{P}$, which is also an absorbing MG. In the proof of this paper, a large part of discussion is under the framework of absorbing MG. When we specify that the discussion is under absorbing MG with absorbing state $s^\dagger$, any transition kernel P' satisfies $P'_h(s^\dagger|s^\dagger, a, b) = 1$ for any $(a, b, h) \in \mathcal{A} \times \mathcal{B} \times [H]$. For the reward functions in this paper, they are all defined under original MG, when applied under absorbing MG, the transition rule follows what we just discussed.

D. Proof of lemmas regarding Crude Exploration (Algorithm 2)

We first prove an upper bound for batch complexity.

Lemma D.1. *The batch complexity of Algorithm 2 is bounded by H.*

Proof of Lemma D.1. Since each $\pi_h =$ uniform mixture of $\{\pi_{h,s,a,b}\}_{(s,a,b)}$ is one general policy and we run H such π_h's in total, the number of batches is bounded by H. □

Note that our Crude Exploration (Algorithm 2) is adapted from Algorithm 2 in Qiao et al. (2022) for the single-agent RL case, and the difference is that here we replace the single version space ϕ in Qiao et al. (2022) with a product policy space $\Pi_A \times \Pi_B$. Therefore, the results in Qiao et al. (2022) directly extend to the multi-agent case. We begin with the property of P^{int} and $\widetilde{P}$.

Lemma D.2 (Lemma E.3 in Qiao et al. (2022))**.** *With probability $1 - \frac{S\delta}{K}$, $\forall\,(h, s, a, b, s') \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times \mathcal{S}$ such that $(h, s, a, b, s') \notin \mathcal{F}$, it holds that*

$$(1 - \frac{1}{H})P_h^{int}(s'|s, a, b) \le \widetilde{P}_h(s'|s, a, b) \le (1 + \frac{1}{H})P_h^{int}(s'|s, a, b).$$

In addition, we have that $\forall\,(h, s, a, b, s') \in \mathcal{F}$, $\widetilde{P}_h(s'|s, a, b) = P_h^{int}(s'|s, a, b) = 0$.

From Lemma D.2, we can see that for those tuples (h, s, a, b, s') not in $\mathcal{F}$, the estimate of the transition kernel satisfies $(1 - \frac{1}{H})P_h^{int}(s'|s, a, b) \le \widetilde{P}_h(s'|s, a, b) \le (1 + \frac{1}{H})P_h^{int}(s'|s, a, b)$ with high probability. In addition, for those states $(h, s, a, b, s') \in \mathcal{F}$, $P_h^{int}(s'|s, a, b) = \widetilde{P}_h(s'|s, a, b) = 0$, which means this inequality holds for all $(h, s, a, b, s') \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times \mathcal{S}$. For simplicity, we use a new definition θ-multiplicatively accurate to describe the relationship between P^{int} and $\widetilde{P}$.

Definition D.3 (θ-multiplicatively accurate for transition kernels (under absorbing MGs))**.** Under the absorbing MG with absorbing state $s^\dagger$, a transition kernel P' is θ-multiplicatively accurate to another transition kernel P'' if

$$(1 - \theta)P'_h(s'|s, a, b) \le P''_h(s'|s, a, b) \le (1 + \theta)P'_h(s'|s, a, b)$$

for all $(h, s, a, b, s') \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times \mathcal{S}$ and there is no requirement for the case when $s' = s^\dagger$.

Because of Lemma D.2, we have that with probability $1 - \frac{S\delta}{K}$, P^{int} is $\frac{1}{H}$-multiplicatively accurate to $\widetilde{P}$. Next, we will compare the visitation probability of each state (h, s, a, b) under two transition kernels that are close to each other.

Lemma D.4 (Lemma E.5 in Qiao et al. (2022))**.** *Define $1_{h,s,a,b}$ to be the reward function r' such that $r'_{h'}(s', a', b') = \mathbb{1}[(h', s', a', b') = (h, s, a, b)]$. Similarly, define $1_{h,s}$ to be the reward function r' such that $r'_{h'}(s', a', b') = \mathbb{1}[(h', s') = (h, s)]$. Then $V^{\mu,\nu}(1_{h,s,a,b}, P')$ and $V^{\mu,\nu}(1_{h,s}, P')$ denote the visitation probability of (h, s, a, b) and (h, s), respectively, under (μ, ν) and P'. Under the absorbing MG with absorbing state $s^\dagger$, if P' is $\frac{1}{H}$-multiplicatively accurate to P'', for any policy pair (μ, ν) and any $(h, s, a, b) \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{B}$, it holds that*

$$\frac{1}{4}V^{\mu,\nu}(1_{h,s,a,b}, P') \le V^{\mu,\nu}(1_{h,s,a,b}, P'') \le 3V^{\mu,\nu}(1_{h,s,a,b}, P').$$

Combining Lemma D.2 and Lemma D.4, we have with high probability, for any policy pair (μ, ν) and any $(h, s, a, b) \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{B}$,

$$\frac{1}{4} V^{\mu,\nu}(1_{h,s,a,b}, P^{int}) \leq V^{\mu,\nu}(1_{h,s,a,b}, \widetilde{P}) \leq 3V^{\mu,\nu}(1_{h,s,a,b}, P^{int}). \tag{3}$$

The structure of the absorbing MG also gives rise to the following lemma about the relationship between $\widetilde{P}$ and P.

Lemma D.5 (Lemma E.6 in Qiao et al. (2022)). *For any policy pair (μ, ν) and any $(h, s, a, b) \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{B}$,*

$$V^{\mu,\nu}(1_{h,s,a,b}, P) \geq V^{\mu,\nu}(1_{h,s,a,b}, \widetilde{P}).$$

Now we are ready to state the key lemma about the difference between $\widetilde{P}$ and P.

Lemma D.6 (Lemma E.12 in Qiao et al. (2022)). *There exists a universal constant $c_1 > 0$, such that with probability $1 - \frac{2S\delta}{K}$, it holds that for any policy pair $(\mu, \nu) \in \Pi_A \times \Pi_B$ and reward function r',*

$$0 \leq V^{\mu,\nu}(r', P) - V^{\mu,\nu}(r', \widetilde{P}) \leq \frac{c_1 H^5 S^3 A^2 B^2 \iota}{T}.$$

Recall that at the end of Crude Exploration (Algorithm 2), we store a policy $\pi =$ uniform mixture of $\{\pi_{h,s,a,b}\}_{(h,s,a,b)}$ for future use. Below we prove some properties of the uniformly explorative policy π.

Lemma D.7. *There exists a universal constant c_2, with probability at least $1 - \frac{S\delta}{K}$, the policy $\pi =$ uniform mixture of $\{\pi_{h,s,a,b}\}_{(h,s,a,b)}$ satisfies that for all $(h, s, a, b, s') \in [H] \times \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times \mathcal{S}$ such that $(h, s, a, b, s') \notin \mathcal{F}$,*

$$T \cdot \mathbb{P}_\pi[(h, s, a, b, s')|P] \geq c_2 H^2 \iota,$$

where $\mathbb{P}_\pi[(h, s, a, b, s')|P]$ means the probability of reaching (s, a, b, s') at time step h under policy π and original MG P.

