Q CAN PLAY THAT GAME: ONLINE FITTED Q-ITERATION FOR CONTINUOUS-ACTION ZERO-SUM MARKOV GAMES WITH CONVEX-CONCAVE FUNCTION APPROXIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Zero-sum Markov games arise in a wide variety of sequential decision-making problems such as adversarial learning and planning against modeled uncertainties. However, prior work on finite-sample guarantees on the learned state-action value function (Q-function) for zero-sum Markov games is largely restricted to finite-action settings, or to continuous-action games in which agents have linear dynamics and quadratic rewards (i.e. linear-quadratic, or LQ, games). A central challenge in extending such guarantees to more general continuous-action zero-sum Markov games is that the associated Bellman operator involves a minimax problem that need not admit a tractable saddle-point solution. To this end, we first introduce a class of neural network function approximators for the Q-function that is convex-concave in the players' actions, which guarantees that this minimax problem admits a pure-strategy saddle point. We then study an online variant of fitted Q-iteration employing this function class and establish, to the best of our knowledge, the first finite-sample guarantees for non-LQ zero-sum Markov games with continuous states and actions.

1 INTRODUCTION

We consider the problem of learning state-action value functions (Q-functions) in discounted, infinite-horizon, two-player zero-sum Markov games with *continuous* state-action spaces. Such games naturally arise in competitive scenarios and robust decision-making, where environmental uncertainties and disturbances are adversarially modeled (Zhou et al., 1996; Morimoto & Doya, 2000; Pinto et al., 2017). Prior work has established finite-sample guarantees for zero-sum Markov games in the finite-action setting (Lagoudakis & Parr, 2002; Perolat et al., 2015; Zhao et al., 2022; Fan et al., 2020; Zhang et al., 2021a; Jia et al., 2019; Sidford et al., 2020; Xie et al., 2020; Alacaoglu et al., 2022; Chen et al., 2024; Feng et al., 2024). However, extending these results to the continuous state-action setting remains challenging. The difficulty stems from the *minimax* problem on Q inside the Bellman operator. At every state, each player wishes to play a security strategy, meaning that one player's policy solves a $\min\max$ problem and the other's solves a $\max\min$. The solutions to these two distinct problems are mutually consistent at *saddle points*, where both players have non-exploitable security strategies. Saddle points are guaranteed to exist when policies map from state to *mixed strategies* over the action space. In finite-action spaces, the resulting saddle-point problem can be solved via a linear program; in continuous action spaces however, it is generally intractable to compute mixed-strategy saddle points as mixed-strategy spaces are infinite-dimensional. Moreover, when policies are restricted to output *pure strategies* at each state, i.e. raw actions, saddle-point solutions generally do not exist for arbitrary Q.

In this work, we propose a new class of neural network function approximators of the Q-function which is convex in the minimizing player's action and concave in the maximizing player's action. This convex-concave structure addresses the challenge above, as the minimax theorem guarantees that a pure-strategy saddle point exists for the minimax problem inside the Bellman operator. Further, this pure-strategy saddle point can also be reliably computed up to any desired accuracy using existing minimax optimization methods (Nemirovski, 2004; Lin et al., 2020). We analyze an on-

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

line variant of minimax fitted Q-iteration which employs this convex-concave function class and establish finite-sample bounds between the learned Q-function and the fixed point of the Bellman operator, assuming that the fixed point admits pure-strategy saddle points.

The prior work closest to ours establishes finite-sample bounds for online minimax fitted Q-iteration with neural network function approximation, but is restricted to finite-action spaces (Fan et al., 2020). In the continuous-action setting, single-player reinforcement learning literature has developed finite-sample analysis for fitted Q-iteration with function approximation (Antos et al., 2007). However, extending these results to the zero-sum Markov game setting remains underexplored. Within the game setting we consider, theoretical finite-sample results for continuous state-action domains remain largely confined to the linear-quadratic (LQ) setting, where agents have linear transition dynamics and quadratic objectives (Başar & Olsder, 1998; Al-Tamimi et al., 2007a;b; Zhang et al., 2019; Bu et al., 2019; Zhang et al., 2021b). These results exploit the fact that in the LQ setting, the optimal Q-function and the resulting policies admit a closed-form solution via the algebraic Riccati equations (Başar & Olsder, 1998). Unfortunately, this Riccati structure does not extend beyond the LQ setting. In this work we move beyond the LQ setting, and make the following contributions:

- **Contribution 1.** We propose a new class of neural network function approximators for the Q-function of zero-sum Markov games. Building on the partially input convex neural network architecture (Amos et al., 2017), this class is convex in the minimizing player's actions and concave in the maximizing player's actions by construction. The convex-concave structure ensures the existence of a saddle-point solution to the minimax problem in the Bellman operator.
- **Contribution 2.** We analyze batched minimax Q-learning employing the above function approximation class, and establish finite-sample bounds comparing the learned Q-function to the fixed point of the Bellman operator, assuming the fixed point admits pure-strategy saddle points. To the best of our knowledge, this is the first such bound for neural network function approximation in zero-sum Markov games with continuous state-action spaces beyond the LQ setting.

