DECENTRALIZED ROBUST V-LEARNING FOR SOLVING MARKOV GAMES WITH MODEL UNCERTAINTY

Anonymous authors
Paper under double-blind review

ABSTRACT

Markov game is a popular reinforcement learning framework for modeling competitive players in a dynamic environment. However, most of the existing works on Markov game focus on computing a certain equilibrium following uncertain interactions among the players but ignore the uncertainty of the environment model, which is ubiquitous in practical scenarios. In this work, we develop a tractable solution to Markov games with environment model uncertainty. Specifically, we propose a new and tractable notion of robust correlated equilibrium for Markov games with environment model uncertainty. In particular, we prove that robust correlated equilibrium has a simple modification structure, and its characterization of equilibrium critically depends on the environment model uncertainty. Moreover, we propose the first fully-decentralized stochastic algorithm for computing such robust correlated equilibrium. Our analysis proves that the algorithm achieves the polynomial sample complexity $\widetilde{\mathcal{O}}(SA^2H^5\epsilon^{-2})$ for computing an approximate robust correlated equilibrium with ϵ accuracy.

1 INTRODUCTION

Markov game is a general and popular reinforcement learning framework for modeling multiple players competing with each other in a dynamic environment (Littman, 1994). In a Markov game, players interact with each other through a Markov decision process, and each player aims to improve its own decision-making to compete for more reward. In particular, many important real-life applications fit into this framework, including multi-player games (e.g., GO game (Silver et al., 2016; 2017)), decentralized multi-agent robotic control (Brambilla et al., 2013) and distributed autonomous driving (Shalev-Shwartz et al., 2016), etc.

One of the central goals of Markov game is to achieve Nash equilibrium (NE) among the players, i.e., an optimal product policy so that no player can improve its gain by deviating from its own policy alone. Such NE has been shown to exist for general Markov games (Filar & Vrieze, 2012). However, it turns out that finding NE of a Markov game is generally a PPAD-complete problem that cannot be efficiently solved in polynomial time (Deng et al., 2021; Jin et al., 2022b), except for some special Markov games with either zero-sum reward (Bai & Jin, 2020; Jin et al., 2018) or potential structure (Leonardos et al., 2021; Zhang et al., 2021). Therefore, instead of computing NE, researchers have proposed a tractable surrogate notion – the correlated equilibrium (CE) (Moulin & Vial, 1978), which is similar to NE but allows dependency among the players' policies (see Definition 2.3). Recently, many computation-efficient algorithms have been developed for computing CE with provable convergence guarantees (Jin et al., 2022a; Liu et al., 2021; Li et al., 2021).

Although Markov games have been extensively investigated, this standard framework only considers the competition among the players but ignores the 'competition' from the environment, i.e., environment model uncertainty, which is a critical factor that often reduces players' gains and must be considered in practical applications. For example, many applications such as autonomous driving and network robotic control naturally involve uncertain environments that have non-stationary or even adversarial transition kernels. As another example, the policy trained in a simulated environment often suffers from significant performance degradation when implemented in the real environment, due to model mismatch. In all these scenarios, it is much desired to learn an optimal

robust policy against such model uncertainty. In fact, to account for model uncertainty, many robust reinforcement learning approaches have been developed and extensively studied in the single-agent case (Wang & Zou, 2021; Li et al., 2022b;a; Neufeld & Sester, 2022). However, model uncertainty is still underexplored in the general case with multiple competing agents, where only two works (Kardeş et al., 2011; Zhang et al., 2020) exist to our knowledge. Specifically, Kardeş et al. (2011) applied the robust Markov game with model uncertainty to the application of queueing control. (Zhang et al., 2020) proposed provably convergent Q-learning and actor-critic type algorithms to compute a certain robust variant of NE of robust Markov games. However, computing robust NE of general Markov games with model uncertainty is in general a PPAD-complete problem and therefore no polynomial-time algorithm can exist. This motivates **the two major goals of this work**: (i) to propose a tractable surrogate robust equilibrium notion and study its fundamental properties; and (ii) to develop a fully decentralized, provably-convergent and computation-efficient algorithm for computing such robust equilibrium.

1.1 Our Contributions

In this work, we study episodic Markov games in an uncertain environment, i.e., the environment transition kernel in every time step is queried from an underlying uncertainty set. To find a proper equilibrium policy of such Markov games with model uncertainty,within polynomial time, we make the following technical contributions.

- We propose a new tractable notion of robust correlated equilibrium (CE) for Markov games with model uncertainty (see Definition 3.4). Specifically, robust CE generalizes the standard CE in that it is defined based on robust value function (see eq. (2)), which corresponds to the worst-case value function achieved under model uncertainty.
- We study the fundamental properties of robust CE. Specifically, we show that robust CE can be equivalently defined using either stochastic modification or deterministic modification (see Proposition 3.5). This indicates that robust CE inherits the modification structure from the standard CE. Moreover, through an illustrative example (see Proposition 3.6), we prove that the characterization of equilibrium of robust CE critically depends on the specific environment model uncertainty model, i.e., robust CE reduces to robust Nash equilibrium under certain uncertainty models.
- We develop a fully decentralized robust V-learning algorithm for finding robust CE of Markov games with model uncertainty. This algorithm is a generalization of the original V-learning algorithm (for solving standard Markov games) and adopts robust TD learning in its critic update. Under low level of model uncertainty, we prove that this algorithm achieves a polynomial episode complexity $\widetilde{\mathcal{O}}(SA^2H^5\epsilon^{-2})$ for computing an approximate robust CE with ϵ accuracy. This is the first non-asymptotic convergence result for solving Markov games with model uncertainty. Moreover, our analysis of robust V-learning is substantially different from that of the original V-learning. Please refer to the elaboration of technical novelty after Corollary 4.5 for more details. To briefly elaborate, this is because the use of robust TD update enables tracking the desired robust value function at the cost of introducing uncertainty to the state transitions when unrolling the iterative updates. Therefore, we need to bound the model uncertainty via a stronger convergence metric, which leads to solving a linear system that involves an upper triangular Toeplitz matrix.

1.2 Related Work

Markov games Markov games, also known as stochastic games, are standard formalism in multi-agent RL (Littman, 1994). The existence of NE for multi-player general-sum Markov games has been established in (Fink, 1964). Various algorithms have been designed to find NE, such as Nash-Q learning (Hu & Wellman, 2003), FF-Q learning (Littman et al., 2001), and correlated-Q learning (Greenwald et al., 2003). The first polynomial-time algorithm for finding NE is developed in (Hansen et al., 2013), but works only for zero-sum games. Recent studies showed that finding NE of general-sum multi-player games is PPAD-complete, so they cannot be solved in polynomial time (Deng et al., 2021; Jin et al., 2022b). Another notable goal in Markov games is to find a weaker version of NE, such as correlated equilibrium (CE) or coarse correlated equilibrium (CCE). Polynomial-time algorithms such as V-learning (Jin et al., 2022a; Mao & Başar, 2022; Song et al., 2021) and Nash value iteration (Liu et al., 2021) have been developed for computing CE and CCE.

Robust reinforcement learning Single-agent robust reinforcement learning has been widely explored (Nilim & Ghaoui, 2003; Nilim & El Ghaoui, 2005; Wiesemann et al., 2013; Satia & Lave Jr, 1973), which assume the environment transition kernel belongs to a given uncertainty set. Under a specific uncertainty model, Roy et al. (2017) and Wang & Zou (2021) developed model-free online robust Q-learning algorithms to solve the robust reinforcement learning problem. For robust multi-agent reinforcement learning, value iteration-based algorithm has been developed in (Kardeş et al., 2011) but with no explicit analytical form. For cooperative multi-agent reinforcement learning with model uncertainty, Huang et al. (2021) proposed a robust policy iteration algorithm to maximize the gain of the whole group. For non-cooperative Markov games with model uncertainty, Zhang et al. (2020) introduced robust Q-learning and actor-critic algorithms with asymptotic convergence guarantees of finding robust NE. To the best of our knowledge, there is no existing polynomial-time algorithm for solving Markov games with model uncertainty.

2 Preliminaries of Markov Game

An episodic m-player Markov game is specified by the five-element tuple $(H, \mathcal{S}, \mathcal{A}, \mathbb{P}, \{r^{(j)}\}_{j=1}^m)$, where H is the length of each episode, $\mathcal{S}$ and $\mathcal{A} := \bigtimes_{j=1}^m \mathcal{A}^{(j)}$ correspond to the state space and joint action space, respectively, and they are assumed to be finite. Moreover, $r^{(j)} : \mathcal{S} \times \mathcal{A} \to [0, 1]$ denotes the reward function of the j-th player and $\mathbb{P} := \{\mathbb{P}_h\}_{h=1}^H$ corresponds to the collection of transition kernels at time steps $h = 1, ..., H$. At every time step h, the players observe a global state $s_h \in \mathcal{S}$ of the environment. Then, they take a joint action $a_h = [a_h^{(1)}, \ldots, a_h^{(m)}]$ following a joint stochastic policy $\pi_h(\cdot|s_{1:h}, a_{1:(h-1)})$, which corresponds to a distribution on the joint action space $\mathcal{A}$ that depends on the past states $s_{1:h} := \{s_t\}_{t=1}^h$ and past actions $a_{1:(h-1)} := \{a_t\}_{t=1}^{h-1}$. After that, the global state transfers to a new state s_{h+1} following the state transition kernel $\mathbb{P}(\cdot|s_h, a_h)$, and each player j receives a local reward $r_h^{(j)}(s_h, a_h)$ from the environment.

In the above Markov game, each player j collects its own rewards over the episodes. In particular, denote $\pi := \{\pi_h\}_{h=1}^H$ as the collection of joint policies over the time steps, we can define the following value function for the j-th player at state s and time step h under policy π.

$$\text{(Value Function):} \quad \boldsymbol{v}_{\pi,h}^{(j)}(s) := \mathbb{E}\Big[\sum_{\ell=h}^{H} r_\ell^{(j)}(s_\ell, a_\ell)\Big| s_h = s, \pi, \mathbb{P}\Big], \tag{1}$$

which corresponds to the expected cumulative reward received by player j starting from state s at time step h under joint policy π. The goal of the player j is to optimize its own policy $\pi^{(j)} := \{\pi_h^{(j)}\}_{h=1}^H$ in order to maximize its associated value function. However, since every player's value function is also affected by the other players' policies and actions, the players must compete with each other to gain more rewards until they reach a certain equilibrium. Here, we introduce two popular equilibrium notions that will be discussed throughout the paper.

Definition 2.1 (Nash Equilibrium (NE))**.** A joint policy π is called an NE if the following two conditions are met: (i) for any time step h, the joint policy π_h is a product of independent policies, i.e., $\pi_h = \pi_h^{(1)} \times \ldots \times \pi_h^{(m)}$; (ii) For any player j with any associated policy $\widetilde{\pi}^{(j)}$, it holds that $\boldsymbol{v}_{\pi,1}^{(j)}(s) \geq \boldsymbol{v}_{\widetilde{\pi}^{(j)} \times \pi^{(\backslash j)},1}^{(j)}(s)$ for all states $s \in \mathcal{S}$. Here, $\pi^{(\backslash j)}$ denotes the joint policy of all the other players excluding the player j, and '$\times$' means that $\widetilde{\pi}^{(j)}$ is independent from $\pi^{(\backslash j)}$.

In the existing literature, it has been shown that computing NE is in general a PPAD-complete problem (Deng et al., 2021; Jin et al., 2022b), for which it is not possible to develop polynomial-time algorithms. This has motivated researchers to propose a surrogate correlated equilibrium (CE) notion (Moulin & Vial, 1978). Before introducing the formal definition of CE, we first define the following stochastic modification operator.

Definition 2.2 (Stochastic Modification)**.** At any time step h, denote $a_h^{(j)}$ as player j's action induced by joint policy π_h. Given the past states and actions $s_{1:h}, a_{1:(h-1)}$, a stochastic modification $\phi_h^{(j)}$ associated with player j randomly maps $a_h^{(j)}$ to another action $\widetilde{a}_h^{(j)}$, i.e., $\widetilde{a}_h^{(j)} \sim \phi_h^{(j)}(\cdot|s_{1:h}, a_{1:h-1}, a_h^{(j)})$. Moreover, we denote $\phi_h^{(j)} \circ \pi_h$ as the joint policy modified by $\phi_h^{(j)}$, i.e., π_h first generates a joint action $a_h := [a_h^{(j)}, a_h^{(\backslash j)}]$, and then $\phi_h^{(j)}$ maps $a_h^{(j)}$ to another $\widetilde{a}_h^{(j)}$.

Throughout, we denote $\phi^{(j)} := \{\phi_h^{(j)}\}_{h=1}^H$ and $\phi^{(j)} \circ \pi := \{\phi_h^{(j)} \circ \pi_h\}_{h=1}^H$ as the collections of stochastic modifications and modified policies over the episode, respectively. We are now ready to introduce the definition of correlated equilibrium (CE).

Definition 2.3 (Correlated Equilibrium (CE))**.** A joint policy π is called a CE if for any player j and any stochastic modification $\phi^{(j)}$, it holds that $\boldsymbol{v}_{\pi,1}^{(j)}(s) \geq \boldsymbol{v}_{\phi^{(j)}\circ\pi,1}^{(j)}(s)$ for all states $s \in \mathcal{S}$.

Intuitively, at CE, no player can improve its value function by modifying its own action induced by the joint CE policy. Compare to NE policies, CE policies do not require joint independence among all the players. In fact, it has been shown that any NE policy is guaranteed to be a CE policy (Jin et al., 2022a; Liu et al., 2021; Song et al., 2021), and hence CE is a weaker equilibrium notion than NE. Moreover, CE can be reformulated as linear programming and hence is tractable.

3 Markov Game with Model Uncertainty

In this section, we study episodic general-sum Markov games with uncertainty in the environment transition kernel. We aim to define a tractable notion of correlated equilibrium under such model uncertainty and study its fundamental properties.

3.1 Robust Correlated Equilibrium

We adopt the same episodic Markov game settings as described in Section 2, but consider uncertain transition kernel. Specifically, at every time step h and for every state-action pair (s, a), the environment transition kernel $\widetilde{\mathbb{P}}_h(\cdot|s, a)$ is uncertain and belongs to a general uncertainty set $\mathcal{P}_h(s, a)$. Below we list some popular examples of uncertainty set.

Example 3.1 (KL divergence)**.** The uncertainty set under KL divergence d_{KL} is defined as $\mathcal{P}_h(s, a) := \big\{\widetilde{\mathbb{P}}_h(\cdot|s, a)\colon d_{\text{KL}}\big(\mathbb{P}_h(\cdot|s, a), \widetilde{\mathbb{P}}_h(\cdot|s, a)\big) \leq \rho\big\}$, where $d_{\text{KL}}\big(\mathbb{P}, \widetilde{\mathbb{P}}\big) := \sum_{s\in\mathcal{S}} \widetilde{\mathbb{P}}(s) \ln \frac{\widetilde{\mathbb{P}}(s)}{\mathbb{P}(s)}$ and $\mathbb{P}_h(\cdot|s, a)$ denotes a fixed transition kernel.

Example 3.2 (R-contamination model)**.** The uncertainty set under R-contamination model is $\mathcal{P}_h(s, a) := \big\{(1 - R)\mathbb{P}_h(\cdot|s, a) + Rq : q \in \Delta^{|\mathcal{S}|}\big\}$, where $\mathbb{P}_h(\cdot|s, a)$ is a fixed transition kernel.

In the above examples, $\mathbb{P}_h(\cdot|s, a)$ can be understood as the original stationary transition kernel, and the parameters $\rho > 0$ and $R > 0$ characterize the level of uncertainty. In a Markov game with model uncertainty, the state transitions are determined by uncertain transition kernels queried from the uncertainty sets. Therefore, it is possible that a certain transition kernel in the uncertainty set can lead to frequent low-reward state transitions, which are unacceptable to the players. Hence, under model uncertainty, each player aims to learn a robust optimal policy that maximizes its expected accumulated reward in the worst case. Motivated by this intuition, we define the following *robust value function* for the j-th player at state s and time step h under joint policy π. For simplicity of notation, we denote $\mathcal{P} := \bigotimes_{h,s,a} \mathcal{P}_h(s, a)$ as the product of uncertainty sets.

$$\text{(Robust Value Function):}\quad \boldsymbol{V}_{\pi,h}^{(j)}(s) := \inf_{\widetilde{\mathbb{P}}\in\mathcal{P}} \mathbb{E}\Big[\sum_{\ell=h}^{H} r_\ell^{(j)}(s_\ell, a_\ell)\Big| s_h = s, \pi, \widetilde{\mathbb{P}}\Big]. \tag{2}$$

Intuitively, the robust value function characterizes the minimum expected total reward one can obtain over all possible transition kernels in the uncertainty set. We note that the above robust value function is defined for every single player in the Markov game. In particular, the worst-case (adversarial) transition kernels associated with the players' robust value functions are generally different from each other. To deal with model uncertainty, the players aim to achieve a certain equilibrium in terms of the robust value function. Specifically, we define the following robust Nash equilibrium (NE).

Definition 3.3 (Robust NE)**.** A joint policy π is called robust NE if (i) for all h, π_h is a product policy; (ii) for any player j with any policy $\widetilde{\pi}^{(j)}$, we have $\boldsymbol{V}_{\pi,1}^{(j)}(s) \geq \boldsymbol{V}_{\widetilde{\pi}^{(j)}\times\pi^{(\setminus j)},1}^{(j)}(s)$ for all $s \in \mathcal{S}$.

It can be seen that robust NE is similar to the NE defined in Definition 2.1, with the main difference being that robust NE is defined based on the robust value function. However, robust NE is generally more difficult to compute than NE. For example, NE is known to be tractable in zero-sum Markov

games. As a comparison, in a zero-sum Markov game with model uncertainty, the environment model uncertainty can be viewed as a third adversarial player that competes with both players and breaks the zero-sum structure. Therefore, robust NE is not tractable in general, and this further motivates us to define the following tractable surrogate notion of robust correlated equlibrium (CE).