Proof of Lemma D.7. Recall that we run $\pi_h =$ uniform mixture of $\{\pi_{h,s,a,b}\}_{(s,a,b)}$ for $T_0 = \frac{T}{H}$ episodes in Algorithm 2. Denote $E_{h,s,a,b,s'} = T \cdot \mathbb{P}_\pi[(h, s, a, b, s')|P]$ and $\overline{E}_{h,s,a,b,s'} = \frac{T}{H} \cdot \mathbb{P}_{\pi_h}[(h, s, a, b, s')|P]$. Then it holds that $E_{h,s,a,b,s'} \geq \overline{E}_{h,s,a,b,s'}$, because π is equivalent to a uniform mixture of $\{\pi_h\}_{h\in[H]}$. Therefore, for any $(h, s, a, b, s') \notin \mathcal{F}$, with probability $1 - \frac{\delta}{HSABK}$,

$$\begin{aligned} &E_{h,s,a,b,s'} + \sqrt{3E_{h,s,a,b,s'}\iota} \geq \overline{E}_{h,s,a,b,s'} + \sqrt{3\overline{E}_{h,s,a,b,s'}\iota} \\ \geq &\text{visitation count of } (h, s, a, b, s') \geq C_1 H^2 \iota, \end{aligned} \tag{4}$$

where the second inequality holds with probability $1 - \frac{\delta}{HSABK}$ due to Multiplicative Chernoff bound (Lemma H.1). The last inequality results from the definition of infrequent tuples $\mathcal{F}$.

Finally, based on (4), with probability $1 - \frac{\delta}{HSABK}$, $E_{h,s,a,b,s'} \geq c_2 H^2 \iota$. The conclusion holds because of a union bound over $(h, s, a, b, s') \notin \mathcal{F}$. □

E. Proof of lemmas regarding Fine Exploration (Algorithm 3)

We first state a conclusion about the batch complexity of Algorithm 3.

Lemma E.1. *The batch complexity of Algorithm 3 is bounded by 2.*

Proof of Lemma E.1. Algorithm 3 will just run two general policies: π and π' for several episodes. □

Below we analyze the properties of the two policies π and π'. Recall that π is constructed as below.

$$\pi = \text{argmin}_{\pi\in\Delta(\Pi_A\times\Pi_B)} \sup_{(\mu',\nu')\in\Pi_A\times\Pi_B} \sum_{h=1}^{H} \sum_{s,a,b} \frac{V^{\mu',\nu'}(1_{h,s,a,b}, P^{int})}{V^{\pi}(1_{h,s,a,b}, P^{int})}. \tag{5}$$

We begin with the uniform coverage property of the policy π in Algorithm 3.

Lemma E.2. *The policy π in Algorithm 3 satisfies that*

$$\sup_{(\mu',\nu')\in\Pi_A\times\Pi_B}\sum_{h=1}^{H}\sum_{s,a,b}\frac{V^{\mu',\nu'}(1_{h,s,a,b},P^{int})}{V^{\pi}(1_{h,s,a,b},P^{int})}\leq HSAB.$$

Proof of Lemma E.2. The proof is by plugging $\mathcal{X}=\{\{V^{\mu,\nu}(1_{h,\cdot,\cdot,\cdot},P^{int})\}_{h=1}^{H}\mid(\mu,\nu)\in\Pi_A\times\Pi_B\}$, $d=SAB$ and $m=H$ into Lemma H.6. □

Together with the relationship between P^{int} and $\widetilde{P}$ ((3)), we can replace the P^{int} with $\widetilde{P}$.

Lemma E.3. *Conditioned on the high-probability case in Lemma D.2, the policy π in Algorithm 3 satisfies that*

$$\sup_{(\mu',\nu')\in\Pi_A\times\Pi_B}\sum_{h=1}^{H}\sum_{s,a,b}\frac{V^{\mu',\nu'}(1_{h,s,a,b},\widetilde{P})}{V^{\pi}(1_{h,s,a,b},\widetilde{P})}\leq 12HSAB.$$

Proof of Lemma E.3. For some pair of (μ',ν'), it holds that

$$\begin{aligned}L.H.S&=\sum_{h=1}^{H}\sum_{s,a,b}\frac{V^{\mu',\nu'}(1_{h,s,a,b},\widetilde{P})}{V^{\pi}(1_{h,s,a,b},\widetilde{P})}\\&\leq\sum_{h=1}^{H}\sum_{s,a,b}\frac{3V^{\mu',\nu'}(1_{h,s,a,b},P^{int})}{\frac{1}{4}V^{\pi}(1_{h,s,a,b},P^{int})}\\&\leq 12HSAB,\end{aligned}\tag{6}$$

where the first inequality is because of Lemma D.4, the last inequality holds due to Lemma E.2. □

Now we state the properties of the auxiliary policy π'. Recall that according to the main algorithm (Algorithm 1), when the input infrequent set and intermediate transition kernel are $\mathcal{F}^k$ and $P^{int,k}$ respectively ($k=1,2$), we have the auxiliary policy $\pi'=\pi_k$ and the number of episodes $T'=NT^{(k)}$ (N is the constant in Algorithm 1). Due to the conclusion of Lemma D.7, we have the following lemma for π' and T'.

Lemma E.4. *There exists a universal constant $N>0$, such that with probability at least $1-\frac{2S\delta}{K}$, for all possible $\mathcal{F}$, π', T' that appear in Algorithm 3, it holds that for all $(h,s,a,b,s')\notin\mathcal{F}$,*

$$T'\cdot\mathbb{P}_{\pi'}[(h,s,a,b,s')|P]\geq 3C_1H^2\iota,$$

where C_1 is the universal constant in Algorithm 2, $\mathbb{P}_{\pi}[(h,s,a,b,s')|P]$ means the probability of reaching (s,a,b,s') at time step h under policy π and original MG P.

Proof of Lemma E.4. According to Lemma D.7, for a fixed pair of $\mathcal{F}^k$, π_k and $T^{(k)}$, with probability at least $1-\frac{S\delta}{K}$, it holds that for all $(h,s,a,b,s')\notin\mathcal{F}^k$,

$$T^{(k)}\cdot\mathbb{P}_{\pi_k}[(h,s,a,b,s')|P]\geq c_2H^2\iota.$$

Therefore, choosing $\pi'=\pi_k$ and $T'=NT^{(k)}$ with $N>\frac{3C_1}{c_2}$ implies that for a fixed pair of $\mathcal{F}^k$, π_k and $T^{(k)}$, with probability at least $1-\frac{S\delta}{K}$, it holds that for all $(h,s,a,b,s')\notin\mathcal{F}^k$,

$$NT^{(k)}\cdot\mathbb{P}_{\pi_k}[(h,s,a,b,s')|P]\geq 3C_1H^2\iota.$$

Finally, with a union bound over $k=1,2$, the proof is complete. □

The goal of running the auxiliary policy π' for T' episodes is to visit each state action pair not in the infrequent set for sufficient times. The property is summarized as below.