Additionally, we empirically investigate whether these convex-concave neural networks lead to stable training when used on a practical version of modified fitted Q-iteration.

2 PRELIMINARIES

Notation. $\mathbb{Z}_{\geq 0}$ denotes the set of non-negative integers, and for $n \in \mathbb{Z}_{\geq 0}$, $[n]$ denotes the set $\{1, 2, \cdots, n\}$. $\mathcal{P}(\mathcal{S})$ denotes the set of all probability measures over a measurable space $\mathcal{S}$. $M(\mathcal{S})$ denotes the space of bounded measurable functions with domain $\mathcal{S}$. For $f \in M(\mathcal{X})$ and a probability measure $\sigma \in \mathcal{P}(\mathcal{X})$, we denote $\|f\|_{2,\sigma} = \sqrt{\mathbb{E}_{\boldsymbol{x}\sim\sigma}\left[f(\boldsymbol{x})^2\right]}$. For a measure μ absolutely continuous with respect to another measure σ, i.e., $\mu \ll \sigma$, $d\mu/d\sigma$ denotes the Radon-Nikodym derivative. We denote a sequence of vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_j$ in short as $\boldsymbol{v}_{1:j}$. We denote the Lebesgue measure by λ, and we denote the Lebesgue measure of an ℓ^2-ball in d dimensions with radius η by $\alpha_d \eta^d$.

We consider two-player, zero-sum, infinite-horizon discounted Markov games given by the tuple $(\mathcal{S}, \mathcal{U}, \mathcal{V}, P, r, \gamma)$, where $\mathcal{S} \subset \mathbb{R}^{d_s}$ is the continuous state space, and $\mathcal{U} \subset \mathbb{R}^{d_u}$ and $\mathcal{V} \subset \mathbb{R}^{d_v}$ are the continuous action spaces of the minimizing and maximizing players, respectively. We take $\mathcal{S}, \mathcal{U}, \mathcal{V}$ to be convex and compact. $P : (\mathcal{S} \times \mathcal{U} \times \mathcal{V}) \to \mathcal{P}(\mathcal{S})$ denotes the transition dynamics of the game, $r : (\mathcal{S} \times \mathcal{U} \times \mathcal{V}) \to \mathbb{R}$ is the immediate reward function, and $\gamma \in (0, 1)$ is the discounting factor. Let player 1 and player 2 be the minimizing and maximizing players respectively. At each time $t \in \mathbb{Z}_{\geq 0}$, at some game state $\boldsymbol{s}_t \in \mathcal{S}$, players 1 and 2 choose actions $\boldsymbol{u}_t \in \mathcal{U}$ and $\boldsymbol{v}_t \in \mathcal{V}$. This yields an immediate reward of $r(\boldsymbol{s}_t, \boldsymbol{u}_t, \boldsymbol{v}_t)$ and the game proceeds to the next state $\boldsymbol{s}_{t+1} \sim P(\cdot|\boldsymbol{s}_t, \boldsymbol{u}_t, \boldsymbol{v}_t)$. Player 1 chooses actions to minimize the expected discounted cumulative return $\mathcal{R} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{u}_t, \boldsymbol{v}_t)\right]$, while player 2 chooses actions to maximize $\mathcal{R}$.

Each player chooses its actions according to a policy. We consider deterministic Markovian policies induced by a state-action value function (Q-function), in which players are restricted to pure strategies. For a given continuous function $Q : (\mathcal{S} \times \mathcal{U} \times \mathcal{V}) \to \mathbb{R}$, at state $\boldsymbol{s}$, player 1 chooses $\pi_Q(\boldsymbol{s}) \in \arg\min_{\boldsymbol{u}\in\mathcal{U}} \max_{\boldsymbol{v}\in\mathcal{V}} Q(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v})$ and player 2 chooses $\mu_Q(\boldsymbol{s}) \in \arg\max_{\boldsymbol{v}\in\mathcal{V}} \min_{\boldsymbol{u}\in\mathcal{U}} Q(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v})$. These are the players' security strategies: player 1 is guaran-