Definition 3.4 (Robust CE)**.** A joint policy π is called a robust CE if for any player j and any stochastic modification $\phi^{(j)}$, it holds that $\boldsymbol{V}_{\pi,1}^{(j)}(s) \geq \boldsymbol{V}_{\phi^{(j)}\circ\pi,1}^{(j)}(s)$ for all states $s \in \mathcal{S}$.

Although robust CE is a straightforward generalization of the standard CE defined in Definition 2.3, it incorporates model uncertainty into the nature of correlated equilibrium and turns out to have more complex structures than the standard CE as we elaborate in the next subsection.

3.2 Properties of Robust Correlated Equilibrium

Stochastic modification is the key element to define CE. In particular, it has been shown that the standard CE defined by stochastic modification is equivalent to that defined by deterministic modification. Our next result shows that robust CE inherits this property.

Proposition 3.5. *In a Markov game with model uncertainty, for any robust CE π and any player j, there exists a deterministic modification $\phi^{(j)}$ such that $\boldsymbol{V}_{\pi,1}^{(j)}(s) = \boldsymbol{V}_{\phi^{(j)}\circ\pi,1}^{(j)}(s)$ for all $s \in \mathcal{S}$.*

This result shows that robust CE can be equivalently defined based on deterministic modifications, which provides a way to simplify the CE notion as there are only finitely many deterministic modifications but uncountable stochastic modifications. and we will leverage this property to build our convergence analysis later. Since robust CE is defined over all stochastic modifications (including deterministic ones), it is non-trivial to establish the above equivalence, and we need to develop the following new techniques.

- Finding the optimal deterministic modification $\phi^{(j)}$ with regard to a robust CE policy π is challenging, since the modification $\phi_h^{(j)}$ at step h depends not only on the past states $s_{1:h}$ and past actions $a_{1:(h-1)}$ but also on player j's current action $a_h^{(j)}$ generated by π_h. Moreover, the optimal choice of $\phi_h^{(j)}$ at step h depends on the optimal choice of $\phi_{h-1}^{(j)}$ for the previous step. These structures motivate us to build an induction over the cases of horizon length $1, 2, ..., H$.

- Specifically, we first rewrite the modified policy $\phi_h^{(j)} \circ \pi_h$ in terms of $\phi_h^{(j)}$ (see eq. (9) in Appendix A) so that the robust value function can be expressed as a functional of $\phi^{(j)}$. To build the induction, we assume that the optimal deterministic $\phi^{(j)}$ exists for the robust value functions with horizon length up to $H-1$. Then, for the robust value function with horizon length H, the optimal deterministic $\phi_H^{(j)}$ for the last step H can be easily found since it only involves the final-step reward $r_H^{(j)}$. After that, we replace $r_{H-1}^{(j)}$ with the surrogate reward $\widetilde{r}_{H-1}^{(j)} = r_{H-1}^{(j)} + \inf_{\mathbb{P}_{H-1}} \mathbb{E}(r_H^{(h)}|s_{1:H-1}, a_{1:H-1}, \phi_H^{(j)} \circ \pi, \mathbb{P}_{H-1})$ which only relies on $s_{1:H-1}, a_{1:H-1}$ (see eq. (11)). Therefore, this robust value function is reduced to that with horizon length $H-1$, for which the optimal deterministic modification exists by the induction assumption. Note that the proof of Lemmas F.2, F.3, F.4, F.6 and F.8 for proving Theorem 4.4 about convergence analysis also use induction, which is different from the induction used for proving Proposition 3.5

Our next result shows that for robust CE, its characterization of equilibrium can be different from that of robust NE and critically depends on the specific environment uncertainty model.

Proposition 3.6. *Robust CE and robust NE have the following relations.*

1. *In any robust Markov game, the set of robust CE includes the set of robust NE, or equivalently, any robust NE is a robust CE.*

2. *There exists a Markov game whose robust CE is not robust NE.*

4 DECENTRALIZED ROBUST V-LEARNING

In this section, we develop a fully decentralized algorithm for finding robust CE of Markov games with model uncertainty. Our algorithm is inspired by the V-learning algorithm for solving standard Markov games (Jin et al., 2022a), and adopts new techniques to address model uncertainty.

4.1 ALGORITHM DESIGN

We let every player j keep a value table $V_h^{(j)} \in \mathbb{R}^{|\mathcal{S}|}$ for each time step h, and denote $\{V_{k,h}^{(j)}\}_{h=1}^H$ as the value tables held by player j in the k-th episode. The main steps of our algorithm consist of a critic step and an actor step. In the critic step, we aim to learn the robust state value function associated with the current policy. To do so, we apply the following robust TD-learning type updates with every state-action transition sample (s, a, s') to update the players' value tables.

$$\widetilde{V}_{k,h}^{(j)}(s) = (1-\alpha_t)\widetilde{V}_{k-1,h}^{(j)}(s) + \alpha_t\left(r_h^{(j)} + \sigma_{\mathcal{P}_h(s,a)}(V_{k-1,h+1}^{(j)}) + \beta_t\right),$$
$$V_{k,h}^{(j)}(s) = \left\{H+1-h, \widetilde{V}_{k,h}^{(j)}(s)\right\},$$

where the first update performs a robust TD type update and the second update performs a simple upper truncation. Here, $\alpha_t > 0$ is a learning rate parameter where $t := N_{k,h}(s)$ denotes that state s has been visited at step h for t times at the beginning of the k-th episode, and the value function mapping $\sigma_{\mathcal{P}_h(s,a)}(\cdot)$ is defined via the following linear program for any value table V.

$$\sigma_{\mathcal{P}_h(s,a)}(V) := \inf_{\widetilde{\mathbb{P}}_h(\cdot|s,a)\in\mathcal{P}_h(s,a)} \langle \widetilde{\mathbb{P}}_h(\cdot|s,a), V(\cdot)\rangle. \tag{3}$$

Intuitively, the above mapping corresponds to the worst-case expected state value of the next state. In particular, when there is no model uncertainty and the transition kernel is $\mathbb{P}_h(\cdot|s,a)$, it reduces to the expected state value at the next state, i.e., $\mathbb{E}_{s'\sim\mathbb{P}_h(\cdot|s,a)}[V(s')]$. Moreover, this linear program can be numerically solved for several important classes of uncertainty sets, as we elaborate below.

Example 4.1 (KL divergence)**.** Consider the uncertainty set $\mathcal{P}_h(s,a)$ defined under the KL divergence in Example 3.1. Then, the linear program (3) reduces to the following one-dimensional optimization problem, as proved in Theorem 1 of (Hu & Hong, 2013).

$$\min_{\alpha\geq 0}\ \alpha \ln \mathbb{E}_{s'\sim\mathbb{P}_h(\cdot|s,a)}\left[e^{V(s')/\alpha}\right] + \alpha\eta. \tag{4}$$

In practice, we can query some samples to approximate the expectation involved in the above one-dimensional problem and solve it to obtain a sample-based estimator $\widehat{\sigma}_{\mathcal{P}_h(s,a)}(V)$.

Example 4.2 (R-contamination model)**.** Consider the uncertainty set $\mathcal{P}_h(s,a)$ defined by the R-contamination model in Example 3.2. Then, the linear program (3) can be approximated by the sample-based estimator $\widehat{\sigma}_{\mathcal{P}_h(s,a)}(V) = R\max_{s\in\mathcal{S}} V(s) + (1-R)V(s')$ (Wang & Zou, 2021).

In the actor step, we leverage the adversarial bandit algorithm developed in (Jin et al., 2022a) to update the current policy. To briefly explain, in step h of episode k, every player j takes an action $a_h^{(j)} \sim \pi_{k,h}^{(j)}(\cdot|s_h)$ and observes an adversarial loss $1 - \frac{r_h^{(j)} + \widehat{\sigma}_{\mathcal{P}_h(s_h,a_h)}(V_{k,h+1}^{(j)})}{H}$, both of which are then fed into the adversarial bandit algorithm to produce the policy $\pi_{k+1,h}^{(j)}(\cdot|s_h)$. The specific updates of this algorithm are shown in Algorithm 2 in Appendix C. In particular, Jin et al. (2022a) proved that it achieves a regret bound in the order of $\widetilde{\mathcal{O}}(B\sqrt{H/t})$ (see Lemma C.1 in the appendix).

The entire decentralized robust V-learning algorithm is summarized in Algorithm 1 below, where we use the estimator $\widehat{\sigma}_{\mathcal{P}_h(s,a)}$ instead of $\sigma_{\mathcal{P}_h(s,a)}$ in the critic update. After obtaining all the policies $\{\pi_{k,h}\}_{k,h}$, the final non-Markov output policy $\hat{\pi}$ is defined by randomly selecting an episode k at each step h and taking an action $a_h \sim \pi_{k,h}$ (see Algorithm 3 in Appendix C for more details).

4.2 CONVERGENCE AND COMPLEXITY ANALYSIS

For any joint policy π, we measure its optimality gap toward achieving exact robust CE as follows, where we define $\boldsymbol{V}_{\phi^*\circ\pi,1}^{(j)}(s) := \max_{\phi^{(j)}} \boldsymbol{V}_{\phi^{(j)}\circ\pi,1}^{(j)}(s)$ as player j's value function associated with

Algorithm 1: Decentralized Robust V-Learning (j-th player)

Initialize: Set $\widetilde{V}_{1,h}^{(j)}(s) = V_{1,h}^{(j)}(s) = H+1-h$, $\pi_{1,h}(a|s) = \frac{1}{|\mathcal{A}|}$, $N_{1,h}^{(j)}(s) = 0$ for all s, a, h

for episode $k = 1, \ldots, K$ **do**

Observe initial state s_1, $V_{k,H+1}^{(j)}(s) = 0$ for all s

for step $h = 1, \ldots, H$ **do**

Take action $a_h^{(j)} \sim \pi_{k,h}^{(j)}(\cdot|s_h)$

Transfer to next state $s_{h+1} \sim \widetilde{\mathbb{P}}_h(\cdot|s_h, a_h)$ with $\widetilde{\mathbb{P}}_h \in \mathcal{P}_h(s_h, a_h)$

Let $\widetilde{V}_{k+1,h}^{(j)} \leftarrow \widetilde{V}_{k,h}^{(j)}$, $V_{k+1,h}^{(j)} \leftarrow V_{k,h}^{(j)}$, $\pi_{k+1,h}^{(j)} \leftarrow \pi_{k,h}^{(j)}$

Receive reward $r_h^{(j)}$ and set $t := N_{k+1,h}^{(j)}(s_h) \leftarrow N_{k,h}^{(j)}(s_h) + 1$

$$\widetilde{V}_{k+1,h}^{(j)}(s_h) = (1-\alpha_t)\widetilde{V}_{k,h}^{(j)}(s_h) + \alpha_t \left(r_h^{(j)} + \widehat{\sigma}_{\mathcal{P}_h(s_h,a_h)}(V_{k,h+1}^{(j)}) + \beta_t^{(j)} \right) \tag{5}$$

$$V_{k+1,h}^{(j)}(s_h) = \min\{H+1-h, \widetilde{V}_{k+1,h}^{(j)}(s_h)\} \tag{6}$$

$$\pi_{k+1,h}^{(j)}(\cdot|s_h) = \texttt{ADV\_BANDIT}\left(t, a_h, 1 - \frac{r_h^{(j)} + \widehat{\sigma}_{\mathcal{P}_h(s_h,a_h)}(V_{k,h+1}^{(j)})}{H}, \pi_{k,h}^{(j)}(\cdot|s_h)\right) \tag{7}$$

end

end

Output: Joint policy $\hat{\pi}$ defined by Algorithm 3 in Appendix C with hyperparameters α_t^i.

the policy π modified by player j's best-response modification ϕ^*.

$$\text{(Optimality gap):} \quad \max_{j\in[J]} \max_{s\in\mathcal{S}} \left[\boldsymbol{V}_{\phi^*\circ\pi,1}^{(j)}(s) - \boldsymbol{V}_{\pi,1}^{(j)}(s) \right] \geq 0. \tag{8}$$

In particular, policy π is a robust CE if the gap vanishes. We also need the following definitions to characterize the impact of model uncertainty on Algorithm 1's convergence rate.

Definition 4.3. Regarding the uncertainty sets $\{\mathcal{P}_h(s,a)\}_{h,s,a}$, value function mapping $\sigma_{\mathcal{P}_h(s,a)}$ and state exploration probability, we define the following quantities.

- *Uncertainty diameter:* $D := \max_{h,s,a,a'} \max_{\mathbb{P}\in\mathcal{P}_h(s,a), \widetilde{\mathbb{P}}\in\mathcal{P}_h(s,a')} \|\mathbb{P}(\cdot) - \widetilde{\mathbb{P}}(\cdot)\|_\infty$.
- *Estimation error:* $\mathfrak{e} := \sup_{h,s,a,V} \left| \sigma_{\mathcal{P}_h(s,a)}(V) - \widehat{\sigma}_{\mathcal{P}_h(s,a)}(V) \right|$, where the supremum is taken over all bounded value tables that satisfy $0 \leq V(s) \leq H+1$ for all s.
- *State exploration:* $p_{\min} := \min_{s,h,k} \mathbb{P}(s_{k,h} = s)$, which denotes the minimum probability of visiting an arbitrary state s at any step h of any episode k.

The uncertainty diameter D defined above characterizes the diameter of the uncertainty set $\mathcal{P}_h$. That is, a larger D means that the transition kernel $\mathbb{P}_h$ can change over a wider range and therefore induces larger uncertainty. For example, for the uncertainty set defined by the R-contamination model in Example 3.2, the uncertainty diameter is analytically given by $D = R \max\left\{ \max_{s'} \mathbb{P}_h(s'|s,a), 1 - \min_{s'} \mathbb{P}_h(s'|s,a) \right\}$, which monotonically increases with regard to the uncertainty set parameter R. We obtain the following convergence rate of decentralized robust V-learning.

Theorem 4.4. *Let $S := |\mathcal{S}|$ and $A := \max_{1\leq j\leq m} |\mathcal{A}^{(j)}|$ correspond to the size of the state space and action space, respectively. Choose $\beta_t^{(j)}$, α_t and α_t^i according to eqs.* (19)-(21)*. Let the diameter of uncertainty set D satisfy $D \leq \max\{\frac{p_{\min}}{H}, \frac{\epsilon}{SH^2}\}$. The output policy $\hat{\pi}$ produced by Algorithm 1 satisfies the following convergence rate with probability at least $1 - c\delta$ for some constant $c > 0$.*

1. If $p_{\min} > \frac{\epsilon}{SH}$, then

$$\max_{j\in[J]} \max_{s\in\mathcal{S}} \left(\boldsymbol{V}_{\phi^*\circ\hat{\pi},1}^{(j)}(s) - \boldsymbol{V}_{\hat{\pi},1}^{(j)}(s) \right) \leq \mathcal{O}\left(\frac{H}{p_{\min} - DH} \left(A\sqrt{\frac{H^3 S}{K} \ln \frac{mKHSA^2}{\delta}} + \mathfrak{e} \right) \right).$$

To achieve an ϵ gap, we set $\mathfrak{e} = \mathcal{O}(\frac{\epsilon p_{\min}}{H})$ and require $K = \widetilde{\mathcal{O}}(SA^2H^5 p_{\min}^{-2} \epsilon^{-2})$ episodes.

2. If $p_{\min} \leq \frac{\epsilon}{SH}$, then

$$\max_{j\in[J]} \max_{s\in\mathcal{S}} \left(\boldsymbol{V}^{(j)}_{\phi^*\circ\hat{\pi},1}(s) - \boldsymbol{V}^{(j)}_{\hat{\pi},1}(s)\right) \leq 5DSH^2 + \mathcal{O}\Big(H\Big(A\sqrt{\frac{H^3S}{K}\ln\frac{mKHSA^2}{\delta}} + \mathfrak{e}\Big)\Big).$$

To achieve an ϵ gap, we set $\mathfrak{e} = \mathcal{O}(\frac{\epsilon}{H})$ and require $K = \widetilde{\mathcal{O}}(SA^2H^5\epsilon^{-2})$ episodes.

Theorem 4.4 characterizes the convergence rate and episode complexity of decentralized robust V-learning. We note that the optimality gap adopted in the above theorem takes the maximum over all the states and hence is stronger than that used in the original V-learning (Jin et al., 2022b). This is because we need to develop new techniques to address the state transition uncertainty caused by the environment model uncertainty. As a result, the environment model uncertainty diameter D should not be too large compared to the state exploration probability $p_{\min}$ and the target accuracy ϵ.

When there is only a single player ($m = 1$), we can prove that every robust CE policy achieves the optimal robust value function, by noticing that for any given policies π, μ over the action space $\mathcal{A}$ there always exists a stochastic modification ϕ such that $\phi \circ \pi(a) = \mu(a)$ for all $a \in \mathcal{A}$. Therefore, the robust V-learning algorithm can be applied to single-agent reinforcement learning to address model uncertainty. We obtain the following corollary.

Corollary 4.5. *In the case of a single player, the output policy $\hat{\pi}$ produced by Algorithm 1 achieves an approximate optimal robust value function at the same convergence rate as that in Theorem 4.4.*

Technical novelty of Theorem 4.4. Our analysis leverages the following technical developments to address model uncertainty and establish the convergence rate.