Lemma E.5. *Under the high-probability case in Lemma E.4, if we run policy π' for T' episodes, with probability $1-\frac{S\delta}{K}$, it holds that for all $(h,s,a,b,s')\notin\mathcal{F}$,*

$$N_h(s,a,b,s')\geq C_1H^2\iota,$$

where $N_h(s,a,b,s')$ is the visitation count of (h,s,a,b,s').

Proof of Lemma E.5. Under the high-probability case in Lemma E.4, for a fixed $(h,s,a,b,s')\notin\mathcal{F}$,

$$T'\cdot\mathbb{P}_{\pi'}[(h,s,a,b,s')|P]\geq 3C_1H^2\iota.$$

Then according to Lemma H.5, with probability $1-\frac{\delta}{HSABK}$, it holds that

$$N_h(s,a,b,s')\geq\frac{1}{2}T'\cdot\mathbb{P}_{\pi'}[(h,s,a,b,s')|P]-\iota\geq C_1H^2\iota.$$

Finally, the proof is complete due to a union bound over $(h,s,a,b,s')\notin\mathcal{F}$. □

Therefore, we could guarantee that with high probability, the refined transition kernel estimate $\widehat{P}$ is $\frac{1}{H}$-multiplicatively accurate to the absorbing MG $\widetilde{P}$, which is summarized in the lemma below.

Lemma E.6. *Under the high probability case in Lemma E.5, with probability $1-\frac{S\delta}{K}$, the output $\widehat{P}$ of Algorithm 3 is $\frac{1}{H}$-multiplicatively accurate to $\widetilde{P}$.*

Proof of Lemma E.6. The proof is identical to the proof of Lemma D.2 (and Lemma E.3 in Qiao et al. (2022)). □

It is known that the uncertainty of estimating $\widetilde{P}_h(\cdot|s,a,b)$ is proportional to $\sqrt{\frac{1}{N_h(s,a,b)}}$, where $N_h(s,a,b)$ is the visitation count of (h,s,a,b) in the data set. Therefore, we prove the following lemma regarding $N_h(s,a,b)$'s.

Lemma E.7. *Under the high probability case in Lemma E.4, if we run π for T episodes and π' for T' episodes, with probability $1-\frac{\delta}{K}$, for all $(h,s,a,b)\in[H]\times\mathcal{S}\times\mathcal{A}\times\mathcal{B}$, at least one of the following two statements hold:*
(1) For all $s'\in\mathcal{S}$, $(h,s,a,b,s')\in\mathcal{F}$, i.e. $\widetilde{P}_h(s^\dagger|s,a,b)=1$ is fixed and known.
(2) $N_h(s,a,b)\geq\frac{1}{2}TV^\pi(1_{h,s,a,b},\widetilde{P})$.

Proof of Lemma E.7. For any $(h,s,a,b)\in[H]\times\mathcal{S}\times\mathcal{A}\times\mathcal{B}$, if there exists some $s'\in\mathcal{S}$, $(h,s,a,b,s')\notin\mathcal{F}$, then the expected visitation count of (h,s,a,b) from running π' for T' episodes is

$$E^1_{h,s,a,b}=T'\mathbb{P}_{\pi'}[(h,s,a,b)|P]\geq T'\mathbb{P}_{\pi'}[(h,s,a,b,s')|P]\geq 3C_1H^2\iota,$$

where the last inequality results from Lemma E.4.

In addition, the expected visitation count of (h,s,a,b) from running π for T episodes is

$$E^2_{h,s,a,b}=TV^\pi(1_{h,s,a,b},P)\geq TV^\pi(1_{h,s,a,b},\widetilde{P}),$$

where the inequality is because of Lemma D.5.

Combining the expectations, the total expected visitation count of (h,s,a,b) is

$$E_{h,s,a,b}=E^1_{h,s,a,b}+E^2_{h,s,a,b}\geq TV^\pi(1_{h,s,a,b},\widetilde{P})+3C_1H^2\iota.$$

According to Lemma H.5, with probability $1-\frac{\delta}{HSABK}$, it holds that

$$N_h(s,a,b)\geq\frac{1}{2}E_{h,s,a,b}-\iota\geq\frac{1}{2}TV^\pi(1_{h,s,a,b},\widetilde{P}).$$

Finally, the proof is complete due to a union bound over all $(h,s,a,b)\in[H]\times\mathcal{S}\times\mathcal{A}\times\mathcal{B}$. □

Now we are ready to prove the following key lemma that bounds the difference between $\widehat{P}$ and $\widetilde{P}$.

Lemma E.8. *Under the high-probability cases in Lemma E.3, Lemma E.6 and Lemma E.7, with probability* $1-\frac{S\delta}{K}$*, the output* $\widehat{P}$ *of Algorithm 3 satisfies that for all* $(\mu,\nu)\in\Pi_A\times\Pi_B$ *and any reward function* r'*,*

$$\left|V^{\mu,\nu}(r',\widehat{P})-V^{\mu,\nu}(r',\widetilde{P})\right|\le c_3\sqrt{\frac{H^3S^2AB\iota}{T}}+c_3\frac{H^2S^2AB\iota}{T}, \tag{7}$$

where c_3 *is a universal constant.*

Proof of Lemma E.8. Define $N_h(s,a,b)$ to be the visitation count of (h,s,a,b) in the data set $\mathcal{D}$. Then according to Bernstein's inequality (Lemma H.2), with probability $1-\frac{S\delta}{K}$, for all $(h,s,a,b,s')\in[H]\times\mathcal{S}\times\mathcal{A}\times\mathcal{B}\times\mathcal{S}$,

$$\left|\widehat{P}_h(s'|s,a,b)-\widetilde{P}_h(s'|s,a,b)\right|\le\sqrt{\frac{2\widetilde{P}_h(s'|s,a,b)\iota}{N_h(s,a,b)}}+\frac{2\iota}{3N_h(s,a,b)}. \tag{8}$$