teed a value no worse than $\min_{\boldsymbol{u}\in\mathcal{U}} \max_{\boldsymbol{v}\in\mathcal{V}} Q(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})$ and player 2 is guaranteed a value no worse than $\max_{\boldsymbol{v}\in\mathcal{V}} \min_{\boldsymbol{u}\in\mathcal{U}} Q(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})$. For such games the *minimax Bellman operator* $\mathcal{T} : M(\mathcal{S}\times\mathcal{U}\times\mathcal{V}) \to M(\mathcal{S}\times\mathcal{U}\times\mathcal{V})$ is defined by

$$\mathcal{T}Q(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}) = r(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}) + \gamma\mathbb{E}_{\boldsymbol{s}'\sim P(\cdot|\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})}\left[\inf_{\tilde{\boldsymbol{u}}\in\mathcal{U}} \sup_{\tilde{\boldsymbol{v}}\in\mathcal{V}} Q(\boldsymbol{s}',\tilde{\boldsymbol{u}},\tilde{\boldsymbol{v}})\right]. \tag{1}$$

It is well known that the minimax Bellman operator is contractive in the L_∞-norm and thus a unique fixed point Q^* exists such that $\mathcal{T}Q^* = Q^*$ (Shapley, 1953; Perolat et al., 2015). Equation (1) reveals that in order to compute the Bellman operator for continuous functions, one needs to solve a *minimax* problem, in this case corresponding to player 1's security strategy π_Q. One can analogously define the *maximin* Bellman operator associated with player 2's security strategy μ_Q. While both minimax and maximin problems admit solutions for compact $\mathcal{U}$ and $\mathcal{V}$ and continuous Q, these solutions need not coincide, i.e., (π_Q, μ_Q) need not be a saddle-point pair for generic Q. We restrict Q to a convex-concave function approximation class, convex in the player 1's action and concave in player 2's action so that the minimax theorem guarantees that at every state, (π_Q, μ_Q) forms a saddle point for every Q in our function class (Sion, 1958). Further, these saddle points can be found up to any desired accuracy using existing minimax optimization methods (Nemirovski, 2004; Lin et al., 2020).

Remark 1 (Mixed Strategies)**.** *An alternative route to ensure existence of a saddle-point solution is to let players employ mixed strategies, where they randomize over pure strategies. In the finite-action setting, mixing ensures that a saddle point of the minimax problem always exists, and it can be computed by solving a linear program at every state $\boldsymbol{s}$. In the continuous-action setting, a saddle point over mixed strategies still exists under mild continuity assumptions (Glicksberg, 1952), but computing it is far less straightforward than in the finite-action setting. The iterative double oracle algorithm (Adam et al., 2021) is guaranteed to return a saddle point up to any desired accuracy; however, each iteration requires computing a global best response for each player, which can be intractable under function approximation. Moreover, an explicit bound on the required number of iterations as a function of the saddle-point accuracy remains unknown. These difficulties motivate our focus on pure strategies with convex-concave function approximation instead.*

To represent the return $\mathcal{R}$ obtained by policies (π, μ), it is useful to define the *state* value function $V^{\pi,\mu} : \mathcal{S} \to \mathbb{R}$, given by $V^{\pi,\mu}(\boldsymbol{s}) = \mathbb{E}\big[\sum_{t=0}^{\infty}\gamma^t r\big(\boldsymbol{s}_t, \pi(\boldsymbol{s}_t), \mu(\boldsymbol{s}_t)\big)|\boldsymbol{s}_0 = \boldsymbol{s}\big]$. The state value function V^* relates to the fixed point Q^* of eq. (1) as $Q^*(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}) = r(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}) + \gamma\mathbb{E}_{\boldsymbol{s}'\sim P(\cdot|\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})}[V^*(\boldsymbol{s}')]$. The state value function can be used to determine whether the policies π, μ form a saddle-point equilibrium via the *Nash gap*. Let Π denote the set of pure Markovian strategies, then for an evaluation distribution $\rho \in \mathcal{P}(\mathcal{S})$, the Nash gap is defined by

$$\texttt{NashGap}_\rho(\pi,\mu) = \mathbb{E}_{\boldsymbol{s}\sim\rho}\left[\max_{\mu'\in\Pi} V^{\pi,\mu'}(\boldsymbol{s}) - \min_{\pi'\in\Pi} V^{\pi',\mu}(\boldsymbol{s})\right]. \tag{2}$$