- To address model uncertainty, our decentralized robust V-learning algorithm adopts the worst-case expected value function estimator $\widehat{\sigma}_{\mathcal{P}_h(s,a)}(V)$ in the critic update, as opposed to the exact value $V(s)$ used in the standard V-learning. Such a nonlinear operator allows us to track the robust value function in the analysis. In particular, we developed various important properties of this operator in Lemma F.1, including boundedness, monotonicity, etc., which are crucial to establish the key Lemmas F.2, F.3, F.4 and F.6 that lead to the desired convergence rate result.
- The proof of the original V-learning algorithm (Jin et al., 2022a) tracks the upper bound of the optimality gap $\delta_{k,h} := V_{k,h}(s_{k,h}) - \underline{V}_{k,h}(s_{k,h})$ at a single state and builds a recursion on it. This approach cannot be applied to our case as this gap turns into an uncertainty form $\sigma_{\mathcal{P}_h(s,a)}(V_{k,h+1}) - \sigma_{\mathcal{P}_h(s,a)}(\underline{V}_{k,h+1})$, which inevitably involves all possible states. To address this issue, we decompose this term as $\sigma_{\mathcal{P}_h(s,a)}(V_{k,h+1}) - \sigma_{\mathcal{P}_h(s,a)}(\underline{V}_{k,h+1}) = \delta_{k,h+1} + \sigma_{\mathcal{P}_h(s,a)}(V_{k,h+1}) - \sigma_{\mathcal{P}_h(s,a)}(\underline{V}_{k,h+1}) - \delta_{k,h+1}$ to link it to the desired term $\delta_{k,h+1}$ (see eq. (12)). Consequently, we need to solve a more challenging recursion that we build in Lemma F.8.
- The decomposition mentioned in the previous bullet point involves an error term $\sigma_{\mathcal{P}_h(s,a)}(V_{k,h+1}) - \sigma_{\mathcal{P}_h(s,a)}(\underline{V}_{k,h+1}) - \delta_{k,h+1}$ that critically depends on the level of uncertainty. For example, this error term vanishes when there is no model uncertainty. We develop an upper bound of this error term in Lemma F.7 by leveraging the uncertainty diameter (Definition 4.3) and introducing a stronger convergence metric $\Delta^{(j)}_{k,h} := \sum_{s\in\mathcal{S}} \left(V^{(j)}_{k,h}(s) - \underline{V}^{(j)}_{k,h}(s)\right)$ compared to $\delta_{k,h}$.
- To derive the desired convergence rate, we build a recursion on the convergence metric $\{\sum_{k=1}^{K} \Delta^{(j)}_{k,h}\}_{h\in[H]}$ in Lemma F.9. The key challenge in solving this recursion is to rewrite it as a vectorized linear system that involves an upper triangular Toeplitz matrix, whose spectrum can be characterized analytically and used to derive the final result.

5 Experiments

In this section, we compare the performance of V-learning and robust V-learning in the two-player coordination game described in the proof of Proposition 3.6. Specifically, the game consists of five states and each player has two actions. The state transition probability at step $h = 1$ is shown in Figure 2, and the state remains unchanged at step $h = 2$. The two-player reward function is defined as $r(s_0, a) = [0.5, 0.5]$, $r(s_1, a) = [0, 1]$, $r(s_2, a) = [1, 0]$, $r(s_3, a) = [0.95, 0.95]$ and $r(s_4, a) = [0, 0]$ for all actions $a \in \mathcal{A}$. We consider the following types of uncertainty models.

1. Discrete uncertainty model $\mathcal{P}_h = \{\mathbb{P}_{h,p} : p \in \{0, \frac{5}{14}\}\}$: In this case, the analytical form of the value function mapping $\sigma_{\mathcal{P}_h(s_h,a_h)}(V)$ can be explicitly calculated.

2. KL divergence model in Example 3.1: We set the centroid transition kernel to be $\mathbb{P}_h = \mathbb{P}_{1,0.1}$ and choose uncertainty level parameter $\rho = 0.1$.
3. R-contamination model in Example 3.2: We set the centroid transition kernel to be $\mathbb{P}_h = \mathbb{P}_{1,0.1}$ and choose uncertainty level parameter $R = 0.01$.

We generate episodic data by randomly querying transition kernels from these uncertainty sets in every time step through a non-stationary process. Then we feed the generated data to both the V-learning algorithm and our robust V-learning algorithm. The performance of the output policy is evaluated by estimating the optimality gap used in our theoretical analysis (defined in eq. (8)). Specifically, to evaluate an approximated optimality gap for the KL divergence and R-contamination uncertainty models, we sample multiple transition kernels uniformly at random from these uncertainty sets and evaluate the optimality gap over them. More details on the experiment setup and hyper-parameter choices are described in Appendix G.

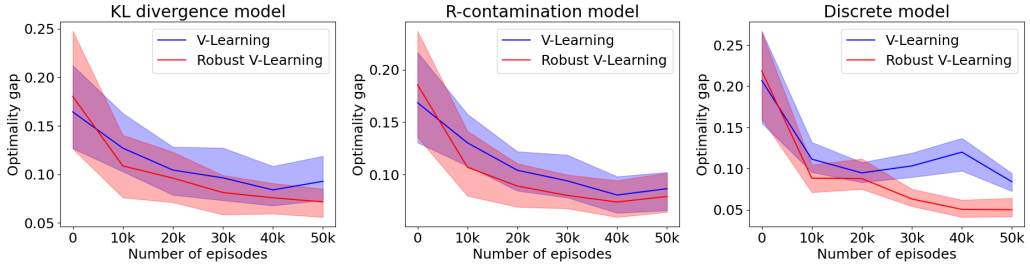

Figure 1: Comparison of estimated optimality gap of the policies produced by V-learning and robust V-learning. The optimality gap we estimate is $\max_{j\in[J]}[\boldsymbol{V}^{(j)}_{\phi^*\circ\pi_{K+1},1}(s_4) - \boldsymbol{V}^{(j)}_{\pi_{K+1},1}(s_4)]$.

Figure 1 shows the results on the estimated optimality gap of the policies produced by both algorithms, where each curve consists of 30 repetitions. It can be observed that robust V-learning consistently outperforms V-learning and achieves a smaller optimality gap under all three types of uncertainty models. This demonstrate that robust V-learning is good at computing approximate robust CE under general environment model uncertainty. In particular, under the discrete uncertainty model, robust V-learning obtains the most substantial improvement over V-learning. This is because the experiment setup chooses a high level of uncertainty and we can compute $\sigma_{\mathcal{P}_h}(V)$ exactly. For both the KL divergence model and R-contamination model, we choose lower levels of uncertainty and approximately compute $\sigma_{\mathcal{P}_h}(V)$. Hence, the corresponding performance gaps between robust V-learning and V-learning are smaller.

6 Conclusion

In this work, we proposed a new and tractable notion of robust correlated equilibrium for Markov games with environment model uncertainty. We showed that the robust correlated equilibrium has a simple modification structure, and its characterization of equilibrium critically depends on the environment model uncertainty. Moreover, we proposed the first fully-decentralized robust V-learning algorithm for computing such robust correlated equilibrium and established a polynomial sample complexity for computing an approximate robust correlated equilibrium. We believe this work provides an initial solution to competitive multi-agent reinforcement learning in uncertain environment, and an interesting future direction is to explore if it is possible to establish convergence of the algorithm under relaxed requirements on the uncertainty diameter.

References

Yu Bai and Chi Jin. Provable self-play algorithms for competitive reinforcement learning. In International conference on machine learning, pp. 551–560. PMLR, 2020.

Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. Swarm Intelligence, 7(1):1–41, 2013.

Xiaotie Deng, Yuhao Li, David Henry Mguni, Jun Wang, and Yaodong Yang. On the complexity of computing markov perfect equilibrium in general-sum stochastic games. arXiv preprint arXiv:2109.01795, 2021.

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

Arlington M Fink. Equilibrium in a stochastic n-person game. Journal of science of the hiroshima university, series ai (mathematics), 28(1):89–93, 1964.

Amy Greenwald, Keith Hall, Roberto Serrano, et al. Correlated q-learning. In ICML, volume 3, pp. 242–249, 2003.

Thomas Dueholm Hansen, Peter Bro Miltersen, and Uri Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. Journal of the ACM (JACM), 60(1):1–16, 2013.

Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. Journal of machine learning research, 4(Nov):1039–1069, 2003.

Zhaolin Hu and L Jeff Hong. Kullback-leibler divergence constrained distributionally robust optimization. Available at Optimization Online, pp. 1695–1724, 2013.

Feng Huang, Ming Cao, and Long Wang. Optimal control of robust team stochastic games. arXiv preprint arXiv:2105.07405, 2021.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? Advances in neural information processing systems, 31, 2018.

Chi Jin, Qinghua Liu, Yuanhao Wang, and Tiancheng Yu. V-learning–a simple, efficient, decentralized algorithm for multiagent rl. In ICLR 2022 Workshop on Gamification and Multiagent Solutions, 2022a.

Yujia Jin, Vidya Muthukumar, and Aaron Sidford. The complexity of infinite-horizon general-sum stochastic games. arXiv preprint arXiv:2204.04186, 2022b.

Erim Kardeş, Fernando Ordóñez, and Randolph W Hall. Discounted robust stochastic games and an application to queueing control. Operations research, 59(2):365–382, 2011.

Stefanos Leonardos, Will Overman, Ioannis Panageas, and Georgios Piliouras. Global convergence of multi-agent policy gradient in markov potential games. ArXiv:2106.01969, 2021.

Jialian Li, Tongzheng Ren, Dong Yan, Hang Su, and Jun Zhu. Policy learning for robust markov decision process with a mismatched generative model. arXiv preprint arXiv:2203.06587, 2022a.

Tianjiao Li, Ziwei Guan, Shaofeng Zou, Tengyu Xu, Yingbin Liang, and Guanghui Lan. Faster algorithm and sharper analysis for constrained markov decision process. arXiv preprint arXiv:2110.10351, 2021.

Yan Li, Tuo Zhao, and Guanghui Lan. First-order policy optimization for robust markov decision process. arXiv preprint arXiv:2209.10579, 2022b.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In Machine learning proceedings 1994, pp. 157–163. Elsevier, 1994.

Michael L Littman et al. Friend-or-foe q-learning in general-sum games. In ICML, volume 1, pp. 322–328, 2001.

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, pp. 7001–7010. PMLR, 2021.

Weichao Mao and Tamer Başar. Provably efficient reinforcement learning in decentralized general-sum markov games. Dynamic Games and Applications, pp. 1–22, 2022.

Hervé Moulin and J-P Vial. Strategically zero-sum games: the class of games whose completely mixed equilibria cannot be improved upon. International Journal of Game Theory, 7(3):201–221, 1978.

Ariel Neufeld and Julian Sester. Robust q-learning algorithm for markov decision processes under wasserstein uncertainty. arXiv preprint arXiv:2210.00898, 2022.

Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. Operations Research, 53(5):780–798, 2005.

Arnab Nilim and Laurent Ghaoui. Robustness in markov decision problems with uncertain transition matrices. Advances in neural information processing systems, 16, 2003.

Aurko Roy, Huan Xu, and Sebastian Pokutta. Reinforcement learning under model mismatch. Advances in neural information processing systems, 30, 2017.

Jay K Satia and Roy E Lave Jr. Markovian decision processes with uncertain transition probabilities. Operations Research, 21(3):728–740, 1973.

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

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. nature, 529(7587):484–489, 2016.

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.

Ziang Song, Song Mei, and Yu Bai. When can we learn general-sum markov games with a large number of players sample-efficiently? arXiv preprint arXiv:2110.04184, 2021.

Yue Wang and Shaofeng Zou. Online robust reinforcement learning with model uncertainty. Advances in Neural Information Processing Systems, 34:7193–7206, 2021.

Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. Mathematics of Operations Research, 38(1):153–183, 2013.

Kaiqing Zhang, Tao Sun, Yunzhe Tao, Sahika Genc, Sunil Mallya, and Tamer Basar. Robust multi-agent reinforcement learning with model uncertainty. Advances in neural information processing systems, 33:10571–10583, 2020.

Runyu Zhang, Zhaolin Ren, and Na Li. Gradient play in multi-agent markov stochastic games: Stationary points and convergence. ArXiv:2106.00198, 2021.

Appendix

Table of Contents

A Proof of Proposition 3.5

First, we re-state Proposition 3.5 here.

Proposition A.1. *In a Markov game with model uncertainty, any robust CE policy π can be achieved by deterministic modifications, i.e., for any player j there exists a deterministic modification $\phi^{(j)}$ such that $\boldsymbol{V}^{(j)}_{\pi,1}(s) = \boldsymbol{V}^{(j)}_{\phi^{(j)}\circ\pi,1}(s)$.*

Proof. We will prove this proposition for a more general setting where each reward $r_h^{(j)} = r_h^{(j)}(s_{1:h}, a_{1:h})$ relies on all the past and current states $s_{1:h} := \{s_{h'}\}_{h'=1}^h$ and actions $a_{1:h} := \{a_{h'}\}_{h'=1}^h$. Then the conclusion directly applies to the special case of interest where $r_h^{(j)} = r_h^{(j)}(s_h, a_h)$.

We will find $\phi^{(j)}$ by applying mathematical induction to the horizon H.

When $H = 1$, the MDP does not involve transition kernel, so

$$
\begin{aligned}
\boldsymbol{V}^{(j)}_{\phi^{(j)}\circ\pi,1}(s) &= \sum_{a_1} (\phi_1^{(j)} \circ \pi_1)(a_1|s) r_1^{(j)}(s, a_1) \\
&\overset{(i)}{=} \sum_{a_1, \widetilde{a}_1^{(j)}} \phi_1^{(j)}(a_1^{(j)}|s, \widetilde{a}_1^{(j)}) \pi_1([\widetilde{a}_1^{(j)}, a_1^{(\backslash j)}]|s) r_1^{(j)}(s, a_1) \\
&= \sum_{\widetilde{a}_1^{(j)}} \Big(\sum_{a_1^{(j)}} \phi_1^{(j)}(a_1^{(j)}|s, \widetilde{a}_1^{(j)}) \sum_{a_1^{(\backslash j)}} \pi_1([\widetilde{a}_1^{(j)}, a_1^{(\backslash j)}]|s) r_1^{(j)}(s, a_1) \Big) \\
&\overset{(ii)}{\leq} \sum_{\widetilde{a}_1^{(j)}} \Big(\max_{a_1^{(j)}} \sum_{a_1^{(\backslash j)}} \pi_1([\widetilde{a}_1^{(j)}, a_1^{(\backslash j)}]|s) r_1^{(j)}(s, a_1) \Big),
\end{aligned}
$$

where (i) uses the following formula that directly follows from the definition of stochastic modification $\phi^{(j)}$, and (ii) becomes "=" using the deterministic modification such that $\phi_1^{(j)}(a_1^{(j)}|s, \widetilde{a}_1^{(j)}) := 1$

for a certain $a_1^{(j)} \in \arg\max_{a_1^{(j)}} \sum_{a_1^{(\setminus j)}} \pi_1([\widetilde{a}_1^{(j)}, a_1^{(\setminus j)}]|s))r_1^{(j)}(s, a_1)$.

$$(\phi_h^{(j)} \circ \pi_h)\big(a_h|s_{1:h}, a_{1:h-1}\big) = \sum_{\widetilde{a}_h^{(j)}} \phi_h^{(j)}(a_h^{(j)}|s_{1:h}, a_{1:h-1}, \widetilde{a}_h^{(j)})\pi_h([\widetilde{a}_h^{(j)}, a_h^{(\setminus j)}]|s_{1:h}, a_{1:h-1}). \quad (9)$$

This proves the existence of the optimal deterministic solution $\phi^{(j)}$ for $H = 1$. Suppose it also exists for horizon $H - 1$. Then it suffices to prove the existence of $\phi^{(j)}$ for H as follows.