Based on the above event and the high-probability cases in Lemma E.3, Lemma E.6 and Lemma E.7, for any $(\mu,\nu)\in\Pi_A\times\Pi_B$ and reward function r', we define $\{f_h(\cdot)\}_{h=1}^{H+1}$ to be the value function of (μ,ν) under $\widetilde{P}$ and r' ($f_{H+1}(\cdot)=0$). Then it holds that

$$\begin{aligned}
&\left|V^{\mu,\nu}(r',\widehat{P})-V^{\mu,\nu}(r',\widetilde{P})\right|\le\sum_{h,s,a,b}V^{\mu,\nu}(1_{h,s,a,b},\widehat{P})\left|\left(\widehat{P}-\widetilde{P}\right)\cdot f_{h+1}(s,a,b)\right|\\
\le&\sum_{h,s,a,b}V^{\mu,\nu}(1_{h,s,a,b},\widehat{P})\cdot\left|\sum_{s'}\left(\widehat{P}_h(s'|s,a,b)-\widetilde{P}_h(s'|s,a,b)\right)\cdot\left(f_{h+1}(s')-\widetilde{P}_h(\cdot|s,a,b)\cdot f_{h+1}(\cdot)\right)\right|\\
\le&\sum_{h,s,a,b}V^{\mu,\nu}(1_{h,s,a,b},\widehat{P})\cdot\sum_{s'}\left(\sqrt{\frac{2\widetilde{P}_h(s'|s,a,b)\iota}{N_h(s,a,b)}}+\frac{2\iota}{3N_h(s,a,b)}\right)\cdot\left|f_{h+1}(s')-\widetilde{P}_h(\cdot|s,a,b)\cdot f_{h+1}(\cdot)\right|\\
\le&4\sum_{h,s,a,b}V^{\mu,\nu}(1_{h,s,a,b},\widetilde{P})\cdot\left[\sqrt{\frac{2S\iota\cdot\mathrm{Var}_{\widetilde{P}_h(\cdot|s,a,b)}f_{h+1}(\cdot)}{N_h(s,a,b)}}+\frac{HS\iota}{N_h(s,a,b)}\right]\\
\le&4\sqrt{2S\iota}\cdot\sqrt{\underbrace{\sum_{h,s,a,b}\frac{V^{\mu,\nu}(1_{h,s,a,b},\widetilde{P})}{N_h(s,a,b)}}_{\text{(i)}}}\cdot\sqrt{\underbrace{\sum_{h,s,a,b}V^{\mu,\nu}(1_{h,s,a,b},\widetilde{P})\mathrm{Var}_{\widetilde{P}_h(\cdot|s,a,b)}f_{h+1}(\cdot)}_{\text{(ii)}}}+4HS\iota\underbrace{\sum_{h,s,a,b}\frac{V^{\mu,\nu}(1_{h,s,a,b},\widetilde{P})}{N_h(s,a,b)}}_{\text{(i)}}\\
\le&c_3\sqrt{\frac{H^3S^2AB\iota}{T}}+c_3\frac{H^2S^2AB\iota}{T}.
\end{aligned} \tag{9}$$

The first inequality is because of simulation lemma (Lemma H.7). The second inequality holds since $\widetilde{P}_h(\cdot|s,a,b)\cdot f_{h+1}(\cdot)$ is independent of s'. The third inequality results from (8). The forth inequality is derived via Cauchy-Schwarz inequality and Lemma E.6. The fifth inequality is derived via Cauchy-Schwarz inequality. The last inequality comes from the inequalities (10) and (11) below.

The upper bound for (i) is shown below:

$$\text{(i)}=\sum_{h,s,a,b}\frac{V^{\mu,\nu}(1_{h,s,a,b},\widetilde{P})}{N_h(s,a,b)}\le\sum_{h,s,a,b}\frac{V^{\mu,\nu}(1_{h,s,a,b},\widetilde{P})}{\frac{1}{2}TV^{\pi}(1_{h,s,a,b},\widetilde{P})}\le\frac{24HSAB}{T}. \tag{10}$$

The first inequality is because of Lemma E.7 while the second inequality is due to Lemma E.3.

The upper bound for (ii) is shown below:

$$\text{(ii)}=\sum_{h,s,a,b}V^{\mu,\nu}(1_{h,s,a,b},\widetilde{P})\mathrm{Var}_{\widetilde{P}_h(\cdot|s,a,b)}f_{h+1}(\cdot)\le H^2, \tag{11}$$

where the inequality results from a recursive application of Law of Total Variance. □

F. Proof of main theorems

F.1. Proof of Theorem 4.1

We first give a proof for the upper bound on the number of stages.

Lemma F.1. *If $T^{(k)} = K^{1-\frac{1}{2^k}}$ for $k = 1, 2\cdots$, we have*

$$K_0 \le \min\{j : \sum_{k=1}^{j} T^{(k)} \ge K\} = O(\log\log K).$$

Proof of Lemma F.1. Take $j = \log_2\log_2 K$, we have $T^{(j)} = \frac{K}{K^{(\log_2 K)^{-1}}} = \frac{K}{2}$, which means that

$$K_0 \le \log_2\log_2 K + 2 = O(\log\log K).$$

□

Then we are able to bound the batch complexity of Algorithm 1.

Lemma F.2. *The batch complexity of Algorithm 1 is bounded by $O(H + \log\log K)$.*

Proof of Lemma F.2. According to Lemma D.1 and Lemma E.1, the batch complexity for each of the first two stages is bounded by $H + 2$, while the batch complexity for each of the remaining stages is bounded by 2. Therefore the total batch complexity is bounded by $2H + 2K_0 = O(H + \log\log K)$. □

Recall that in Algorithm 1, in each stage k ($k = 1, 2$), we run Algorithm 2 to construct the infrequent tuples $\mathcal{F}^k$ and the intermediate transition kernel $P^{int,k}$. The absorbing MG $\widetilde{P}^k$ is constructed as in Definition 3.1 based on $\mathcal{F}^k$ and the real MG P. Then we run Algorithm 3 to construct an empirical estimate of $\widetilde{P}^k$, which is $\widehat{P}^k$. After the first two stages, for the following stages ($k = 3, 4, \cdots, K_0$), we keep the infrequent tuples $\mathcal{F}^2$, the absorbing MG $\widetilde{P}^2$ and the intermediate transition kernel $P^{int,2}$. In each stage, we run Algorithm 3 to construct an empirical estimate of $\widetilde{P}^2$, which is $\widehat{P}^k$.

Lemma F.3. *There exists a universal constant $C > 0$, such that with probability $1-\delta$, for any $k = 1, 2$ and $(\mu, \nu) \in \Pi_A^k \times \Pi_B$,*

$$|V^{\mu,\nu}(r, \widehat{P}^k) - V^{\mu,\nu}(r, P)| \le C\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(k)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(k)}}\right),$$

while for any $k \ge 3$ and $(\mu, \nu) \in \Pi_A^k \times \Pi_B$,

$$|V^{\mu,\nu}(r, \widehat{P}^k) - V^{\mu,\nu}(r, P)| \le C\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(k)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(2)}}\right).$$

Proof of Lemma F.3. Conditioned on the high probability case in Lemma D.6 (for $k = 1, 2$), it holds that for any $k = 1, 2$ and $(\mu, \nu) \in \Pi_A^k \times \Pi_B$,

$$\left|V^{\mu,\nu}(r, P) - V^{\mu,\nu}(r, \widetilde{P}^k)\right| \le \frac{c_1H^5S^3A^2B^2\iota}{T^{(k)}},$$

where c_1 is a universal constant and r is the real reward function.