The Nash gap is always non-negative, and is zero only when the input policies form a saddle-point Nash equilibrium with respect to ρ. Formally, $\texttt{NashGap}_\rho(\pi,\mu) = 0 \iff V^{\pi,\mu'}(\boldsymbol{s}) \le V^{\pi,\mu}(\boldsymbol{s}) \le V^{\pi',\mu}(\boldsymbol{s})$ for all $\pi' \in \Pi$, $\mu' \in \Pi$, and ρ-almost every $\boldsymbol{s} \in \mathcal{S}$. In particular, when the fixed point Q^* of the minimax Bellman operator (1) admits pure-strategy saddle points, the induced policies (π_{Q^*}, μ_{Q^*}) satisfy $\texttt{NashGap}_\rho(\pi_{Q^*}, \mu_{Q^*}) = 0$ for every evaluation distribution $\rho \in \mathcal{P}(\mathcal{S})$. The goal of this paper is to obtain bounds on the Q-function learned from a finite number of samples, and in turn bound the Nash gap corresponding to its induced security strategies.

3 Convex-Concave Function Approximation

We now describe the convex-concave neural network architecture used to approximate the Q-function: for every network parameter vector $\theta \in \Theta$, and every state $\boldsymbol{s} \in \mathcal{S}$, the Q-function network $Q_\theta : (\mathcal{S}\times\mathcal{U}\times\mathcal{V}) \to \mathbb{R}$ should be convex-concave, i.e., $Q_\theta(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})$ must be convex in $\boldsymbol{u} \in \mathcal{U}$ for every fixed $(\boldsymbol{s},\boldsymbol{v})$ and concave in $\boldsymbol{v} \in \mathcal{V}$ for every fixed $(\boldsymbol{s},\boldsymbol{u})$. To construct such a network, we build on *partially input convex neural networks* (PICNNs) (Amos et al., 2017), a network architecture that guarantees convexity in a designated subset of the network's inputs, while simultaneously leaving the dependence on the other inputs unconstrained. Formally, given two inputs $\boldsymbol{x}$ and $\boldsymbol{y}$ where convexity in $\boldsymbol{y}$ is desired, the PICNN architecture constructs a neural network $N_\theta(\boldsymbol{x},\boldsymbol{y})$, such that for

every choice of parameters θ, $N_\theta(\boldsymbol{x}, \boldsymbol{y})$ is convex in $\boldsymbol{y}$ for fixed $\boldsymbol{x}$, while retaining full expressivity in $\boldsymbol{x}$. PICNNs achieve this by constraining a subset of weights along the computation graph edges traceable back to $\boldsymbol{y}$ to be non-negative and pairing them with convex, non-decreasing activations, while leaving weights traceable to $\boldsymbol{x}$ unconstrained. We review the PICNN architecture in more detail in Section C.

To develop a *convex-concave* $Q_\theta(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v})$, we use three neural networks with parameters $\theta^1, \theta^2, \theta^3$, representing the concatenated parameters as $\theta = (\theta^1; \theta^2; \theta^3) \in \Theta$. We use two PICNNs $N^1_{\theta^1}(\boldsymbol{s}, \boldsymbol{u}) : (\mathcal{S} \times \mathcal{U}) \to \mathbb{R}$ (convex in the input $\boldsymbol{u}$) and $N^2_{\theta^2}(\boldsymbol{s}, \boldsymbol{v}) : (\mathcal{S} \times \mathcal{V}) \to \mathbb{R}$ (convex in the input $\boldsymbol{v}$), as well as a standard feedforward matrix-valued neural network $M_{\theta^3}(\boldsymbol{s}) : \mathcal{S} \to \mathbb{R}^{d_{\boldsymbol{u}} \times d_{\boldsymbol{v}}}$. We construct a convex-concave $Q_\theta(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v})$ as

$$Q_\theta(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}) := N^1_{\theta^1}(\boldsymbol{s}, \boldsymbol{u}) - N^2_{\theta^2}(\boldsymbol{s}, \boldsymbol{v}) + \boldsymbol{u}^\top M_{\theta^3}(\boldsymbol{s})\, \boldsymbol{v}. \tag{3}$$

Note that while the first two terms $N^1_{\theta^1}(\boldsymbol{s}, \boldsymbol{u}) - N^2_{\theta^2}(\boldsymbol{s}, \boldsymbol{v})$ alone would be convex in $\boldsymbol{u}$ and concave in $\boldsymbol{v}$, they would make Q_θ only additively separable in $\boldsymbol{u}$ and $\boldsymbol{v}$. The third, bilinear term $\boldsymbol{u}^\top M_{\theta^3}(\boldsymbol{s})\, \boldsymbol{v}$ captures interactions between the players' actions beyond the separable structure, while being simultaneously convex in $\boldsymbol{u}$ and concave in $\boldsymbol{v}$. Note that it may be possible to include coupling terms that are more expressive than a bilinear mapping. We do not pursue this direction here, as the bilinear term suffices to guarantee joint convexity-concavity for our theoretical result in Section 4, and we found it to be adequate for our experiments in Section 5 as well. Our theoretical results will need the the Lipschitz constant of Q_θ in the joint actions, as well as the covering number. We bound these for a particular architectural choice of N^i and M in Theorem 2 of Section C.2, and Theorem 3 of Section C.3 respectively.