$$\begin{aligned}
&\boldsymbol{V}^{(j)}_{\phi^{(j)}\circ\pi,1}(s): \\
&= \inf_{\mathbb{P}\in\mathcal{P}} \mathbb{E}\Big[\sum_{h=1}^{H} r_h^{(j)}(s_{1:h}, a_{1:h})\Big| s_1 = s, \phi^{(j)} \circ \pi, \mathbb{P}\Big] \\
&= \inf_{\mathbb{P}_{1:H-2}\in\mathcal{P}_{1:H-2}} \Bigg(\mathbb{E}\Big[\sum_{h=1}^{H-1} r_h^{(j)}(s_{1:h}, a_{1:h})\Big| s_1 = s, \{\phi_h^{(j)} \circ \pi_h\}_{h=1}^{H-1}, \mathbb{P}_{1:H-2}\Big] \\
&\quad + \inf_{\mathbb{P}_{H-1}\in\mathcal{P}_{H-1}} \mathbb{E}\Big[r_H^{(j)}(s_{1:H}, a_{1:H})\Big| s_1 = s, \phi^{(j)} \circ \pi, \mathbb{P}_{1:H-1}\Big]\Bigg) \\
&= \inf_{\mathbb{P}_{1:H-2}\in\mathcal{P}_{1:H-2}} \Bigg(\mathbb{E}\Big[\sum_{h=1}^{H-1} r_h^{(j)}(s_{1:h}, a_{1:h})\Big| s_1 = s, \{\phi_h^{(j)} \circ \pi_h\}_{h=1}^{H-1}, \mathbb{P}_{1:H-2}\Big] \\
&\quad + \sum_{s_{1:H-1}, a_{1:H-1}} \Pr(s_{1:H-1}, a_{1:H-1}|s_1 = s, \{\phi_h^{(j)} \circ \pi_h\}_{h=1}^{H-1}, \mathbb{P}_{1:H-2}) \\
&\quad \inf_{\substack{\mathbb{P}_{H-1}(\cdot|s_{H-1},a_{H-1}) \\ \in\mathcal{P}_{H-1}(s_{H-1},a_{H-1})}} \sum_{s_H} \mathbb{P}_{H-1}(s_H|s_{H-1}, a_{H-1}) \sum_{a_H} (\phi_H^{(j)} \circ \pi_H)(a_H|s_{1:H}, a_{1:H-1}) r_H^{(j)}(s_{1:H}, a_{1:H})\Bigg) \\
&\overset{(i)}{=} \inf_{\mathbb{P}_{1:H-2}\in\mathcal{P}_{1:H-2}} \Bigg(\mathbb{E}\Big[\sum_{h=1}^{H-1} r_h^{(j)}(s_{1:h}, a_{1:h})\Big| s_1 = s, \{\phi_h^{(j)} \circ \pi_h\}_{h=1}^{H-1}, \mathbb{P}_{1:H-2}\Big] \\
&\quad + \sum_{s_{1:H-1}, a_{1:H-1}} \Pr(s_{1:H-1}, a_{1:H-1}|s_1 = s, \{\phi_h^{(j)} \circ \pi_h\}_{h=1}^{H-1}, \mathbb{P}_{1:H-2}) \\
&\quad \inf_{\substack{\mathbb{P}_{H-1}(\cdot|s_{H-1},a_{H-1}) \\ \in\mathcal{P}_{H-1}(s_{H-1},a_{H-1})}} \sum_{s_H} \mathbb{P}_{H-1}(s_H|s_{H-1}, a_{H-1}) \\
&\quad \sum_{a_H, \widetilde{a}_H^{(j)}} \phi_H^{(j)}(a_H^{(j)}|s_{1:H}, a_{1:H-1}, \widetilde{a}_H^{(j)}) \pi_H([\widetilde{a}_H^{(j)}, a_H^{(\setminus j)}]|s_{1:H}, a_{1:H-1}) r_H^{(j)}(s_{1:H}, a_{1:H})\Bigg) \\
&= \inf_{\mathbb{P}_{1:H-2}\in\mathcal{P}_{1:H-2}} \Bigg(\mathbb{E}\Big[\sum_{h=1}^{H-1} r_h^{(j)}(s_{1:h}, a_{1:h})\Big| s_1 = s, \{\phi_h^{(j)} \circ \pi_h\}_{h=1}^{H-1}, \mathbb{P}_{1:H-2}\Big] \\
&\quad + \sum_{s_{1:H-1}, a_{1:H-1}} \Pr(s_{1:H-1}, a_{1:H-1}|s_1 = s, \{\phi_h^{(j)} \circ \pi_h\}_{h=1}^{H-1}, \mathbb{P}_{1:H-2}) \\
&\quad \inf_{\substack{\mathbb{P}_{H-1}(\cdot|s_{H-1},a_{H-1}) \\ \in\mathcal{P}_{H-1}(s_{H-1},a_{H-1})}} \sum_{s_H, \widetilde{a}_H^{(j)}} \mathbb{P}_{H-1}(s_H|s_{H-1}, a_{H-1}) \sum_{a_H^{(j)}} \phi_H^{(j)}(a_H^{(j)}|s_{1:H}, a_{1:H-1}, \widetilde{a}_H^{(j)}) \\
&\quad \sum_{a_H^{(\setminus j)}} \pi_H([\widetilde{a}_H^{(j)}, a_H^{(\setminus j)}]|s_{1:H}, a_{1:H-1}) r_H^{(j)}(s_{1:H}, a_{1:H})\Bigg) \\
&\overset{(ii)}{\leq} \inf_{\mathbb{P}_{1:H-2}\in\mathcal{P}_{1:H-2}} \Bigg(\mathbb{E}\Big[\sum_{h=1}^{H-1} r_h^{(j)}(s_{1:h}, a_{1:h})\Big| s_1 = s, \{\phi_h^{(j)} \circ \pi_h\}_{h=1}^{H-1}, \mathbb{P}_{1:H-2}\Big] \\
&\quad + \sum_{s_{1:H-1}, a_{1:H-1}} \Pr(s_{1:H-1}, a_{1:H-1}|s_1 = s, \{\phi_h^{(j)} \circ \pi_h\}_{h=1}^{H-1}, \mathbb{P}_{1:H-2}) \inf_{\substack{\mathbb{P}_{H-1}(\cdot|s_{H-1},a_{H-1}) \\ \in\mathcal{P}_{H-1}(s_{H-1},a_{H-1})}}
\end{aligned}$$

$$\sum_{s_H, \widetilde{a}_H^{(j)}} \mathbb{P}_{H-1}(s_H|s_{H-1}, a_{H-1}) \max_{a_H^{(j)}} \sum_{a_H^{(\backslash j)}} \pi_H([\widetilde{a}_H^{(j)}, a_H^{(\backslash j)}]|s_{1:H}, a_{1:H-1}) r_H^{(j)}(s_{1:H}, a_{1:H}) \Bigg)$$

$$\overset{(iii)}{=} \inf_{\mathbb{P}_{1:H-2} \in \mathcal{P}_{1:H-2}} \mathbb{E}\Big[\sum_{h=1}^{H-2} r_h^{(j)}(s_{1:h}, a_{1:h}) + \widetilde{r}_{H-1}^{(j)}(s_{1:H-1}, a_{1:H-1}) \Big| s_1 = s, \{\phi_h^{(j)} \circ \pi_h\}_{h=1}^{H-1}, \mathbb{P}_{1:H-2} \Big], \tag{10}$$

where we denote $\mathcal{P}_{1:h} := \{\mathcal{P}_{h'}\}_{h'=1}^{h}$ and $\mathbb{P}_{1:h} := \{\mathbb{P}_{h'}\}_{h'=1}^{h}$, (i) uses eq. (9), (ii) becomes "=" using the deterministic modification $\phi_H^{(j)}$ such that $\phi_H^{(j)}(a_H^{(j)}|s_{1:H}, a_{1:H-1}, \widetilde{a}_H^{(j)}) := 1$ for a certain $a_H^{(j)} \in \arg\max_{a_H^{(j)}} \sum_{a_H^{(\backslash j)}} \pi_H([\widetilde{a}_H^{(j)}, a_H^{(\backslash j)}]|s_{1:H}, a_{1:H-1}) r_H^{(j)}(s_{1:H}, a_{1:H})$, and (iii) denotes the following surrogate reward at step $H-1$,

$$\widetilde{r}_{H-1}^{(j)}(s_{1:H-1}, a_{1:H-1}) = r_{H-1}^{(j)}(s_{1:H-1}, a_{1:H-1}) + \inf_{\substack{\mathbb{P}_{H-1}(\cdot|s_{H-1}, a_{H-1}) \\ \in \mathcal{P}_{H-1}(s_{H-1}, a_{H-1})}} \sum_{s_H, \widetilde{a}_H^{(j)}} \mathbb{P}_{H-1}(s_H|s_{H-1}, a_{H-1})$$

$$\max_{a_H^{(j)}} \sum_{a_H^{(\backslash j)}} \pi_H([\widetilde{a}_H^{(j)}, a_H^{(\backslash j)}]|s_{1:H}, a_{1:H-1}) r_H^{(j)}(s_{1:H}, a_{1:H}). \tag{11}$$

Note that eq. (10) can be seen as the value function with horizon $H-1$, so there are deterministic modifications $\phi_h^{(j)*}$ for $h \in [H-1]$ that maximize eq. (10), which along with $\phi_H^{(j)}$ above forms the deterministic modification $\phi^{(j)} := \{\phi_h^{(j)}\}_{h \in [H]}$ that maximizes $\boldsymbol{V}_{\phi^{(j)} \circ \pi, 1}^{(j)}(s)$. This completes the proof. □

B Proof of Proposition 3.6

Proposition 3.6. *Robust CE and robust NE have the following relations.*

1. *In any robust Markov game, the set of robust CE includes the set of robust NE, or equivalently, any robust NE is a robust CE.*
2. *There exists a Markov game whose robust CE is not robust NE.*

Proof. First, we will prove item 1 that the set of robust CE always includes the set of robust NE. This part of proof is directly taken from Proposition 9 (Jin et al., 2022a) with changing the V-function to the robust V-function. Let $\pi = \pi_1 \times \pi_2 \times \cdots \times \pi_m$ be a robust Nash equilibrium; then

$$\max_{\phi_i} \boldsymbol{V}_{(\phi_i \circ \pi_i) \times \pi^{(\backslash i)}}^{(i)}(s)$$

$$\overset{(i)}{=} \max_{\pi_i'} \boldsymbol{V}_{\pi_i' \times \pi^{(\backslash i)}}^{(i)}(s)$$

$$\overset{(ii)}{\leq} \boldsymbol{V}_{\pi}^{(i)}(s),$$

where (i) is because that π is a product policy and (ii) applies the definition of robust Nash equilibrium (see Definition 2.1). It implies that π is also a robust CE by Definition 2.3.

Next, we prove item 2. It suffices to give an example of a Markov game setting and a robust CE policy π that is not NE. Consider a two-player coordination game in which there are five states $\mathcal{S} = \{s_i\}_{i=0}^4$ and each player has two actions $\mathcal{A} = \{a_i^{(1)}\}_{i=0}^1 \times \{a_i^{(2)}\}_{i=0}^1$. At time step $h = 1$, Figure 2 depicts the transition kernel $\mathbb{P}_{1,p}$ parameterized by a parameter $p \in [0, \frac{1}{2})$. At time step $h = 2$, we set the transition kernel $\mathbb{P}_{2,p}(s|s, a) = 1$ for all s and a, i.e., players stay in their current state no matter what actions are taken. The rewards of both players are set as $r(s_0, a) = [0.5, 0.5]$, $r(s_1, a) = [0, 1]$, $r(s_2, a) = [1, 0]$, $r(s_3, a) = [0.95, 0.95]$, and $r(s_4, a) = [0, 0]$ for any action $a \in \mathcal{A}$. The initial state is fixed to be $s = s_4$. We consider the uncertainty set $\mathcal{P}_h = \{\mathbb{P}_{h,p} : p \in (\frac{10}{29}, \frac{1}{2})\}$ where there are two robust NE, i.e., $\pi_1(a = [0, 1]|s = s_4) = 1$ and $\pi_1(a = [1, 0]|s = s_4) = 1$ (π_2 can be arbitrary). Moreover, any convex combination of these two policies is a robust CE but not robust CE.

□

C POLICIES IN ALGORITHM 1

In this section, we will elaborate how is $\pi_{k,h}$ in Algorithm 1 obtained from the adversarial bandit algorithm, and the definition of the output policy $\hat{\pi}$.

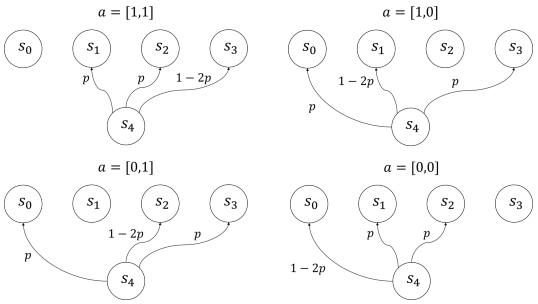

Figure 2: Transition kernel at $h = 1$

Obtaining $\pi_{k,h}$: In Algorithm 1, to output robust equilibrium (robust CE), we adopt V-learning algorithm (Jin et al., 2022a) for single-agent adversarial bandit to update the current policy. To elaborate, we denote the i-th iteration of this algorithm as $\pi_{i+1}(\cdot) \leftarrow \texttt{ADV\_BANDIT}(b_i, \ell_i(\cdot))$, where the player takes action $b_i \in \mathcal{B}$ following its own policy $\pi_i(\cdot)$ obtained from its previous iteration and observes the noisy bandit-feedback $\ell_i(b_i)$ with the loss function ℓ_i selected by the adversary. The procedure of implementing ADV_BANDIT algorithm for multiple iterations is shown in Algorithm 6 of (Jin et al., 2022a) and we extracted the i-th iteration as shown in the following Algorithm 2.

Algorithm 2: Adversarial bandit algorithm (ADV_BANDIT)

Input: Iteration index i, action $\widetilde{b}$ and the corresponding bandit-feedback $\ell(\widetilde{b})$, the previous policy π_i.

for each action $b \in \mathcal{B}$ **do**

$\hat{\ell}_i(\widetilde{b}|b) \leftarrow \frac{\pi_i(b)\ell(\widetilde{b})}{\pi_i(\widetilde{b})+\gamma_i}$

$\hat{\ell}_i(b'|b) \leftarrow 0, \forall b' \in \mathcal{B}/\{\widetilde{b}\}$

$\widetilde{\pi}(\cdot|b) \propto \exp\big[-\frac{\eta_i}{w_i}\sum_{j=1}^{i} w_j \hat{\ell}_j(\cdot|b)\big]$, where $\hat{\ell}_j$ is obtained from the j-th iteration of ADV_BANDIT algorithm

end

Output: π_{i+1} obtained by solving the linear equation $\pi_{i+1}(\cdot) = \sum_{b\in\mathcal{B}} \pi_{i+1}(b)\widetilde{\pi}(\cdot|b)$

It has been proved by Corollary 25 of (Jin et al., 2022b) that Algorithm 2 has the following convergence rate.

Lemma C.1. *Implement Algorithm 2 for iterations $i = 1, \ldots, t$ with hyperparameter choices $w_t = \frac{\alpha_t}{\prod_{i=2}^{t}(1-\alpha_i)}$ (α_i is defined in eq. (21)), $\gamma_t = \eta_t = \sqrt{(H \ln B)/t}$. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$, we have*

$$\min_{\phi} \sum_{i=1}^{t} \alpha_t^i \big[\langle \pi_i(\cdot), l_i(\cdot)\rangle - \langle (\phi \circ \pi_i)(\cdot), l_i(\cdot)\rangle\big] \le 10B\sqrt{\frac{H\ln(B^2/\delta)}{t}},$$

where $\phi \circ \pi_i$ can be defined by reducing the definition of stochastic modification in Definition 2.2 to single-agent policy π_i.

Implementing output policy $\hat{\pi}$: After obtaining all $\pi_{k,h}$ by calling Algorithm 2 in Algorithm 1, we define the final output policy $\hat{\pi}$ as Algorithm 4 below.

To facilitate the technical proof in Lemma F.4 and Lemma F.6, we also define policy $\hat{\pi}_{k,h}$ for all $k \in [K]$ and $h \in [H]$ in Algorithm 4, which can be seen as part of Algorithm 3.

D PROOF OF THEOREM 4.4

Definition D.1. Let $s_{k,h}$ be the state visited at h-th step k-th episode. The density of $s_{k,h}$ is universally bounded below by $p_{\min}$; that is

$$p_{\min} = \inf_{s\in\mathcal{S}, k\in\mathbb{N}, h\in[H]} \mathbb{P}(s_{k,h} = s).$$

Algorithm 3: Implement output policy $\hat{\pi}$.

Input: States $s_{1:H}$, policies $\{\pi_{k,h}\}_{k\in[K],h\in[H]}$ obtained from Algorithm 1.
Sample $k \in [K]$ uniformly at random.
for step $h = 1, \ldots, H$ **do**
  Let $t \leftarrow N_{k,h}(s_h)$ and $\{k_h^i(s_h)\}_{i=1}^t$ ($k_h^1(s_h) < k_h^2(s_h) < \ldots < k_h^t(s_h) < k$) be the episodes where state s_h is visited at the h-th step, i.e., $s_{k_h^i(s_h),h} = s_h$.
  Randomly select $i \in [t]$ with probability α_t^i and set $k \leftarrow k_h^i(s_h)$.
  Generate $a_h \sim \pi_{k,h}(\cdot|s_h)$.
end
Output: Joint actions $a_{1:H}$.

Algorithm 4: Implement policy $\hat{\pi}_{k,h}$.

Input: Time step h, episode k, states $s_{h:H}$, policies $\{\pi_{k',h:H}\}_{k'=1}^k$ obtained from Algorithm 1.
for step $h' = h, h+1, \ldots, H$ **do**
  Set $t \leftarrow N_{k,h'}(s_{h'})$ and let $\{k_{h'}^i(s_{h'})\}_{i=1}^t$ ($k_{h'}^1(s_{h'}) < k_{h'}^2(s_{h'}) < \ldots < k_{h'}^t(s_{h'}) < k$) be the episodes where state $s_{h'}$ is visited at h'-th step, i.e., $s_{k_{h'}^i(s_{h'}),h} = s_{h'}$.
  Randomly select $i \in [t]$ with probability α_t^i and set $k \leftarrow k_{h'}^i(s_{h'})$.
  $a_{h'} \sim \pi_{k,h'}(\cdot|s_{h'})$.
end
Output: Joint actions $a_{h:H}$.

Theorem 4.4. *Let $S := |\mathcal{S}|$ and $A := \max_{1\le j\le m} |\mathcal{A}^{(j)}|$ correspond to the size of the state space and action space, respectively. Choose $\beta_t^{(j)}$, α_t and α_t^i according to eqs. (19)-(21). Let the diameter of uncertainty set D satisfy $D \le \max\{\frac{p_{\min}}{H}, \frac{\epsilon}{SH^2}\}$. The output policy $\hat{\pi}$ produced by Algorithm 1 satisfies the following convergence rate with probability at least $1 - c\delta$ for some constant $c > 0$.*

1. *If $p_{\min} > \frac{\epsilon}{SH}$, then*

$$\max_{j\in[J]} \max_{s\in\mathcal{S}} \left(\boldsymbol{V}^{(j)}_{\phi^*\circ\hat{\pi},1}(s) - \boldsymbol{V}^{(j)}_{\hat{\pi},1}(s)\right) \le \mathcal{O}\Big(\frac{H}{p_{\min} - DH}\Big(A\sqrt{\frac{H^3S}{K}\ln\frac{mKHSA^2}{\delta}} + \mathfrak{e}\Big)\Big).$$

   To achieve an ϵ gap, we set $\mathfrak{e} = \mathcal{O}(\frac{\epsilon p_{\min}}{H})$ and require $K = \widetilde{\mathcal{O}}(SA^2H^5p_{\min}^{-2}\epsilon^{-2})$ episodes.

2. *If $p_{\min} \le \frac{\epsilon}{SH}$, then*

$$\max_{j\in[J]} \max_{s\in\mathcal{S}} \left(\boldsymbol{V}^{(j)}_{\phi^*\circ\hat{\pi},1}(s) - \boldsymbol{V}^{(j)}_{\hat{\pi},1}(s)\right) \le 5DSH^2 + \mathcal{O}\Big(H\Big(A\sqrt{\frac{H^3S}{K}\ln\frac{mKHSA^2}{\delta}} + \mathfrak{e}\Big)\Big).$$

   To achieve an ϵ gap, we set $\mathfrak{e} = \mathcal{O}(\frac{\epsilon}{H})$ and require $K = \widetilde{\mathcal{O}}(SA^2H^5\epsilon^{-2})$ episodes.