In addition, conditioned on the high probability case in Lemma E.8 (for $k = 1, 2, \cdots, K_0$), it holds that for $k = 1$ and any $(\mu, \nu) \in \Pi_A^1 \times \Pi_B$,

$$\left|V^{\mu,\nu}(r, \widehat{P}^1) - V^{\mu,\nu}(r, \widetilde{P}^1)\right| \le c_3\sqrt{\frac{H^3S^2AB\iota}{T^{(1)}}} + c_3\frac{H^2S^2AB\iota}{T^{(1)}},$$

while for any $k \ge 2$ and $(\mu, \nu) \in \Pi_A^k \times \Pi_B$,

$$\left|V^{\mu,\nu}(r, \widehat{P}^k) - V^{\mu,\nu}(r, \widetilde{P}^2)\right| \le c_3\sqrt{\frac{H^3S^2AB\iota}{T^{(k)}}} + c_3\frac{H^2S^2AB\iota}{T^{(k)}}.$$

By combining the above inequalities and choosing $C = c_1 + c_3$, the conclusion holds.

The failure probability is bounded by $O\left(\frac{S\delta}{K}\right) \times O(\log\log K) \le \delta$, if $K \ge \widetilde{\Omega}(S)$. □

Based on the above high-probability case, the estimations of value functions are uniformly accurate for all policy pairs, which implies a sub-optimality bound for the remaining policies in the version space.

Lemma F.4. *With probability $1-\delta$, for the policies that have not been eliminated at stage k, i.e. $\mu \in \Pi_A^{k+1}$, we have that*

$$V^\star(r,P) - V^{\mu,\dagger}(r,P) \le 4C\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(k)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(k)}}\right),\ k=1,2.$$

$$V^\star(r,P) - V^{\mu,\dagger}(r,P) \le 4C\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(k)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(2)}}\right),\ k\ge 3.$$

Proof of Lemma F.4. We only prove the case for $k=1$, the remaining cases can be proven similarly.

We prove under the high probability case in Lemma F.3, which says that for any $(\mu,\nu)\in\Pi_A^1\times\Pi_B$,

$$|V^{\mu,\nu}(r,\widehat{P}^1) - V^{\mu,\nu}(r,P)| \le C\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(1)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(1)}}\right).$$

Recall that $\mu^\star$ is the Nash policy of the max-player. Then it holds that

$$\begin{aligned}
&\sup_{\mu\in\Pi_A^1}\inf_{\nu\in\Pi_B} V^{\mu,\nu}(r,\widehat{P}^1) - \inf_{\nu\in\Pi_B} V^{\mu^\star,\nu}(r,\widehat{P}^1) = \inf_{\nu\in\Pi_B} V^{\widehat{\mu},\nu}(r,\widehat{P}^1) - \inf_{\nu\in\Pi_B} V^{\mu^\star,\nu}(r,\widehat{P}^1)\\
\le&\left|\inf_{\nu\in\Pi_B} V^{\widehat{\mu},\nu}(r,\widehat{P}^1) - \inf_{\nu\in\Pi_B} V^{\widehat{\mu},\nu}(r,P)\right| + \inf_{\nu\in\Pi_B} V^{\widehat{\mu},\nu}(r,P) - \inf_{\nu\in\Pi_B} V^{\mu^\star,\nu}(r,P)\\
&+\left|\inf_{\nu\in\Pi_B} V^{\mu^\star,\nu}(r,P) - \inf_{\nu\in\Pi_B} V^{\mu^\star,\nu}(r,\widehat{P}^1)\right|\\
\le&2C\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(1)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(1)}}\right),
\end{aligned}$$

where the last inequality holds due to Lemma H.8.

Therefore, the Nash policy $\mu^\star$ will not be eliminated. In addition, for $\mu\in\Pi_A^2$,

$$\begin{aligned}
\inf_{\nu\in\Pi_B} V^{\mu^\star,\nu}(r,\widehat{P}^1) - \inf_{\nu\in\Pi_B} V^{\mu,\nu}(r,\widehat{P}^1) \le& \sup_{\mu\in\Pi_A^1}\inf_{\nu\in\Pi_B} V^{\mu,\nu}(r,\widehat{P}^1) - \inf_{\nu\in\Pi_B} V^{\mu,\nu}(r,\widehat{P}^1)\\
\le&2C\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(1)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(1)}}\right).
\end{aligned}$$

Finally, together with Lemma H.8, it holds that

$$\begin{aligned}
&V^\star(r,P) - V^{\mu,\dagger}(r,P)\\
\le&\left|V^\star(r,P) - \inf_{\nu\in\Pi_B} V^{\mu^\star,\nu}(r,\widehat{P}^1)\right| + \inf_{\nu\in\Pi_B} V^{\mu^\star,\nu}(r,\widehat{P}^1) - \inf_{\nu\in\Pi_B} V^{\mu,\nu}(r,\widehat{P}^1) + \left|\inf_{\nu\in\Pi_B} V^{\mu,\nu}(r,\widehat{P}^1) - \inf_{\nu\in\Pi_B} V^{\mu,\nu}(r,P)\right|\\
\le&4C\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(1)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(1)}}\right).
\end{aligned}$$

The conclusions for the remaining stages can be proven by identical techniques and induction. □

Given the sub-optimality bound for the policies, we are ready to provide the final regret bound.

Lemma F.5. *Conditioned on the high probability event in Lemma F.3, if $K\ge\widetilde{\Omega}(H^{14}S^8A^6B^6)$, the total regret is bounded by $\widetilde{O}(\sqrt{H^2S^2ABT})$, where $T:=HK$ is the number of steps.*

Proof of Lemma F.5. The regret for the first stage (stage 1) is at most $(N+2)HT^{(1)} = O(HK^{\frac{1}{2}})$. Because of Lemma F.4, the regret for the second stage (stage 2) is at most $(N+2)T^{(2)} \times 4C\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(1)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(1)}}\right)$. For stage $k \geq 3$, since the remaining policies (any $\mu \in \Pi_A^k$) are at most $4C\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(k-1)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(2)}}\right)$ sub-optimal and the policy π is a mixture of remaining policies, the regret due to running π for T episodes is at most $4CT^{(k)}\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(k-1)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(2)}}\right)$. Besides, the regret due to running π_2 for $NT^{(2)}$ episodes is bounded by $NT^{(2)} \times 4C\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(1)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(1)}}\right)$. Combining the results, the regret for the k-th stage is at most $O\left(T^{(k)}\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(k-1)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(2)}}\right) + T^{(2)}\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(1)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(1)}}\right)\right)$.