Proof. Note that $V_{k,h}^{(j)}(s)$ (defined by eqs. (5) & (6)) and $\underline{V}_{k,h}^{(j)}(s)$ (defined by eqs. (25) & (26)) are respectively the upper bound and the lower bound of $\boldsymbol{V}^{(j)}_{\hat{\pi}_{k,h},h}(s)$ (Since $V_{k,h}^{(j)}(s) \ge \boldsymbol{V}^{(j)}_{\phi^*\circ\hat{\pi}_{k,h},h}(s) \ge \boldsymbol{V}^{(j)}_{\hat{\pi}_{k,h},h}(s) \ge \underline{V}_{k,h}^{(j)}(s)$ based on Lemmas F.4 & F.6). Denote the gap between the upper bound and the lower bound as follows

$$\delta_{k,h}^{(j)} := V_{k,h}^{(j)}(s_{k,h}) - \underline{V}_{k,h}^{(j)}(s_{k,h}) \ge 0.$$

Let $s_{k,h}$, $a_{k,h}$ respectively be the state and action at the h-th step in the k-th episode. Let $\{k_{k,h}^i\}_{1\le i\le n_{k,h}}$ ($k_{k,h}^1 < k_{k,h}^2 < \ldots < k_{k,h}^{n_{k,h}} < k$) be the set of episodes in which the state $s_{k,h}$ is visited at the h-th step. $n_{k,h} := N_{k,h}(s_{k,h})$ is the number of such visits.

Then we unroll the update rule for both $V_{k,h}^{(j)}(s_{k,h})$ and $\underline{V}_{k,h}^{(j)}(s_{k,h})$ along k as follows.

$$\delta_{k,h}^{(j)} \overset{(i)}{\le} \widetilde{V}_{k,h}^{(j)}(s_{k,h}) - \underline{V}_{k,h}^{(j)}(s_{k,h})$$

$$
\begin{aligned}
&\overset{(ii)}{=} \alpha^0_{n_{k,h}}(H-h+1) + \sum_{i=1}^{n_{k,h}} \alpha^i_{n_{k,h}} \left(\widehat{\sigma}_{\mathcal{P}_h(s_{k,h},a_{k^i_{k,h},h})}(V^{(j)}_{k^i_{k,h},h+1}) - \widehat{\sigma}_{\mathcal{P}_h(s_{k,h},a_{k^i_{k,h},h})}(\underline{V}^{(j)}_{k^i_{k,h},h+1}) + 2\beta^{(j)}_i \right) \\
&\overset{(iii)}{\leq} \sum_{i=1}^{n_{k,h}} \alpha^i_{n_{k,h}} \left(\sigma_{\mathcal{P}_h(s_{k,h},a_{k^i_{k,h},h})}(V^{(j)}_{k^i_{k,h},h+1}) - \sigma_{\mathcal{P}_h(s_{k,h},a_{k^i_{k,h},h})}(\underline{V}^{(j)}_{k^i_{k,h},h+1}) \right) + 2\sum_{i=1}^{n_{k,h}} \alpha^i_{n_{k,h}} \beta^{(j)}_i + 2\mathfrak{e} \sum_{i=1}^{n_{k,h}} \alpha^i_{n_{k,h}} \\
&\overset{(iv)}{\leq} \sum_{i=1}^{n_{k,h}} \alpha^i_{n_{k,h}} \left(\sigma_{\mathcal{P}_h(s_{k,h},a_{k^i_{k,h},h})}(V^{(j)}_{k^i_{k,h},h+1}) - \sigma_{\mathcal{P}_h(s_{k,h},a_{k^i_{k,h},h})}(\underline{V}^{(j)}_{k^i_{k,h},h+1}) \right) + \Theta\Big(A_j \sqrt{\frac{H^3}{n_{k,h}} \ln \frac{mKHSA_j^2}{\delta}} \Big) + 4\mathfrak{e},
\end{aligned}
$$

where (i) uses eqs. (6) & (26), (ii) unrolls the update rules (5) & (25), (iii) uses eq. (27) and $\alpha^0_t = 0$ (eq. (21)), and (iv) uses eqs. (22) & (24). By summing over k, we obtain the following recursion:

$$
\begin{aligned}
\sum_{k=1}^{K} \delta^{(j)}_{k,h} \leq & \sum_{k=1}^{K} \sum_{i=1}^{n_{k,h}} \alpha^i_{n_{k,h}} \left(\sigma_{\mathcal{P}_h(s_{k,h},a_{k^i_{k,h},h})}(V^{(j)}_{k^i_{k,h},h+1}) - \sigma_{\mathcal{P}_h(s_{k,h},a_{k^i_{k,h},h})}(\underline{V}^{(j)}_{k^i_{k,h},h+1}) \right) \\
& + \sum_{k=1}^{K} \Theta\Big(A_j \sqrt{\frac{H^3}{n_{k,h}} \ln \frac{mKHSA_j^2}{\delta}} \Big) + 4K\mathfrak{e} \\
\overset{(i)}{\leq} & \sum_{k'=1}^{K} \left(\sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(V^{(j)}_{k',h+1}) - \sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(\underline{V}^{(j)}_{k',h+1}) \right) \sum_{i=n^{k'}_h+1}^{\infty} \alpha^{n^{k'}_h}_i \\
& + \sum_{k=1}^{K} \Theta\Big(A_j \sqrt{\frac{H^3}{n_{k,h}} \ln \frac{mKHSA_j^2}{\delta}} \Big) + 4K\mathfrak{e} \\
\overset{(ii)}{\leq} & \Big(1 + \frac{1}{H}\Big) \sum_{k'=1}^{K} \left(\sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(V^{(j)}_{k',h+1}) - \sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(\underline{V}^{(j)}_{k',h+1}) \right) \\
& + \sum_{s} \sum_{n=1}^{N_{K+1,h}(s)} \Theta\Big(A_j \sqrt{\frac{H^3}{n} \ln \frac{mKHSA_j^2}{\delta}} \Big) + 4K\mathfrak{e} \\
\overset{(iii)}{\leq} & \Big(1 + \frac{1}{H}\Big) \sum_{k'=1}^{K} \left(\sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(V^{(j)}_{k',h+1}) - \sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(\underline{V}^{(j)}_{k',h+1}) \right) \\
& + S \cdot \frac{1}{S} \sum_{s} \Theta\Big(A_j \sqrt{H^3 N_{K+1,h}(s) \ln \frac{mKHSA_j^2}{\delta}} \Big) + 4K\mathfrak{e} \\
\overset{(iv)}{=} & \Big(1 + \frac{1}{H}\Big) \sum_{k'=1}^{K} \left(\sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(V^{(j)}_{k',h+1}) - \sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(\underline{V}^{(j)}_{k',h+1}) \right) \\
& - \Big(1 + \frac{1}{H}\Big) \sum_{k'=1}^{K} \left(V^{(j)}_{k',h+1}(s_{k',h+1}) - \underline{V}^{(j)}_{k',h+1}(s_{k',h+1}) \right) \\
& + \Big(1 + \frac{1}{H}\Big) \sum_{k'=1}^{K} \left(V^{(j)}_{k',h+1}(s_{k',h+1}) - \underline{V}^{(j)}_{k',h+1}(s_{k',h+1}) \right) \\
& + S\Theta\Big(A_j \sqrt{H^3 \frac{1}{S} \sum_{s} N_{K+1,h}(s) \ln \frac{mKHSA_j^2}{\delta}} \Big) + 4K\mathfrak{e} \\
\overset{(v)}{=} & \Big(1 + \frac{1}{H}\Big) \sum_{k'=1}^{K} \left(\sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(V^{(j)}_{k',h+1}) - \sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(\underline{V}^{(j)}_{k',h+1}) \right) \\
& - \Big(1 + \frac{1}{H}\Big) \sum_{k'=1}^{K} \left(V^{(j)}_{k',h+1}(s_{k',h+1}) - \underline{V}^{(j)}_{k',h+1}(s_{k',h+1}) \right)
\end{aligned} \tag{12}
$$

$$+\Big(1+\frac{1}{H}\Big)\sum_{k'=1}^{K}\delta_{k,h+1}^{(j)}+\Theta\Big(A_j\sqrt{H^3SK\ln\frac{mKHSA_j^2}{\delta}}\Big)+4K\mathfrak{e}$$

where (i) changes the order of summation, (ii) uses eq. (23) and pigeonhole argument, (iii) uses $\sum_{n=1}^{N_{K+1,h}(s)}\sqrt{1/n}=\Theta\big(N_{K+1,h}(s)\big)$, (iv) applies Jensen's inequality to the convex function $\sqrt{\cdot}$, and (v) is by the definition of $\delta_{k,h+1}$. Now we apply Lemma F.7 to the first two terms above on the right-hand side,

$$\sum_{k'=1}^{K}\Big(\sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(V_{k',h+1}^{(j)})-\sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(\underline{V}_{k',h+1}^{(j)})\Big)$$
$$-\sum_{k'=1}^{K}\Big(V_{k',h+1}^{(j)}(s_{k',h+1})-\underline{V}_{k',h+1}^{(j)}(s_{k',h+1})\Big). \tag{13}$$

Then we obtain the recursion

$$\sum_{k=1}^{K}\delta_{k,h}^{(j)}\le D\Big(1+\frac{1}{H}\Big)\sum_{k=1}^{K}\sum_{s\in\mathcal{S}}\Big(V_{k,h+1}^{(j)}(s)-\underline{V}_{k,h+1}^{(j)}(s)\Big)+\Big(1+\frac{1}{H}\Big)\sqrt{32KH^2\ln\frac{2mH}{\delta}}$$
$$+\Big(1+\frac{1}{H}\Big)\sum_{k=1}^{K}\delta_{k,h+1}^{(j)}+\Theta\Big(A_j\sqrt{H^3SK\ln\frac{mKHSA_j^2}{\delta}}\Big)+4K\mathfrak{e}. \tag{14}$$

We apply Lemma F.8 to solve this recursive relation by setting

$$\mathsf{a}_h=\sum_{k=1}^{K}\delta_{k,h}^{(j)},\mathsf{b}_h=\sum_{k=1}^{K}\sum_{s\in\mathcal{S}}\Big(V_{k,h}^{(j)}(s)-\underline{V}_{k,h}^{(j)}(s)\Big),\mathsf{C}_1=1+\frac{1}{H},\mathsf{C}_2=D\Big(1+\frac{1}{H}\Big),$$
$$\mathsf{C}_3=\Big(1+\frac{1}{H}\Big)\sqrt{32KH^2\ln\frac{2mH}{\delta}}+\Theta\Big(A_j\sqrt{H^3SK\ln\frac{mKHSA_j^2}{\delta}}\Big)+4K\mathfrak{e}.$$

Then we obtain

$$\sum_{k=1}^{K}\delta_{k,h}^{(j)}\le D\Big(1+\frac{1}{H}\Big)\sum_{i=0}^{H-h}\Big(1+\frac{1}{H}\Big)^i\sum_{k=1}^{K}\sum_{s\in\mathcal{S}}\Big(V_{k,h+1+i}^{(j)}(s)-\underline{V}_{k,h+1+i}^{(j)}(s)\Big) \tag{15}$$
$$+3H\Big[\Big(1+\frac{1}{H}\Big)\sqrt{32KH^2\ln\frac{2mH}{\delta}}+\Theta\Big(A_j\sqrt{H^3SK\ln\frac{mKHSA_j^2}{\delta}}\Big)+4K\mathfrak{e}\Big],$$

where we also apply $(1+\frac{1}{H})^{H-h+1}<3$. For convenience, we define

$$\Delta_{k,h}^{(j)}:=\sum_{s\in\mathcal{S}}\Big(V_{k,h}^{(j)}(s)-\underline{V}_{k,h}^{(j)}(s)\Big)$$

and

$$\mathscr{U}_j:=3H\Big[\Big(1+\frac{1}{H}\Big)\sqrt{32KH^2\ln\frac{2mH}{\delta}}+\Theta\Big(A_j\sqrt{H^3SK\ln\frac{mKHSA_j^2}{\delta}}\Big)+4K\mathfrak{e}\Big].$$

Then we can have the following compact form of eq. 15:

$$\sum_{k=1}^{K}\delta_{k,h}^{(j)}\le D\sum_{i=0}^{H-h}(1+\frac{1}{H})^{i+1}\sum_{k=1}^{K}\Delta_{k,h+1+i}^{(j)}+\mathscr{U}_j.$$

Starting here, we let $D\le\max\{\frac{p_{\min}}{H},\frac{\epsilon}{SH^2}\}$. For, the case where $D\le\frac{p_{\min}}{H}$, we take expectation on both sides and obtain

$$\sum_{k=1}^{K}\sum_{s\in\mathcal{S}}\mathbb{P}(s_{k,h}=s)\Big(V_{k,h}^{(j)}(s)-\underline{V}_{k,h}^{(j)}(s)\Big)\le D\sum_{i=0}^{H-h}(1+\frac{1}{H})^{i+1}\sum_{k=1}^{K}\Delta_{k,h+1+i}^{(j)}+\mathscr{U}_j.$$

The left-hand side can be further lower bounded using Definition D.1 when $p_{\min} > \frac{\epsilon}{SH}$. We have

$$p_{\min} \sum_{k=1}^{K} \Delta_{k,h}^{(j)} \le D \sum_{i=0}^{H-h} (1 + \frac{1}{H})^{i+1} \sum_{k=1}^{K} \Delta_{k,h+1+i}^{(j)} + \mathscr{U}_j. \tag{16}$$

To solve this recursion, we apply Lemma F.9 by setting

$$\mathsf{a}_h = \sum_{k=1}^{K} \Delta_{k,h}^{(j)}, \mathsf{C}_1 = \frac{\mathscr{U}_j}{p_{\min}}, \text{ and } \mathsf{C}_2 = \frac{D}{p_{\min}}.$$

Let $\frac{D}{p_{\min}} < \frac{1}{H}$. Then we obtain the following upper bound:

$$\max_{h \in [H]} \sum_{k=1}^{K} \Delta_{k,h}^{(j)} \le \frac{\mathscr{U}_j}{p_{\min} - HD}. \tag{17}$$

Lastly, we bound the optimality gap at the step h. In expectation with respect choosing k, we have

$$\begin{aligned}
&\max_{j \in [m]} \max_{s \in \mathcal{S}} [\boldsymbol{V}_{\phi^* \circ \hat{\pi}, 1}^{(j)}(s) - \boldsymbol{V}_{\hat{\pi}, 1}^{(j)}(s)] \\
\le &\max_{j \in [m]} \sum_{s \in \mathcal{S}} [\boldsymbol{V}_{\phi^* \circ \hat{\pi}, 1}^{(j)}(s) - \boldsymbol{V}_{\hat{\pi}, 1}^{(j)}(s)] \\
\overset{(i)}{\le} &\max_{j \in [m]} \frac{1}{K} \sum_{k=1}^{K} \sum_{s \in \mathcal{S}} [\boldsymbol{V}_{\phi^* \circ \hat{\pi}_{k,1}, 1}^{(j)}(s) - \boldsymbol{V}_{\hat{\pi}_{k,1}, 1}^{(j)}(s)] \\
\overset{(ii)}{\le} &\max_{j \in [m]} \frac{1}{K} \sum_{k=1}^{K} \sum_{s \in \mathcal{S}} \Big(V_{k,1}^{(j)}(s) - \underline{V}_{k,1}^{(j)}(s) \Big) \\
\overset{(iii)}{=} &\max_{j \in [m]} \frac{1}{K} \sum_{k=1}^{K} \Delta_{k,1}^{(j)} \\
\overset{(iv)}{\le} &\frac{3H}{p_{\min} - HD} \Big[2\sqrt{\frac{32H^2}{K} \ln \frac{2mKHSA}{\delta}} + \Theta\Big(A \sqrt{\frac{H^3 S}{K} \ln \frac{mKHSA^2}{\delta}} \Big) + 4\mathfrak{e} \Big]
\end{aligned}$$

where (i) uses the definitions of $\hat{\pi}$ and $\hat{\pi}_{k,h}$ given by Algorithms 3 & 4 respectively, (ii) uses Lemmas F.4 & F.6 and sampling rule of k given in Algorithm 3, (iii) is by the definition of $\Delta_{k,h}^{(j)}$, and (iv) uses eq. (17) and $A := \max_{1 \le j \le m} A_j$. It completes the proof.

When evaluating the sample complexity, we set $\mathfrak{e} = \frac{p_{\min} - HD}{24H} \epsilon$. Then the last term is bounded by $\epsilon/2$. Then we let $\frac{1}{p_{\min} - HD} \sqrt{H^5 S A^2 \ln \frac{mKHSA^2}{\delta} / K} \le \frac{1}{2} \epsilon$. It solves the number of episodes for achieving ϵ-approximation of robust correlated equilibrium is $K = \widetilde{\mathcal{O}}(SA^2 H^5 p_{\min}^{-2} \epsilon^{-2})$.