Adding up the regret for each stage , we have that the total regret is bounded by

$$\begin{aligned}
\text{Regret}(K) &\leq O(HK^{\frac{1}{2}}) + O\left(\sum_{k=2}^{K_0} T^{(k)}\sqrt{\frac{H^3S^2AB\iota}{T^{(k-1)}}}\right) + O\left(K \cdot \frac{H^5S^3A^2B^2\iota}{T^{(2)}}\right) \\
&\quad + O\left(T^{(2)}\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(1)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(1)}}\right)\right) \cdot O(\log\log K) \\
&= O(HK^{\frac{1}{2}}) + O(\sqrt{H^3S^2ABK\iota} \cdot \log\log K) + O(H^5S^3A^2B^2K^{\frac{1}{4}}\iota \cdot \log\log K) \\
&= \widetilde{O}(\sqrt{H^3S^2ABK}),
\end{aligned}$$

where the last equality is because $K \geq \widetilde{\Omega}(H^{14}S^8A^6B^6)$. □

Then Theorem 4.1 holds because of Lemma F.2, Lemma F.3 and Lemma F.5.

F.2. Proof of Theorem 4.2

Since two-player zero-sum MG strictly generalizes single-agent MDPs, the lower bound directly follows from Theorem 2 of Zhang et al. (2022).

F.3. Proof of Theorem 4.3

We can output any policy μ in the remaining policy set $\Pi_A^{K_0+1}$ (μ is a Markov policy). Because there are $K_0 = O(\log\log K)$ stages in total, the maximal $T^{(k)}$ is larger than $\Omega(\frac{K}{\log\log K})$. According to Lemma F.4, for any $\mu \in \Pi_A^{K_0+1}$,

$$V^\star(r,P) - V^{\mu,\dagger}(r,P) \leq 4C\left(\sqrt{\frac{H^3S^2AB\iota}{T^{(k)}}} + \frac{H^5S^3A^2B^2\iota}{T^{(2)}}\right), \forall\, k = 2,3,\cdots,K_0.$$

Combining these two results, we have for any $\mu \in \Pi_A^{K_0+1}$,

$$V^\star(r,P) - V^{\mu,\dagger}(r,P) \leq \widetilde{O}\left(\sqrt{\frac{H^3S^2AB}{K}}\right).$$

Then $K = \widetilde{O}\left(\frac{H^3S^2AB}{\epsilon^2}\right)$ bounds the above by ϵ.

G. Proof for the results in Section 5

G.1. Details for the bandit game setting

Recall that bandit game is a special case of Markov games where $H = S = 1$. The action set for the max-player is $\mathcal{A}$ ($A := |\mathcal{A}|$) and the action set for the min-player is $\mathcal{B}$ ($B := |\mathcal{B}|$). Therefore, a policy μ for the max-player is a distribution

over $\mathcal{A}$ while a policy ν for the min-player is a distribution over $\mathcal{B}$. For simplicity, here we denote the reward function by $r(a,b)$. In addition, the value function under policy pair (μ,ν) and the Nash value are denoted by $r(\mu,\nu)$ and $r^\star$ respectively. The goal is to minimize the regret for the max-player:

$$\text{Regret}(K) = \sum_{k=1}^{K} \left[r^\star - r(\mu^k, \dagger)\right] = \sum_{k=1}^{K} \left[r^\star - \inf_{\nu} r(\mu^k, \nu)\right].$$

Now we apply Algorithm 1 (with some revision) to the bandit game setting. The main difference is that now we do not need to construct the absorbing model since we can directly try each action. The detailed algorithm is shown below.

Algorithm 6 Algorithm for bandit games

Require: Number of episodes K. Universal constant C. Failure probability δ.
Initialize: $T^{(k)} = K^{1-\frac{1}{2^k}}$, $k \le K_0 = O(\log\log K)$. $\Pi_A^1 :=$ {All stochastic policies for the max-player}, $\mu_0 :=$ uniform distribution over $\mathcal{A}$, $\nu_0 :=$ uniform distribution over $\mathcal{B}$. $\iota = \log(2ABK/\delta)$, $T^0 = 2AB\iota$. Data set $\mathcal{D} = \emptyset$.
for $k = 1, 2, \cdots, K_0$ **do**
  ◇ Number of episodes in k-th stage:
  if $\sum_{i=1}^{k} T^{(i)} + kT^0 \ge K$ **then**
    $T^{(k)} = K - kT^0 - \sum_{i=1}^{k-1} T^{(i)}$ (o.w. $T^{(k)} = K^{1-\frac{1}{2^k}}$).
  end if
end for
for $k = 1, 2, \cdots, K_0$ **do**
  ◇ Construct an explorative policy for the max-player:
  Construct policy $\mu = \text{argmin}_{\mu\in\Pi_A^k} \sup_{\mu'\in\Pi_A^k} \sum_{a\in\mathcal{A}} \frac{\mu'(a)}{\mu(a)}$.
  ◇ Run the explorative policy and the uniform policy for several episodes:
  Run $\pi = (\mu, \nu_0)$ for $T^{(k)}$ episodes and store the rewards in $\mathcal{D}$.
  Run $\pi_0 = (\mu_0, \nu_0)$ for T^0 episodes and store the rewards in $\mathcal{D}$.
  Construct an empirical reward function $\widehat{r}^k(a,b)$ (and therefore $\widehat{r}^k(\mu,\nu)$) using $\mathcal{D}$.
  ◇ Policy elimination for the max-player:
  $\Pi_A^{k+1} \leftarrow \left\{\mu \in \Pi_A^k \mid \min_{b\in\mathcal{B}} \widehat{r}^k(\mu,b) \ge \sup_{\mu\in\Pi_A^k} \min_{b\in\mathcal{B}} \widehat{r}^k(\mu,b) - 2C\left(\sqrt{\frac{AB\iota}{T^{(k)}}}\right)\right\}$.
  ◇ Reset the data set:
  Clear the data set: $\mathcal{D} \leftarrow \emptyset$.
end for

Now we restate the upper bounds and give a proof.

Theorem G.1 (Restate Theorem 5.1)**.** *If we run Algorithm 6 (adapted from Algorithm 1) under a two-player zero-sum bandit game for K episodes, the batch complexity is bounded by $O(\log\log K)$ while the regret is $\widetilde{O}(\sqrt{ABK})$ with high probability. Furthermore, the algorithm is computationally efficient.*

Proof of Theorem G.1. Using identical proof of Lemma F.1, we have the number of stages $K_0 \le O(\log\log K)$. In addition, in each stage, we only run two policies π and π_0, which can be done in two batches. Therefore, the batch complexity is bounded by $2K_0 = O(\log\log K)$.

For the regret part, we first have the following property of the μ^k constructed in k-th stage: for any $\mu' \in \Pi_A^k$,

$$\sum_{a\in\mathcal{A}} \frac{\mu'(a)}{\mu^k(a)} \le A,$$

which follows from Lemma H.6. In addition, using identical proof of Lemma E.7, it holds that with probability $1-\delta/2$, for all $(k,a,b) \in [K_0]\times\mathcal{A}\times\mathcal{B}$,

$$N^k(a,b) \ge \frac{T^{(k)}}{2} \cdot \frac{\mu^k(a)}{B},$$

where $N^k(a,b)$ is the visitation count of (a,b) in the data set $\mathcal{D}$ from the k-th stage.

Now we are ready to bound the estimation error of $\widehat{r}^k$. With probability $1-\delta$, uniformly for all $(k,\mu,b)\in[K_0]\times\Pi_A^k\times\mathcal{B}$,

$$\begin{aligned}\left|\widehat{r}^k(\mu,b)-r(\mu,b)\right|\leq&\sum_{a\in\mathcal{A}}\mu(a)|\widehat{r}^k(a,b)-r(a,b)|\\ \leq&C\sum_{a\in\mathcal{A}}\mu(a)\sqrt{\frac{\iota}{N^k(a,b)}}\leq C\sum_{a\in\mathcal{A}}\mu(a)\sqrt{\frac{B\iota}{T^{(k)}\mu^k(a)}}\\ \leq&C\sqrt{B\iota\sum_{a\in\mathcal{A}}\mu(a)}\cdot\sqrt{\sum_{a\in\mathcal{A}}\frac{\mu(a)}{T^{(k)}\mu^k(a)}}\leq C\sqrt{\frac{AB\iota}{T^{(k)}}},\end{aligned}$$

where the second inequality holds for some universal constant C with probability $1-\delta/2$ due to Hoeffding's inequality. The forth inequality holds due to Cauchy-Schwarz inequality.

Based on the uniform estimation error, the regret bound of $\widetilde{O}(\sqrt{ABK})$ can be derived using identical proof as Lemma F.4 and Lemma F.5.

Lastly, we deal with the computational complexity. Note that any policy μ for the max-player can be represented as a vector $\mu=(p_1,\cdots,p_A)$ such that $\sum_{i=1}^A p_i=1$. We prove by induction that the constrains on the remaining policy set are at most $O(B\log\log K)$ linear constraints and all steps are efficient. Assume the induction assumption is true at the beginning of the k-th stage, then $\mu=\operatorname{argmin}_{\mu\in\Pi_A^k}\sup_{\mu'\in\Pi_A^k}\sum_{a\in\mathcal{A}}\frac{\mu'(a)}{\mu(a)}$ can be constructed (approximately) in $\operatorname{poly}(A,B,K)$ time according to Lemma H.6. Then given the data set $\mathcal{D}^k$, we do policy elimination. Recall that we will keep the policy μ if $\min_{b\in\mathcal{B}}\widehat{r}^k(\mu,b)\geq\sup_{\mu\in\Pi_A^k}\min_{b\in\mathcal{B}}\widehat{r}^k(\mu,b)-2C\left(\sqrt{\frac{AB\iota}{T^{(k)}}}\right)$. Now we consider the computation.

The computation of the R.H.S is equivalent to solving
$\max x$ s.t. $x\leq\sum_{i=1}^A p_i\widehat{r}^k(i,b)$ for all $b\in\mathcal{B}$ and the previous conditions on $(p_1,\cdots,p_A)$.
According to the induction assumption, the previous conditions are at most $O(B\log\log K)$ linear constraints. Therefore the problem is a linear programming (LP) problem with at most $A+1$ variables and $O(B\log\log K)$ linear constraints, which can be solved in polynomial time (Nemhauser and Wolsey, 1988). With the value c of R.H.S, the elimination step is equivalent to adding the following B linear constraints to the policy set: $\sum_{i=1}^A p_i\widehat{r}^k(i,b)\geq c$ for all $b\in\mathcal{B}$. Since there are $O(\log\log K)$ stages in total, the total number of linear constraints is bounded by $O(B\log\log K)$.

As a result, the induction assumption always holds and the computation is efficient.

□

G.2. Proof for Theorem 5.2

In this part, we restate Theorem 5.2 and provide a proof.

Theorem G.2 (Restate Theorem 5.2)**.** *The batch complexity of Algorithm 4 is bounded by $H+2$. There exists a constant $c>0$ such that, for any $\epsilon>0$ and any $\delta>0$, if the number of total episodes K satisfies that*

$$K>c\left(\frac{H^3S^2AB\iota'}{\epsilon^2}+\frac{H^5S^3A^2B^2\iota'}{\epsilon}\right),$$

where $\iota'=\log(\frac{HSAB}{\epsilon\delta})$, then there exists a choice of N_0 and N_1 such that $N_0+N_1=K$ and with probability $1-\delta$, for any reward function r, Algorithm 4 will output a policy pair $\pi^r=(\mu^r,\nu^r)$ that is ϵ-approximate Nash.

Proof of Lemma G.2. We first prove the batch complexity upper bound. According to Lemma D.1 and Lemma E.1, the batch complexity of Algorithm 4 is bounded by $H+2$.

For the sample complexity, due to Lemma D.6, with probability $1-\delta/2$, for any policy pair $(\mu,\nu)\in\Pi_A\times\Pi_B$ and reward function r, it holds that (for some universal constant c_1)

$$\left|V^{\mu,\nu}(r,P)-V^{\mu,\nu}(r,\widetilde{P})\right|\leq\frac{c_1H^5S^3A^2B^2\iota}{N_0}.$$

In addition, resulting from Lemma E.8, with probability $1-\delta/2$, for all $(\mu,\nu)\in\Pi_A\times\Pi_B$ and any reward function r, it holds that (for some universal constant c_3)

$$\left|V^{\mu,\nu}(r,\widehat{P})-V^{\mu,\nu}(r,\widetilde{P})\right|\le c_3\sqrt{\frac{H^3S^2AB\iota}{N_1}}+c_3\frac{H^2S^2AB\iota}{N_1}. \tag{12}$$

Combining the results, we have with probability $1-\delta$, for all $(\mu,\nu)\in\Pi_A\times\Pi_B$ and any reward function r,

$$\left|V^{\mu,\nu}(r,\widehat{P})-V^{\mu,\nu}(r,P)\right|\le\frac{c_1H^5S^3A^2B^2\iota}{N_0}+c_3\sqrt{\frac{H^3S^2AB\iota}{N_1}}+c_3\frac{H^2S^2AB\iota}{N_1}. \tag{13}$$

Therefore there exists some universal constant c such that by choosing $N_0=\frac{cH^5S^3A^2B^2\iota'}{4\epsilon}$ and $N_1=\frac{cH^3S^2AB\iota'}{\epsilon^2}+\frac{cH^2S^2AB\iota'}{2\epsilon}$, the R.H.S of (13) is bounded by $\epsilon/2$. In this case, the total sample complexity is $\frac{cH^5S^3A^2B^2\iota'}{2\epsilon}+\frac{cH^3S^2AB\iota'}{\epsilon^2}+\frac{cH^2S^2AB\iota'}{2\epsilon}\le c\left(\frac{H^3S^2AB\iota'}{\epsilon^2}+\frac{H^5S^3A^2B^2\iota'}{\epsilon}\right)$.