For the second case where $D \le \frac{\epsilon}{SH^2}$, we recall that the first term of the recursion eq. (15),

$$D\Big(1 + \frac{1}{H}\Big) \sum_{i=0}^{H-h} \Big(1 + \frac{1}{H}\Big)^i \sum_{k=1}^{K} \sum_{s \in \mathcal{S}} \Big(V_{k,h+1+i}^{(j)}(s) - \underline{V}_{k,h+1+i}^{(j)}(s) \Big)$$

measures the influence of diameter of uncertainty set on the convergence error of robust V-learning. We can estimate this term with a universal upper bound; that is, for $h = 1$,

$$\begin{aligned}
&D\Big(1 + \frac{1}{H}\Big) \sum_{i=1}^{H} \Big(1 + \frac{1}{H}\Big)^{i-1} \sum_{k=1}^{K} \sum_{s \in \mathcal{S}} \Big(V_{k,1+i}^{(j)}(s) - \underline{V}_{k,1+i}^{(j)}(s) \Big) \\
\le &D\Big(1 + \frac{1}{H}\Big) \sum_{i=0}^{H-1} \Big(1 + \frac{1}{H}\Big)^{i} \sum_{k=1}^{K} \sum_{s \in \mathcal{S}} \Big(H - i \Big) \\
= &DSK \sum_{i=0}^{H-1} \Big(1 + \frac{1}{H}\Big)^{i} (H - i)
\end{aligned}$$

$$\leq 5DSKH^2.$$

Then eq. (15) can be bounded by

$$\sum_{k=1}^{K} \delta_{k,1}^{(j)} \leq 5DSKH^2 + 3H\Big[\Big(1+\frac{1}{H}\Big)\sqrt{32KH^2 \ln\frac{2mH}{\delta}} + \Theta\Big(A_j\sqrt{H^3SK\ln\frac{mKHSA_j^2}{\delta}}\Big) + 4K\mathfrak{e}\Big]. \tag{18}$$

The derived upper bound of optimality gap based on this inequality becomes

$$\max_{j\in[m]}[\boldsymbol{V}^{(j)}_{\phi^*\circ\hat{\pi},1}(s_1) - \boldsymbol{V}^{(j)}_{\hat{\pi},1}(s_1)]$$
$$\leq 5DSH^2 + 3H\Big[2\sqrt{\frac{32H^2}{K}\ln\frac{2mKHSA}{\delta}} + \Theta\Big(A\sqrt{\frac{H^3S}{K}\ln\frac{mKHSA^2}{\delta}}\Big) + 4\mathfrak{e}\Big],$$

where s_1 is the initial state. Since the initial state can be any state over $\mathcal{S}$ due to the initialization, we obtain the desired bound. If $D \leq \frac{\epsilon}{SH^2}$, then this bound implies the same sample complexity; the number of episodes for achieving ϵ-approximation of robust correlated equilibrium is $K = \widetilde{\mathcal{O}}(SA^2H^5\epsilon^{-2})$.

□

E Proof of Corollary 4.5

Corollary E.1. *In the case of a single player, the output policy $\hat{\pi}$ produced by Algorithm 1 achieves an approximate optimal robust value function at the same convergence rate as that in Theorem 4.4.*

Proof. It suffices to prove that for the single-agent case, all stochastic modifications of a policy form the space of all policies. Let Π be all distributions over $\mathcal{A}$ and $\pi \in \Pi$ is given with $\pi(a) > 0$ for all $a \in \mathcal{A}$. We will prove that

$$\{\phi \circ \pi : \phi \text{ is a stochastic modification}\} = \Pi.$$

For any $\mu \in \Pi$, we can construct the desired ϕ as follows: define $\phi(\cdot|b) = \mu$ for all $b \in \mathcal{A}$. Then we will show that $\phi \circ \pi$ is same μ.

$$\begin{aligned}\phi \circ \pi(a) &= \sum_{b\in\mathcal{A}} \pi(b)\phi(a|b) \\ &= \sum_{b\in\mathcal{A}} \pi(b)\mu(a) = \mu(a).\end{aligned}$$

This implies that $\Pi = \{\phi \circ \pi : \phi \text{ is a stochastic modification}\}$. It concludes that enumerating all stochastic modifications of a given policy is equivalent to enumerating all policies. □

F Supporting Lemmas

Hyperparameter choices Throughout this subsection, we use the following hyperparameter choices

$$\beta_t^{(j)} := cA_j\sqrt{\frac{H^3}{t}\ln\frac{mKHSA_j^2}{\delta}} + \mathfrak{e}, \tag{19}$$

$$\alpha_t := \frac{H+1}{H+t}, \tag{20}$$

$$\alpha_t^0 = 0, \quad \alpha_t^i = \alpha_i \prod_{j=i+1}^{t}(1-\alpha_j)(i \geq 1) \tag{21}$$

where $c > 0$ is an absolute constant. It can be seen that the above hyperparameters satisfy the following conditions. (Eq. (22) is obvious and eqs. (23) & (24) are proved in Lemma 10 of Jin et al. (2022a).)

$$\sum_{i=1}^{t} \alpha_t^i = 1 \tag{22}$$

$$\sum_{t=i}^{\infty} \alpha_t^i = 1 + \frac{1}{H} \tag{23}$$

$$\sum_{i=1}^{t} \alpha_t^i \beta_i^{(j)} = \Theta\left(A_j \sqrt{\frac{H^3}{t} \ln \frac{mKHSA_j^2}{\delta}}\right) + \mathfrak{e} \tag{24}$$

Pessimistic estimation of V function: To facilitate the proof, we also provide a pessimistic estimator of V functions denoted as $\underline{V}_{k,h}^{(j)}$, which is constructed by the following update rules with initial values $\underline{V}_{1,h}^{(j)}(s) = \underline{V}_{k,H+1}^{(j)}(s) = 0$ for all s, h, k, j

$$\underline{V}_{k+1,h}^{(j)}(s_h) \leftarrow (1-\alpha_t)\underline{V}_{k,h}^{(j)}(s_h) + \alpha_t \left(r_h^{(j)} + \widehat{\sigma}_{\mathcal{P}_h(s_h,a_h)}(\underline{V}_{k,h+1}^{(j)}) - \beta_t^{(j)}\right); \tag{25}$$

$$\underline{V}_{k+1,h}^{(j)}(s_h) \leftarrow \max\{0, \underline{V}_{k+1,h}^{(j)}(s_h)\}. \tag{26}$$

The above update rules are similar to those for optimistic estimation in eqs. (5) & (6), with the major difference that $+\beta_t^{(j)} > 0$ in eq. (5) yields optimism (i.e. $V_{k,h}^{(j)}(s) \geq \boldsymbol{V}_{\phi^* \circ \hat{\pi}_{k,h},h}^{(j)}(s) \geq \boldsymbol{V}_{\hat{\pi}_{k,h},h}^{(j)}(s)$ as shown by Lemma F.6) while $-\beta_t^{(j)} < 0$ in eq. (25) yields pessimism (i.e. $\underline{V}_{k,h}^{(j)}(s) \leq \boldsymbol{V}_{\hat{\pi}_{k,h},h}^{(j)}(s)$ as shown by Lemma F.4)

Lemma F.1. *The operator $\sigma_{\mathcal{P}_h(s,a)}$ defined in eq.* (3) *has the following properties:*

1. *Boundedness:* $\inf_{s'} V(s') \leq \sigma_{\mathcal{P}_h(s,a)}(V) \leq \sup_{s'} V(s')$.
2. *Monotonicity: $\sigma_{\mathcal{P}_h(s,a)}(V') \leq \sigma_{\mathcal{P}_h(s,a)}(V)$ for any V-tables V, V' such that $V'(s) \leq V(s), \forall s$.*
3. *Estimation bound: The estimator $\widehat{\sigma}_{\mathcal{P}_h(s,a)}$ has the following bounds for any V function V.*

$$\sigma_{\mathcal{P}_h(s,a)}(V) - \mathfrak{e} \leq \widehat{\sigma}_{\mathcal{P}_h(s,a)}(V) \leq \sigma_{\mathcal{P}_h(s,a)}(V) + \mathfrak{e}. \tag{27}$$

Proof. **Proof of boundedness:** The upper bound $\sigma_{\mathcal{P}_h(s,a)}(V) \leq \sup_{s'} V(s')$ can be directly proved based on eq. (3) as follows.

$$\begin{aligned}
\sigma_{\mathcal{P}_h(s,a)}(V) &= \inf_{\widetilde{\mathbb{P}}_h(\cdot|s,a) \in \mathcal{P}_h(s,a)} \sum_{s' \in \mathcal{S}} \widetilde{\mathbb{P}}_h(s'|s,a) V(s') \\
&\leq \inf_{\widetilde{\mathbb{P}}_h(\cdot|s,a) \in \mathcal{P}_h(s,a)} \sum_{s' \in \mathcal{S}} \widetilde{\mathbb{P}}_h(s'|s,a) \sup_{s'' \in \mathcal{S}} V(s'') \\
&\overset{(i)}{=} \sup_{s'' \in \mathcal{S}} V(s''),
\end{aligned}$$

where (i) uses $\sum_{s' \in \mathcal{S}} \widetilde{\mathbb{P}}_h(s'|s,a) = 1$. The proof logic for the lower bound $\inf_{s'} V(s') \leq \sigma_{\mathcal{P}_h(s,a)}(V)$ is similar.

Proof of monotonicity: Suppose $p \in \mathcal{P}_h(s,a)$ achieves the infimum in $\sigma_{\mathcal{P}_h(s,a)}(V)$ defined by eq. (3), i.e.

$$\sigma_{\mathcal{P}_h(s,a)}(V) = \sum_{s' \in \mathcal{S}} p(s') V(s'). \tag{28}$$

Then the monotonicity can be proved as follows.

$$\sigma_{\mathcal{P}_h(s,a)}(V) - \sigma_{\mathcal{P}_h(s,a)}(V')$$

$$= \sum_{s' \in \mathcal{S}} p(s')V(s') - \inf_{\widetilde{\mathbb{P}}_h(\cdot|s,a) \in \mathcal{P}_h(s,a)} \sum_{s' \in \mathcal{S}} \widetilde{\mathbb{P}}_h(s'|s,a)V'(s')$$
$$\overset{(i)}{\geq} \sum_{s' \in \mathcal{S}} p(s')V(s') - \sum_{s' \in \mathcal{S}} p(s')V'(s') \overset{(ii)}{\geq} 0, \tag{29}$$

where (i) will be used later and (ii) uses $p(s') \geq 0$ and $V(s') \geq V'(s')$ for all $s' \in \mathcal{S}$.

Proof of estimation bound: $\mathfrak{e} := \sup_{h,s,a,V} \left|\sigma_{\mathcal{P}_h(s,a)}(V) - \widehat{\sigma}_{\mathcal{P}_h(s,a)}(V)\right|$ defined by Definition 4.3 directly implies eq. (27). □

Lemma F.2. *For any player j and all $s \in \mathcal{S}$, the V-table $\widetilde{V}_{k,h}^{(j)}$ and $\underset{\sim}{V}_{k,h}^{(j)}$ tracked by Algorithm 1 at the h-th step in the k-th episode satisfying $\widetilde{V}_{k,h}^{(j)}(s) \geq 0$ and $\underset{\sim}{V}_{k,h}^{(j)}(s) \leq H + 1 - h$.*

Proof. We will only prove $\underset{\sim}{V}_{k,h}^{(j)}(s) \leq H + 1 - h$ since the proof logic for $\widetilde{V}_{k,h}^{(j)}(s) \geq 0$ is similar. For $k = 1$, the initial value $\underset{\sim}{V}_{1,h}^{(j)}(s) := 0 \leq H + 1 - h$. Then we assume $\underset{\sim}{V}_{k,h}^{(j)}(s) \leq H + 1 - h$ for a certain fixed $k \geq 1$ and we prove $\underset{\sim}{V}_{k+1,h}^{(j)}(s) \leq H + 1 - h$ as follows.

$$\begin{aligned}
\underset{\sim}{V}_{k+1,h}^{(j)}(s_h) &\overset{(i)}{=} (1-\alpha_t)\underset{\sim}{V}_{k,h}^{(j)}(s_h) + \alpha_t\big(r_h^{(j)}(s_h,a_h) + \widehat{\sigma}_{\mathcal{P}_h(s_h,a_h)}(\underline{V}_{k,h+1}^{(j)}) - \beta_t^{(j)}\big) \\
&\overset{(ii)}{\leq} (1-\alpha_t)(H+1-h) + \alpha_t\big(1 + \sigma_{\mathcal{P}_h(s_h,a_h)}(\underline{V}_{k,h+1}^{(j)}) - \beta_t^{(j)} + \mathfrak{e}\big) \\
&\overset{(iii)}{\leq} H + 1 - h,
\end{aligned}$$

where (i) uses the update rules (25) & (26), (ii) uses $\underset{\sim}{V}_{k,h}^{(j)}(s) \leq H + 1 - h$ and eq. (27), and (iii) uses $\beta_t^{(j)} \geq \mathfrak{e}$ based on eq. (19) and the following inequality based on item 1 of Lemma F.1. This concludes the proof.

$$\sigma_{\mathcal{P}_h(s_h,a_h)}(\underline{V}_{k,h+1}^{(j)}) \leq \max_s \underline{V}_{k,h+1}^{(j)}(s) \leq \max_s \left[\max\left(0, \underset{\sim}{V}_{k,h+1}^{(j)}(s)\right)\right] \leq H - h.$$

□

The following lemma says that the tracked upper confidence bound $V_{k,h}^{(j)}(s)$ is always larger than the lower confidence bound $\underline{V}_{k,h}^{(j)}(s)$.

Lemma F.3. *For any player j and all $s \in \mathcal{S}$, the V-tables $V_{k,h}^{(j)}(s)$ and $\underline{V}_{k,h}^{(j)}(s)$ tracked by Algorithm 1 at the h-th step in the k-th episode satisfy the following inequality*

$$V_{k,h}^{(j)}(s) \geq \underline{V}_{k,h}^{(j)}(s). \tag{30}$$

Proof. It suffices to show $\widetilde{V}_{k,h}^{(j)}(s) \geq \underset{\sim}{V}_{k,h}^{(j)}(s)$ since it implies eq. (30) as follows

$$\begin{aligned}
&V_{k,h}^{(j)}(s) - \underline{V}_{k,h}^{(j)}(s) \\
&\overset{(i)}{=} \min\{H + 1 - h, \widetilde{V}_{k,h}^{(j)}(s_h)\} - \max\{0, \underset{\sim}{V}_{k,h}^{(j)}(s_h)\} \\
&\overset{(ii)}{=} \min\{H + 1 - h, \max[0, \widetilde{V}_{k,h}^{(j)}(s_h)]\} - \min\{H + 1 - h, \max[0, \underset{\sim}{V}_{k,h}^{(j)}(s_h)]\} \\
&\overset{(iii)}{\geq} 0,
\end{aligned} \tag{31}$$

where (i) uses eqs. (6) and (26), (ii) uses Lemma F.2, and (iii) uses $\widetilde{V}_{k,h}^{(j)}(s) \geq \underset{\sim}{V}_{k,h}^{(j)}(s)$.

Then we prove $\widetilde{V}_{k,h}^{(j)}(s) \geq \underset{\sim}{V}_{k,h}^{(j)}(s)$ via induction with regards to k. For $k = 1$, $\widetilde{V}_{1,h}^{(j)}(s) = H+1-h \geq \underset{\sim}{V}_{1,h}^{(j)}(s) = 0$ due to initialization. Suppose $\widetilde{V}_{k,h}^{(j)}(s) \geq \underset{\sim}{V}_{k,h}^{(j)}(s)$ and thus eq. (30) holds for a certain fixed k. Then we aim to prove $\widetilde{V}_{k+1,h}^{(j)}(s) \geq \underset{\sim}{V}_{k+1,h}^{(j)}(s)$ (i.e., eq. (30) also holds for $k+1$). It suffices

to consider the case where s is the state visited at the h-th step in the k-th episode, that is, $s = s_{k,h}$. Otherwise, $\widetilde{V}^{(j)}_{k+1,h}(s) = \widetilde{V}^{(j)}_{k,h}(s) \geq \underline{V}^{(j)}_{k,h}(s) = \underline{V}^{(j)}_{k+1,h}(s)$. When $s = s_{k,h}$, the update rules (5) & (25) imply that

$$\begin{aligned}
&\widetilde{V}^{(j)}_{k+1,h}(s) - \underline{V}^{(j)}_{k+1,h}(s) \\
=&(1-\alpha_t)\left(\widetilde{V}^{(j)}_{k,h}(s) - \underline{V}^{(j)}_{k,h}(s)\right) + \alpha_t\left(\widehat{\sigma}_{\mathcal{P}_h(s,a)}(V^{(j)}_{k,h+1}) - \widehat{\sigma}_{\mathcal{P}_h(s,a)}(\underline{V}^{(j)}_{k,h+1})\right) + 2\alpha_t\beta^{(j)}_t \\
\overset{(i)}{\geq}&\alpha_t\left(\sigma_{\mathcal{P}_h(s,a)}(V^{(j)}_{k,h+1}) - \sigma_{\mathcal{P}_h(s,a)}(\underline{V}^{(j)}_{k,h+1}) - 2\mathfrak{e} + 2\beta^{(j)}_t\right) \overset{(ii)}{\geq} 0
\end{aligned} \tag{32}$$

where (i) uses $\widetilde{V}^{(j)}_{k,h}(s) \geq \underline{V}^{(j)}_{k,h}(s)$ and eq. (27), and (ii) uses $V^{(j)}_{k,h+1} \geq \underline{V}^{(j)}_{k,h+1}$, the monotonicity of $\sigma_{\mathcal{P}_h(s,a)}$ (see item 2 of Lemma F.1) and $\beta^{(j)}_t \geq \mathfrak{e}$ (see eq. (19)). This concludes the proof.