Finally, we prove that $\mu^r=\mathrm{argmax}_{\mu\in\Pi_A}\inf_{\nu\in\Pi_B}V^{\mu,\nu}(r,\widehat{P})$ is ϵ-approximate Nash. Due to Lemma H.8, we have

$$\begin{aligned}
&V^\star(r,P)-V^{\mu^r,\dagger}(r,P)=V^\star(r,P)-\inf_{\nu\in\Pi_B}V^{\mu^r,\nu}(r,P)\\
\le&\left|\inf_{\nu\in\Pi_B}V^{\mu^\star,\nu}(r,P)-\inf_{\nu\in\Pi_B}V^{\mu^\star,\nu}(r,\widehat{P})\right|+\inf_{\nu\in\Pi_B}V^{\mu^\star,\nu}(r,\widehat{P})-\inf_{\nu\in\Pi_B}V^{\mu^r,\nu}(r,\widehat{P})\\
&+\left|\inf_{\nu\in\Pi_B}V^{\mu^r,\nu}(r,\widehat{P})-\inf_{\nu\in\Pi_B}V^{\mu^r,\nu}(r,P)\right|\\
\le&\frac{\epsilon}{2}+0+\frac{\epsilon}{2}=\epsilon.
\end{aligned}$$

The conclusion that ν^r is also ϵ-approximate Nash is proven by symmetry. □

H. Technical lemmas

Lemma H.1 (Multiplicative Chernoff bound (Chernoff et al., 1952)). *Let X be a Binomial random variable with parameter p,n. For any $\delta\in[0,1]$, we have that $\mathbb{P}[X>(1+\delta)pn]<\left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{np}$. A slightly looser bound that suffices for our propose:*

$$\mathbb{P}[X>(1+\delta)pn]<e^{-\frac{\delta^2pn}{3}}.$$

Lemma H.2 (Bernstein's inequality). *Let $x_1,\cdots,x_n$ be independent bounded random variables such that $\mathbb{E}[x_i]=0$ and $|x_i|\le A$ with probability 1. Let $\sigma^2=\frac{1}{n}\sum_{i=1}^n\mathrm{Var}[x_i]$, then with probability $1-\delta$ we have*

$$\left|\frac{1}{n}\sum_{i=1}^n x_i\right|\le\sqrt{\frac{2\sigma^2\log(2/\delta)}{n}}+\frac{2A}{3n}\log(2/\delta).$$

Lemma H.3 (Empirical Bernstein's inequality (Maurer and Pontil, 2009)). *Let $x_1,\cdots,x_n$ be i.i.d random variables such that $|x_i|\le A$ with probability 1. Let $\overline{x}=\frac{1}{n}\sum_{i=1}^n x_i$, and $\widehat{V}_n=\frac{1}{n}\sum_{i=1}^n(x_i-\overline{x})^2$, then with probability $1-\delta$ we have*

$$\left|\frac{1}{n}\sum_{i=1}^n x_i-\mathbb{E}[x]\right|\le\sqrt{\frac{2\widehat{V}_n\log(2/\delta)}{n}}+\frac{7A}{3n}\log(2/\delta).$$

Lemma H.4 (Lemma F.4 in (Dann et al., 2017)). *Let F_i for $i=1,\cdots$ be a filtration and $X_1,\cdots,X_n$ be a sequence of Bernoulli random variables with $\mathbb{P}(X_i=1|F_{i-1})=P_i$ with P_i being F_{i-1}-measurable and X_i being F_i measurable. It holds that*

$$\mathbb{P}\left[\exists\,n:\sum_{t=1}^n X_t<\sum_{t=1}^n P_t/2-W\right]\le e^{-W}.$$

Lemma H.5. *Let F_i for $i = 1, \cdots$ be a filtration and $X_1, \cdots, X_n$ be a sequence of Bernoulli random variables with $\mathbb{P}(X_i = 1|F_{i-1}) = P_i$ with P_i being F_{i-1}-measurable and X_i being F_i measurable. It holds that*

$$\mathbb{P}\left[\exists\, n : \sum_{t=1}^{n} X_t < \sum_{t=1}^{n} P_t/2 - \iota\right] \leq \frac{\delta}{HSABK},$$

where $\iota = \log(2HSABK/\delta)$.

Proof of Lemma H.5. Directly plug in $W = \iota$ in lemma H.4. □

Lemma H.6 (Lemma 1 in Zhang et al. (2022))**.** *Let $d > 0$ be an integer. Let $\mathcal{X} \subset (\Delta^d)^m$. Then there exists a distribution $\mathcal{D}$ over $\mathcal{X}$, such that*

$$\max_{x=\{x_i\}_{i=1}^{dm} \in \mathcal{X}} \sum_{i=1}^{dm} \frac{x_i}{y_i} = md,$$

where $y = \{y_i\}_{i=1}^{dm} = \mathbb{E}_{x\sim\mathcal{D}}[x]$. Moreover, if $\mathcal{X}$ has a boundary set $\partial\mathcal{X}$ with finite cardinality, we can find an approximation solution for $\mathcal{D}$ in poly($|\partial\mathcal{X}|$) time.

Lemma H.7 (Simulation lemma (Dann et al., 2017))**.** *For any two MGs M' and M'' with rewards r' and r'' and transition probabilities $\mathcal{P}'$ and $\mathcal{P}''$, the difference in values V', V'' with respect to the same policy pair (μ, ν) can be written as*

$$V'_h(s) - V''_h(s) = \mathbb{E}_{M'',\mu,\nu}\left[\sum_{i=h}^{H} [r'_i(s_i, a_i, b_i) - r''_i(s_i, a_i, b_i) + (\mathbb{P}'_i - \mathbb{P}''_i)V'_{i+1}(s_i, a_i, b_i)] \mid s_h = s\right].$$

Lemma H.8. *If for transition kernels P' and P'', a constant $\epsilon > 0$, a reward function r and policy sets Π_A, Π_B, it holds that for any $(\mu, \nu) \in \Pi_A \times \Pi_B$, $|V^{\mu,\nu}(r, P') - V^{\mu,\nu}(r, P'')| \leq \epsilon$, then we have for all $\mu \in \Pi_A$,*

$$\left|\inf_{\nu\in\Pi_B} V^{\mu,\nu}(r, P') - \inf_{\nu\in\Pi_B} V^{\mu,\nu}(r, P'')\right| \leq \epsilon.$$

As a result, it holds that

$$\left|\sup_{\mu\in\Pi_A}\inf_{\nu\in\Pi_B} V^{\mu,\nu}(r, P') - \sup_{\mu\in\Pi_A}\inf_{\nu\in\Pi_B} V^{\mu,\nu}(r, P'')\right| \leq \epsilon.$$

Proof of Lemma H.8. The lemma directly results from the property of $\sup$ and $\inf$. □