□

Lemma F.4. *The V-tables $\boldsymbol{V}^{(j)}_{\hat{\pi}_{k,h},h}$ and $\underline{V}^{(j)}_{k,h}$ satisfy the following inequality with probability at least $1-\delta$ for all players $j \in [m]$, episodes $k \in [K]$, time steps $h \in [H]$ and states $s \in \mathcal{S}$ and any $\delta \in (0, \frac{1}{2})$,*

$$\boldsymbol{V}^{(j)}_{\hat{\pi}_{k,h},h}(s) \geq \underline{V}^{(j)}_{k,h}(s).$$

Proof. It suffices to prove $\boldsymbol{V}^{(j)}_{\hat{\pi}_{k,h},h}(s) \geq \underline{V}^{(j)}_{k,h}(s)$; it implies $\boldsymbol{V}^{(j)}_{\hat{\pi}_{k,h},h}(s) \geq \underline{V}^{(j)}_{k,h}(s)$ because $\boldsymbol{V}^{(j)}_{\hat{\pi}_{k,h},h}(s) \geq 0$ by its definition (2) (note that $r^{(j)}_\ell(s_\ell, a_\ell) \geq 0$) and $\underline{V}^{(j)}_{k,h}(s) := \max\{0, \underline{V}^{(j)}_{k,h}(s_h)\}$. Now we start to prove $\boldsymbol{V}^{(j)}_{\hat{\pi}_{k,h},h}(s) \geq \underline{V}^{(j)}_{k,h}(s)$ by induction with respect to h backward. When $h = H+1$, the proof is trivial as $\boldsymbol{V}^{(j)}_{\pi,H+1}(s) = \underline{V}^{(j)}_{0,H+1}(s) = 0$ for any policy π. Then suppose $\boldsymbol{V}^{(j)}_{\hat{\pi}_{k,h+1},h+1}(s) \geq \underline{V}^{(j)}_{k,h+1}(s)$ holds so $\boldsymbol{V}^{(j)}_{\hat{\pi}_{k,h+1},h+1}(s) \geq \underline{V}^{(j)}_{k,h+1}(s)$ for a certain fixed h, and we will prove that $\boldsymbol{V}^{(j)}_{\hat{\pi}_{k,h},h}(s) \geq \underline{V}^{(j)}_{k,h}(s)$. Let $\{k_i\}_{1\leq i\leq t}$ ($k_1 < k_2 < \ldots < k_t < k$) be the set of episodes where the state s is visited at the h-th step. Then we unroll the update rule (25) of $\underline{V}^{(j)}_{k,h}(s)$ as follows with respect to the episode k.

$$\begin{aligned}
\underline{V}^{(j)}_{k,h}(s) &= \sum_{i=1}^{t} \alpha^i_t \left[r^{(j)}_h(s, a_{k^i,h}) + \widehat{\sigma}_{\mathcal{P}_h(s,a_{k^i,h})}(\underline{V}^{(j)}_{k^i,h+1}) - \beta^{(j)}_i\right] \\
&\overset{(i)}{\leq} \sum_{i=1}^{t} \alpha^i_t \left[r^{(j)}_h(s, a_{k^i,h}) + \sigma_{\mathcal{P}_h(s,a_{k^i,h})}(\underline{V}^{(j)}_{k^i,h+1}) + \mathfrak{e} - \beta^{(j)}_i\right],
\end{aligned} \tag{33}$$

where (i) uses eq. (27). Let

$$X_i := \alpha^i_t \left[r^{(j)}_h(s, a_{k^i,h}) + \sigma_{\mathcal{P}_h(s_h,a_{k^i,h})}(\underline{V}^{(j)}_{k^i,h+1})\right].$$

Then X_i always has the following bound since $\underline{V}^{(j)}_{k^i,h+1} = \max\{0, \underline{V}^{(j)}_{k^i,h+1}(s_h)\} \leq H - h$ based on Lemma F.2.

$$0 \leq X_i \leq \alpha^i_t(H+1-h).$$

Then by using Azuma's inequality and applying union bound to all $j \in [m], i \in [t] \subset [K], h \in [H], s \in \mathcal{S}$, we have the following bound with probability at least $1-\delta$.

$$\sum_{i=1}^{t} X_i - \mathbb{E}\Big[\sum_{i=1}^{t} X_i\Big] \leq \sqrt{\frac{1}{2}\Big(\ln\frac{2mKHS}{\delta}\Big)\sum_{i=1}^{t}(\alpha^i_t)^2(H+1-h)^2} \overset{(i)}{\leq} \sqrt{\frac{H^3}{t}\ln\frac{2mKHS}{\delta}}, \tag{34}$$

where (i) uses $\sum_{i=1}^{t}(\alpha^i_t)^2 \leq \frac{2H}{t}$ in Lemma 10 of Jin et al. (2022a). Therefore, with probability at least $1-\delta$, the following inequality holds for all j, k, h, s.

$$\underline{V}^{(j)}_{k,h}(s) \overset{(i)}{\leq} \sum_{i=1}^{t} X_i$$

$$
\begin{aligned}
&\overset{(ii)}{\leq} \sum_{i=1}^{t} \alpha_t^i \mathbb{E}_{\pi_{k^i,h}} \left[r_h^{(j)}(s, a_{k^i,h}) + \sigma_{\mathcal{P}_h(s_h, a_{k^i,h})}(\underline{V}^{(j)}_{k^i,h+1}) \right] \\
&\quad + \sqrt{\frac{H^3}{t} \ln \frac{2mKHS}{\delta}} - \sum_{i=1}^{t} \alpha_t^i (\beta_i^{(j)} - \mathfrak{e}) \\
&\overset{(iii)}{\leq} \sum_{i=1}^{t} \alpha_t^i \mathbb{E}_{\pi_{k^i,h}} \left[r_h^{(j)}(s, a_{k^i,h}) + \sigma_{\mathcal{P}_h(s_h, a_{k^i,h})}(\boldsymbol{V}^{(j)}_{\hat{\pi}_{k^i,h+1},h+1}) \right] \\
&\overset{(iv)}{=} \boldsymbol{V}^{(j)}_{\hat{\pi}_{k,h},h}(s),
\end{aligned}
$$

where (i) uses eq. (33), (ii) uses eq. (34), (iii) uses eq. (24) and the assumption that $\boldsymbol{V}^{(j)}_{\hat{\pi}_{k^i},h+1}(s) \geq \underline{V}^{(j)}_{k^i,h+1}(s)$ holds for $\ell = k^i \leq k-1$, and (iv) uses the robust Bellman equation and the definition of $\hat{\pi}_{k,h}$ given by Algorithm 4. □

F.1 Lemmas on Robust Correlated equilibrium

The following lemma follows Lemma 15 of (Jin et al., 2022a), with the bandit input changed from $(a_h, \frac{H - r_h - V_{h+1}(s_{h+1})}{H})$ to $(a_h, \frac{H - r_h - \widehat{\sigma}_{\mathcal{P}_h(s_h,a_h)}(V_{h+1})}{H})$.

Lemma F.5. *Let $\pi_{k+1,h}$ be the policy given by the ADV_BANDIT_UPDATE algorithm at the h-th step of the k-th episode. Then the following bound holds for all $j \in [m]$, $k \in [K]$, $h \in [H]$ and $s \in \mathcal{S}$ with probability at least $1-\delta$ under Lemma C.1.*

$$
\begin{aligned}
&\max_{\phi^{(j)}} \sum_{i=1}^{t} \alpha_t^i \Big[\mathbb{E}_{a \sim \phi^{(j)} \circ (\pi_{k^i,h})} [r_h^{(j)}(s,a) + \sigma_{\mathcal{P}_h(s,a)}(V^{(j)}_{k^i,h+1})] \Big] \\
\leq &\sum_{i=1}^{t} \alpha_t^i \Big[\mathbb{E}_{a \sim \pi_{k^i,h}} [r_h^{(j)}(s,a) + \sigma_{\mathcal{P}_h(s,a)}(V^{(j)}_{k^i,h+1})] \Big] + 10 A_j \sqrt{\frac{H^3}{t} \ln \frac{mKHSA_j^2}{\delta}},
\end{aligned} \tag{35}
$$

Proof. By applying Lemma C.1 to the loss function $l_i(a) = \frac{H - r_h(s,a) - \sigma_{\mathcal{P}_h(s,a)}(V_{h+1})}{H}$ for any s, we obtain that the following bound holds for all $k \in [K]$, $h \in [H]$ and $s \in \mathcal{S}$ with probability $1-\delta$ (we replace δ in the bound in Lemma C.1 with $\frac{\delta}{KHS}$ by applying union bound to all $k \in [K]$, $h \in [H]$ and $s \in \mathcal{S}$), which is equivalent to the above bound and thus concludes the proof.

$$
\begin{aligned}
&\max_{\phi^{(j)}} \sum_{i=1}^{t} \alpha_t^i \mathbb{E}_{a \sim \pi_{k^i,h}} \Big[\frac{H - r_h^{(j)}(s,a) - \sigma_{\mathcal{P}_h(s,a)}(V^{(j)}_{k^i,h+1})}{H} \Big] \\
\leq &\max_{\phi^{(j)}} \sum_{i=1}^{t} \alpha_t^i \mathbb{E}_{a \sim \phi^{(j)} \circ \pi_{k^i,h}} \Big[\frac{H - r_h^{(j)}(s,a) - \sigma_{\mathcal{P}_h(s,a)}(V^{(j)}_{k^i,h+1})}{H} \Big] \\
&+ 10 A_j \sqrt{\frac{H}{t} \ln \frac{mKHSA_j^2}{\delta}}.
\end{aligned}
$$

□

Lemma F.6. *For the j-th player, the V-tables $\boldsymbol{V}^{(j)}_{\phi^* \circ \hat{\pi}_{k,h},h}(s) := \max_{\phi^{(j)}} \boldsymbol{V}^{(j)}_{\phi^{(j)} \circ \hat{\pi}_{k,h},h}(s)$ and $V^{(j)}_{k,h}(s)$ at h-th step in the k-th episode, satisfy the following inequality with probability at least $1-2\delta$ for any $\delta \in (0, \frac{1}{2})$*

$$
V^{(j)}_{k,h}(s) \geq \boldsymbol{V}^{(j)}_{\phi^* \circ \hat{\pi}_{k,h},h}(s)
$$

for all $s \in \mathcal{S}$.

Proof. It suffices to prove $\widetilde{V}^{(j)}_{k,h}(s) \geq \boldsymbol{V}^{(j)}_{\phi^* \circ \hat{\pi}_{k,h},h}(s)$; it implies $V^{(j)}_{k,h}(s) \geq \boldsymbol{V}^{(j)}_{\phi^* \circ \hat{\pi}_{k,h},h}(s)$, since $V^{(j)}_{k,h}(s_h) = \min\{H+1-h, \widetilde{V}^{(j)}_{k,h}(s_h)\}$ (see eq. (6)) and $\boldsymbol{V}^{(j)}_{\phi^* \circ \hat{\pi}_{k,h},h}(s) := \max_{\phi^{(j)}} \boldsymbol{V}^{(j)}_{\phi^{(j)} \circ \hat{\pi}_{k,h},h}(s) \leq$

$H+1-h$ (since $r_h^{(j)} \leq 1$ for all j,h). Let $\{k_i\}_{1\leq i\leq t}$ ($k_1 < k_2 < \ldots < k_t < k$) be the set of episodes where the state s is visited at the h-th step. Then we unroll the update rule (5) of $\widetilde{V}_{k,h}^{(j)}(s)$ with respect to the episode k as follows.

$$\begin{aligned}\widetilde{V}_{k,h}^{(j)}(s) &= \alpha_t^0(H-h+1) + \sum_{i=1}^{t}\alpha_t^i\left[r_h^{(j)}(s,a_{k^i,h}) + \widehat{\sigma}_{\mathcal{P}_h(s,a_{k^i,h})}(V_{k^i,h+1}^{(j)}) + \beta_i^{(j)}\right] \\ &\geq \sum_{i=1}^{t}\alpha_t^i\left[r_h^{(j)}(s,a_{k^i,h}) + \sigma_{\mathcal{P}_h(s,a_{k^i,h})}(V_{k^i,h+1}^{(j)}) + \beta_i^{(j)} - \mathfrak{e}\right].\end{aligned}$$

where the above $\geq$ uses $\alpha_t^0 = 0$ (see eq. (21)) and eq. (27). Substituting (34) which holds with probability at least $1-\delta$ into the above inequality, we obtain that the following bound holds for all j,k,h,s with probability at least $1-\delta$,

$$\begin{aligned}\widetilde{V}_{k,h}^{(j)}(s) \geq &\sum_{i=1}^{t}\alpha_t^i\mathbb{E}_{\pi_h^{k^i}}\left[r_h^{(j)}(s,a_{k^i,h}) + \sigma_{\mathcal{P}_h(s,a_{k^i,h})}(V_{k^i,h+1}^{(j)})\right] \\ &+ \sum_{i=1}^{t}\alpha_t^i(\beta_i^{(j)} - \mathfrak{e}) - \sqrt{\frac{H^3}{t}\ln\frac{2mKHS}{\delta}}. \end{aligned} \tag{36}$$

By substituting eq. (35) into eq. (36), we obtain the following bound which holds for all j,k,h,s with probability at least $1-2\delta$ (since eq. (35) holds with probability at least $1-\delta$ and so does eq. (36)).

$$\begin{aligned}\widetilde{V}_{k,h}^{(j)}(s) \geq &\max_{\phi^{(j)}}\sum_{i=1}^{t}\alpha_t^i\mathbb{E}_{\phi^{(j)}\circ\pi_{k^i,h}}\left[r_h^{(j)}(s,a_{k^i,h}) + \sigma_{\mathcal{P}_h(s,a_{k^i,h})}(V_{k^i,h+1}^{(j)})\right] \\ &+ \sum_{i=1}^{t}\alpha_t^i(\beta_i^{(j)} - \mathfrak{e}) - \sqrt{\frac{H^3}{t}\ln\frac{2mKHS}{\delta}} - 10A_j\sqrt{\frac{H^3}{t}\ln\frac{mKHSA_j^2}{\delta}} \\ \overset{(i)}{\geq} &\max_{\phi_h^{(j)}}\sum_{i=1}^{t}\alpha_t^i\mathbb{E}_{a\sim\phi_h^{(j)}\circ\pi_{k^i,h}}\left[r_h^{(j)}(s,a) + \sigma_{\mathcal{P}_h(s,a)}(V_{k^i,h+1}^{(j)})\right] \end{aligned} \tag{37}$$

where (i) holds using eq. (24).

Then we can apply induction to h backward to prove $\widetilde{V}_{k,h}^{(j)}(s) \geq \boldsymbol{V}_{\phi^*\circ\hat{\pi}_{k^i},h}^{(j)}$. For the base case $h = H+1$, it can be easily seen that $\widetilde{V}_{k,H+1}^{(j)}(s) = \boldsymbol{V}_{\phi^*\circ\hat{\pi}_k,H+1}^{(j)}(s) = 0$ based on Algorithm 1 and the definition of $\boldsymbol{V}_{\pi,h}^{(j)}$ given by eq. (2). Suppose $\widetilde{V}_{k,h+1}^{(j)}(s) \geq \boldsymbol{V}_{\phi^*\circ\hat{\pi}_k,h+1}^{(j)}(s)$ for a certain fixed h and all j,k,s, so $V_{k,h+1}^{(j)}(s) \geq \boldsymbol{V}_{\phi^*\circ\hat{\pi}_k,h+1}^{(j)}(s)$. Then eq. (37) further implies the following inequality, which concludes the proof. (Note that the whole induction builds on eq. (37) which holds for all k,h,j,s with probability at least $1-2\delta$.)

$$\widetilde{V}_{k,h}^{(j)}(s) \geq \max_{\phi_h^{(j)}}\sum_{i=1}^{t}\alpha_t^i\mathbb{E}_{a\sim\phi_h^{(j)}\circ\pi_{k^i,h}}\left[r_h^{(j)}(s,a) + \sigma_{\mathcal{P}_h(s,a)}(\boldsymbol{V}_{\phi^*\circ\hat{\pi}_{k^i,h+1},h+1}^{(j)})\right] \geq \boldsymbol{V}_{\phi^*\circ\hat{\pi}_{k^i,h},h}^{(j)},$$

where the second $\leq$ uses the following inequality obtained via the same proof logic as Lemma 13 of Jin et al. (2022a) (see the beginning of page 19 of Jin et al. (2022a)).

$$\begin{aligned}&\boldsymbol{V}_{\phi^*\circ\hat{\pi}_{k,h},h}^{(j)}(s) := \max_{\phi^{(j)}}\boldsymbol{V}_{\phi^{(j)}\circ\hat{\pi}_{k,h},h}^{(j)}(s) \\ &\overset{(i)}{=} \max_{\phi_h^{(j)}}\max_{\phi_{(h+1):H}^{(j)}}\mathbb{E}_{a\sim\phi_h^{(j)}\circ[\hat{\pi}_{k,h}]_h}\left(r_h^{(j)}(s,a) + \sigma_{\mathcal{P}_h(s,a)}\big(\boldsymbol{V}_{\phi_{(h+1):H}^{(j)}\circ\hat{\pi}_{k,h+1},h+1}^{(j)}\big)\right) \\ &\overset{(ii)}{=} \max_{\phi_h^{(j)}}\max_{\phi_{(h+1):H}^{(j)}}\sum_{i=1}^{t}\alpha_t^i\mathbb{E}_{a\sim\phi_h^{(j)}\circ\pi_{k^i,h}}\left(r_h^{(j)}(s,a) + \sigma_{\mathcal{P}_h(s,a)}\big(\boldsymbol{V}_{\phi_{(h+1):H}^{(j)}\circ\hat{\pi}_{k^i,h+1},h+1}^{(j)}\big)\right)\end{aligned}$$

$$\overset{(iii)}{\leq} \max_{\phi_h^{(j)}} \sum_{i=1}^{t} \alpha_t^i \mathbb{E}_{a\sim\phi_h^{(j)}\circ\pi_{k^i,h}} \Big(r_h^{(j)}(s,a) + \max_{\phi_{(h+1):H}^{(j)}} \inf_{\widetilde{\mathbb{P}}_h(\cdot|s,a)\in\mathcal{P}_h(s,a)} \sum_{s'} \widetilde{\mathbb{P}}_h(s'|s,a) \boldsymbol{V}^{(j)}_{\phi_{(h+1):H}^{(j)}\circ\hat{\pi}_{k^i,h+1},h+1}(s') \Big)$$

$$\leq \max_{\phi_h^{(j)}} \sum_{i=1}^{t} \alpha_t^i \mathbb{E}_{a\sim\phi_h^{(j)}\circ\pi_{k^i,h}} \Big(r_h^{(j)}(s,a) + \inf_{\widetilde{\mathbb{P}}_h(\cdot|s,a)\in\mathcal{P}_h(s,a)} \sum_{s'} \widetilde{\mathbb{P}}_h(s'|s,a) \max_{\phi_{(h+1):H}^{(j)}} \boldsymbol{V}^{(j)}_{\phi_{(h+1):H}^{(j)}\circ\hat{\pi}_{k^i,h+1},h+1}(s') \Big)$$

$$\overset{(iv)}{=} \max_{\phi_h^{(j)}} \sum_{i=1}^{t} \alpha_t^i \mathbb{E}_{a\sim\phi_h^{(j)}\circ\pi_{k^i,h}} \Big(r_h^{(j)}(s,a) + \sigma_{\mathcal{P}_h(s,a)}\big(\boldsymbol{V}^{(j)}_{\phi^*\circ\hat{\pi}_{k^i,h+1},h+1}\big) \Big),$$

where (i) uses robust Bellman equation and denotes $[\hat{\pi}_{k,h}]_h$ as the marginal distribution of a_h based on policy $\hat{\pi}_{k,h}$ defined by Algorithm 4, (ii) uses the definition of $\hat{\pi}_{k,h}$ given by Algorithm 4, (iii) and (iv) use the definition of $\sigma_{\mathcal{P}_h(s,a)}$ given by eq. (3).

□

F.2 Key Lemmas to Handle Uncertainty

Lemma F.7.

$$\begin{aligned}
&\sum_{k'=1}^{K} \Big(\sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(V^{(j)}_{k',h+1}) - \sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(\underline{V}^{(j)}_{k',h+1}) \Big) \\
&- \sum_{k'=1}^{K} \Big(V^{(j)}_{k',h+1}(s_{k',h+1}) - \underline{V}^{(j)}_{k',h+1}(s_{k',h+1}) \Big) \\
\leq & D \sum_{k=1}^{K} \sum_{s\in\mathcal{S}} \Big(V^{(j)}_{k,h+1}(s) - \underline{V}^{(j)}_{k,h+1}(s) \Big) + \sqrt{32KH^2 \ln\frac{2mH}{\delta}}.
\end{aligned} \tag{38}$$

Proof. For the k'-th episode, let $\sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(\underline{V}^{(j)}_{k',h+1}) = \sum_s p_{k',h}(s)\underline{V}^{(j)}_{k',h+1}(s) = p^T_{k',h}\underline{V}^{(j)}_{k',h+1}$, i.e., the minimum is achieved at $p^T_{k',h}$. Then

$$\begin{aligned}
&\sum_{k'=1}^{K} \Big(\sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(V^{(j)}_{k',h+1}) - \sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(\underline{V}^{(j)}_{k',h+1}) \Big) \\
&- \sum_{k'=1}^{K} \Big(V^{(j)}_{k',h+1}(s_{k',h+1}) - \underline{V}^{(j)}_{k',h+1}(s_{k',h+1}) \Big) \\
\leq & \sum_{k'=1}^{K} p^T_{k',h} \Big(V^{(j)}_{k',h+1} - \underline{V}^{(j)}_{k',h+1} \Big) - \sum_{k'=1}^{K} \Big(V^{(j)}_{k',h+1}(s_{k',h+1}) - \underline{V}^{(j)}_{k',h+1}(s_{k',h+1}) \Big) \\
\leq & \underbrace{\sum_{k'=1}^{K} p^T_{k',h} \Big(V^{(j)}_{k',h+1} - \underline{V}^{(j)}_{k',h+1} \Big) - \sum_{k'=1}^{K} \mathbb{E}\Big(V^{(j)}_{k',h+1}(s_{k',h+1}) - \underline{V}^{(j)}_{k',h+1}(s_{k',h+1}) \Big| s_{k',h}, a_{k',h} \Big)}_{(A)} \\
&+ \underbrace{\sum_{k'=1}^{K} \mathbb{E}\Big(V^{(j)}_{k',h+1}(s_{k',h+1}) - \underline{V}^{(j)}_{k',h+1}(s_{k',h+1}) \Big| s_{k',h}, a_{k',h} \Big) - \sum_{k'=1}^{K} \Big(V^{(j)}_{k',h+1}(s_{k',h+1}) - \underline{V}^{(j)}_{k',h+1}(s_{k',h+1}) \Big)}_{(B)}.
\end{aligned}$$

Then we will bound terms (A) and (B), respectively. For term (A), we define $p'_{k,h}(s) := \mathbb{P}_h(s_{k,h+1} = s | s_{k,h}, a_{k,h})$ for some distribution sampled from the uncertainty set $\mathcal{P}_h(s_{k,h}, a_{k,h})$. Then we obtain

$$\sum_{k'=1}^{K} p^T_{k',h} \Big(V^{(j)}_{k',h+1} - \underline{V}^{(j)}_{k',h+1} \Big) - \sum_{k'=1}^{K} \mathbb{E}\Big(V^{(j)}_{k,h+1}(s_{k',h+1}) - \underline{V}^{(j)}_{k',h+1}(s_{k',h+1}) \Big| s_{k',h}, a_{k',h} \Big)$$

$$
\begin{aligned}
&\overset{(i)}{=} \sum_{k'=1}^{K} p_{k',h}^{T}\left(V_{k',h+1}^{(j)}-\underline{V}_{k',h+1}^{(j)}\right)-\sum_{k'=1}^{K} p_{k',h}^{\prime T}\left(V_{k',h+1}^{(j)}-\underline{V}_{k',h+1}^{(j)}\right)\\
&\overset{(ii)}{=} \sum_{k'=1}^{K}\left(p_{k',h}-p_{k',h}^{\prime}\right)^{T}\left(V_{k',h+1}^{(j)}-\underline{V}_{k',h+1}^{(j)}\right)\\
&\overset{(iii)}{\leq} \max_{s\in\mathcal{S}}\left|p_{k',h}(s)-p_{k,h}^{\prime}(s)\right| \sum_{k'=1}^{K}\sum_{s\in\mathcal{S}}\left(V_{k',h+1}^{(j)}(s)-\underline{V}_{k',h+1}^{(j)}(s)\right)\\
&\overset{(iv)}{\leq} D\sum_{k=1}^{K}\sum_{s\in\mathcal{S}}\left(V_{k,h+1}^{(j)}(s)-\underline{V}_{k,h+1}^{(j)}(s)\right),
\end{aligned}
$$

where (i) expands the conditional expectation, (ii) combines the same term together, (iii) applies the Hölder's inequality $\langle u, v\rangle \leq \|u\|_\infty \|v\|_1$, and (iv) uses $\mathfrak{e} := \sup_{h,s,a,V}\left|\sigma_{\mathcal{P}_h(s,a)}(V)-\widehat{\sigma}_{\mathcal{P}_h(s,a)}(V)\right|$ defined by Definition 4.3.

Now we turn to bound term (B). Let

$$
Y_{k'}=\mathbb{E}\left(V_{k',h+1}^{(j)}(s_{k',h+1})-\underline{V}_{k',h+1}^{(j)}(s_{k',h+1})\middle| s_{k',h},a_{k',h}\right)-\left(V_{k',h+1}^{(j)}(s_{k',h+1})-\underline{V}_{k',h+1}^{(j)}(s_{k',h+1})\right).
$$

Then $\sum_{k'=1}^{k} Y_{k'}$ forms a martingale with $|Y_{k'}| \leq 4H$. By Azuma-Hoeffding inequality, with a probability at least $1-\delta$,

$$
\sum_{k'=1}^{K} Y_{k'} \leq \sqrt{32KH^2\ln\frac{2mH}{\delta}}.
$$

Combining the bounds of (A) and (B), we obtain the upper bound of eq. (13):

$$
\begin{aligned}
&\sum_{k'=1}^{K}\left(\sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(V_{k',h+1}^{(j)})-\sigma_{\mathcal{P}_h(s_{k,h},a_{k',h})}(\underline{V}_{k',h+1}^{(j)})\right)\\
\leq& D\sum_{k=1}^{K}\sum_{s\in\mathcal{S}}\left(V_{k,h+1}^{(j)}(s)-\underline{V}_{k,h+1}^{(j)}(s)\right)+\sqrt{32KH^2\ln\frac{2mH}{\delta}}.
\end{aligned}
$$

□

This lemma gives a more general version of recursion used in Jin et al. (2022a). When setting $b_h \equiv 0$ and iterating to $h=1$, this result is reduced to Jin et al. (2022a).

Lemma F.8. *Suppose the sequence $\{\mathsf{a}_h, \mathsf{b}_h\}_{h\in[H+1]}$ satisfies the following recursion:*

$$
\begin{aligned}
&\mathsf{a}_{H+1}=\mathsf{b}_{H+1}=0,\\
&\mathsf{a}_h \leq \mathsf{C}_1\mathsf{a}_{h+1}+\mathsf{C}_2\mathsf{b}_{h+1}+\mathsf{C}_3.
\end{aligned}
$$

Then for any $h\in[H]$,

$$
\mathsf{a}_h \leq \mathsf{C}_2\sum_{i=0}^{H-h}\mathsf{C}_1^i\mathsf{b}_{h+1+i}+\Big(\frac{\mathsf{C}_1^{H-h+1}-1}{\mathsf{C}_1-1}\Big)\mathsf{C}_3.
$$

Proof. We prove it by induction with respect to h backward. For $h=H$, the statement obviously holds. Then assuming the statement holds for $h+1$ (for some $h<H$), we consider the upper bound of a_h:

$$
\begin{aligned}
\mathsf{a}_h &\overset{(i)}{\leq} \mathsf{C}_1\mathsf{a}_{h+1}+\mathsf{C}_2\mathsf{b}_{h+1}+\mathsf{C}_3\\
&\overset{(ii)}{\leq} \mathsf{C}_1\Big[\mathsf{C}_2\sum_{i=0}^{H-h-1}\mathsf{C}_1^i\mathsf{b}_{h+2+i}+\Big(\frac{\mathsf{C}_1^{H-h}-1}{\mathsf{C}_1-1}\Big)\mathsf{C}_3\Big]+\mathsf{C}_2\mathsf{b}_{h+1}+\mathsf{C}_3\\
&\overset{(iii)}{=} \mathsf{C}_2\Big[\sum_{i=0}^{H-h-1}\mathsf{C}_1^{i+1}\mathsf{b}_{h+2+i}+\mathsf{b}_{h+1}\Big]+\Big[\mathsf{C}_1\Big(\frac{\mathsf{C}_1^{H-h}-1}{\mathsf{C}_1-1}\Big)+1\Big]\mathsf{C}_3
\end{aligned}
$$

$$= \mathsf{C}_2 \sum_{i=0}^{H-h} \mathsf{C}_1^i \mathsf{b}_{h+1+i} + \Big(\frac{\mathsf{C}_1^{H-h+1}-1}{\mathsf{C}_1-1}\Big)\mathsf{C}_3,$$

where (i) uses the recursion, (ii) applies the induction hypothesis, and (iii) rearranges the order of each term. It completes the proof. □

Lemma F.9. *Suppose the sequence* $\{\mathsf{a}_h, \mathsf{b}_h\}_{h\in[H+1]}$ *satisfies the following recursion:*

$$\mathsf{a}_{H+1} = 0,$$
$$\mathsf{a}_h \le \mathsf{C}_1 + \mathsf{C}_2 \sum_{i=0}^{H-h} \Big(1+\frac{1}{H}\Big)^{i+1} \mathsf{a}_{h+1+i}.$$

If $\mathsf{C}_2 < 1/H$*, then*

$$\max_h \mathsf{a}_h \le \frac{\mathsf{C}_1}{1-H\mathsf{C}_2}.$$

Proof. We re-write the recursion in matrix form. Here inequality holds for entry-wise.

$$\begin{bmatrix}\mathsf{a}_1\\ \mathsf{a}_2\\ \vdots\\ \mathsf{a}_H\end{bmatrix} \le \mathsf{C}_1 \mathbf{1}_H + \mathsf{C}_2 \begin{bmatrix} 0 & \left(1+\frac{1}{H}\right) & \left(1+\frac{1}{H}\right)^2 & \dots & \left(1+\frac{1}{H}\right)^{H-1} \\ 0 & 0 & \left(1+\frac{1}{H}\right) & \dots & \left(1+\frac{1}{H}\right)^{H-2} \\ 0 & 0 & 0 & \dots & \left(1+\frac{1}{H}\right)^{H-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix}\mathsf{a}_1\\ \mathsf{a}_2\\ \vdots\\ \mathsf{a}_H\end{bmatrix}.$$

Denote the upper triangular Toeplitz matrix by T. Then we take $\|\cdot\|_\infty$ on both sides and obtain

$$\begin{aligned}\max_h \mathsf{a}_h &\le \mathsf{C}_1 + \mathsf{C}_2 \|\mathsf{T}\|_\infty \max_h \mathsf{a}_h \\ &\le \mathsf{C}_1 + \mathsf{C}_2 \sum_{h'=1}^{H-1} \Big(1+\frac{1}{H}\Big)^{h'} \max_h \mathsf{a}_h \\ &\le \mathsf{C}_1 + \mathsf{C}_2 H \Big[\Big(1+\frac{1}{H}\Big)^H - 1\Big] \max_h \mathsf{a}_h \\ &\le \mathsf{C}_1 + H\mathsf{C}_2 \max_h \mathsf{a}_h.\end{aligned}$$

When $\mathsf{C}_2 < 1/H$, it solves the upper bound of $\max_h \mathsf{a}_h$ as

$$\max_h \mathsf{a}_h \le \frac{\mathsf{C}_1}{1-H\mathsf{C}_2}.$$

□

G Experiments Setup

In this section, we clarify the details of the experiment setup. For fair comparison, both V-learning and robust V-learning are implemented in the same way as follows:

1. Adversarial bandit algorithm: the maximum size of saved $\hat{\ell}_j$ in Algorithm 2 is limited to 10000; for example, when reaching the iterate $i = 10001$, we will remove the first element to save more computation memory.
2. Hyper-parameters: Both $\{\alpha_t\}$ and $\{\beta_t^{(j)}\}$ depend on environment parameters S, A, H and the high-probability bound parameter δ. For simplicity, we set the same fixed coefficients for V-learning and robust V-learning.
3. Solve $\sigma(V)$ for KL-type uncertainty set: For numerical stability, the minimum is solved over a bounded interval $[0.01, 9.9]$.

To simulate the non-stationary environment, we sample the transition probability as follows: (1) Pre-set a list of environments with the length n from the given uncertainty set. For discrete model, the uncertainty set contains exact two elements $\mathcal{P}_h = \{\mathbb{P}_{h,p} : p \in \{0, \frac{5}{14}\}\}$. For KL-divergence model and R-contamination model, we sample multiple Gaussian noises with the same dimension as the transition kernel; then we remove unqualified transition until there are 5 left. (2) We define the index function $i(h) = \lfloor \sin(h) + X_h \rfloor \%n$, where $\%$ is the module operation, n is the number of transitions, X_h is a Gaussian random variable. For each step h, we choose the transition selected by the index function to sample the next state.

For KL-divergence and R-contamination model, exactly solving the optimality gap over the whole uncertainty set is not possible. The steps below are used to calculate the optimality gap of an output policy π numerically in our experiment: (1) Evenly draw 5 transition kernels from the uncertainty set. (2) For each sampled transition and each player, we estimate the expected future return of all deterministic modifications of π starting from the initial state by running the environment for 100 times. (3) We choose the modification with the highest worst-case expected future return among all simulations; this is the best response of π. (4) The gap between the best response of π and the V-table of π at the initial state is the optimality gap. We use this to evaluate the performance of all models.

H Additional Experiments

We add additional experiments on different parameters:

1. Discrete uncertainty model: (a) $\mathcal{P}_h = \{\mathbb{P}_{h,p} : p \in \{0, 0.1\}\}$; (b) $\mathcal{P}_h = \{\mathbb{P}_{h,p} : p \in \{0, 0.2\}\}$; and (c) $\mathcal{P}_h = \{\mathbb{P}_{h,p} : p \in \{0, \frac{5}{14}\}\}$.
2. KL divergence model in Example 3.1: We set the centroid transition kernel to be $\mathbb{P}_h = \mathbb{P}_{1,0.1}$ and choose uncertainty level parameter (a) $\rho = 0.15$; (b) $\rho = 0.12$; and (c) $\rho = 0.1$.
3. R-contamination model in Example 3.2: We set the centroid transition kernel to be $\mathbb{P}_h = \mathbb{P}_{1,0.1}$ and choose uncertainty level parameter (a) $R = 0.03$; (b) $R = 0.02$; and (c) $R = 0.01$.

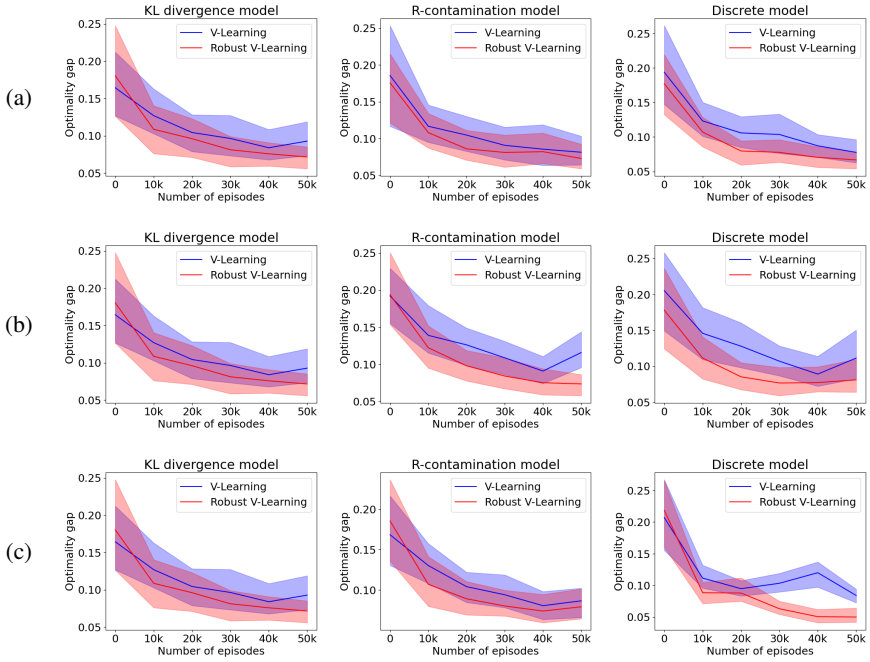

Figure 3: Comparison of estimated optimality gap of the policies produced by V-learning and robust V-learning. The optimality gap we estimate is $\max_{j\in[J]}[\boldsymbol{V}^{(j)}_{\phi^*\circ\pi_{K+1},1}(s_4) - \boldsymbol{V}^{(j)}_{\pi_{K+1},1}(s_4)]$.