4 Online Minimax Fitted Q Iteration

We now describe our modified minimax fitted Q-iteration procedure, given in Algorithm 1. We restrict the class of function approximators class to be the convex-concave neural network architecture described in Section 3, which we denote as $\mathcal{F}$. Further, we assume access to a minimax oracle $\mathcal{O}_\epsilon : \mathcal{F} \to (\mathcal{U} \times \mathcal{V})$. For a convex-concave objective $f(\boldsymbol{u}, \boldsymbol{v})$ and any desired $\epsilon > 0$, $\mathcal{O}_\epsilon$ returns an approximate saddle point $(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) \in (\mathcal{U} \times \mathcal{V})$ such that the gap $|\max_{\boldsymbol{v} \in \mathcal{V}} f(\tilde{\boldsymbol{u}}, \boldsymbol{v}) - \min_{\boldsymbol{u} \in \mathcal{U}} f(\boldsymbol{u}, \tilde{\boldsymbol{v}})| < \epsilon$. Several existing minimax optimization methods can be feasibly used as the oracle for $Q \in \mathcal{F}$, because $\mathcal{F}$ guarantees convexity-concavity. At the k^{th} iteration of Algorithm 1, when the parameter estimate for the Q-function is θ_k, we denote the corresponding network Q_{θ_k} as Q_k.

Algorithm 1 Online Minimax Fitted Q-Iteration

1: **Hyper-parameters:** Number of iterations K, batch size n.
2: **Inputs:** Markov game $(\mathcal{S}, \mathcal{U}, \mathcal{V}, P, r, \gamma)$, initial Q-function $Q_0 \in \mathcal{F}$, fixed sampling distribution $\sigma \in \mathcal{P}(\mathcal{S} \times \mathcal{U} \times \mathcal{V})$, minimax oracle $\mathcal{O}_\epsilon$.
3: **for** $k = 0$ to $K - 1$ **do**
4: Sample n i.i.d. samples $\{(\boldsymbol{s}_i, \boldsymbol{u}_i, \boldsymbol{v}_i)\}_{i=1}^n \sim \sigma$
5: Draw $\boldsymbol{s}'_i \sim P(\cdot \mid \boldsymbol{s}_i, \boldsymbol{u}_i, \boldsymbol{v}_i)$ for each $i \in [n]$
6: For each $i \in [n]$, form the target

$$(\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{v}}_i) = \mathcal{O}_\epsilon\big(Q_k(\boldsymbol{s}'_i, \cdot, \cdot)\big)$$
$$y_i = r(\boldsymbol{s}_i, \boldsymbol{u}_i, \boldsymbol{v}_i) + \gamma Q_k(\boldsymbol{s}'_i, \tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{v}}_i)$$

7: Solve the regression problem exactly

$$\widehat{L}_n(f; Q_k) := \frac{1}{n} \sum_{i=1}^n \big(f(\boldsymbol{s}_i, \boldsymbol{u}_i, \boldsymbol{v}_i) - y_i\big)^2,$$
$$Q_{k+1} := \arg\min_{f \in \mathcal{F}} \widehat{L}_n(f; Q_k)$$

8: **end for**
9: **Output:** Learned Q-function Q_K

Algorithm 1 differs from standard fitted Q-iteration in the sense that we use data in an *online* manner and do not reuse it. Specifically, at iteration k, using a fixed sampling distribution $\sigma \in \mathcal{P}(\mathcal{S}\times\mathcal{U}\times\mathcal{V})$, we sample n i.i.d. state-action tuples $(\boldsymbol{s}_i, \boldsymbol{u}_i, \boldsymbol{v}_i) \sim \sigma$ for $i \in [n]$. Using these states and actions, we sample the next states according to the transition dynamics $\boldsymbol{s}_i' \sim P(\,\cdot\,|\boldsymbol{s}_i, \boldsymbol{u}_i, \boldsymbol{v}_i)$ for $i \in [n]$. For each $i \in [n]$, we invoke the minimax oracle $\mathcal{O}_\epsilon$ on the fitted Q-function $Q_k \in \mathcal{F}$ evaluated at the next state $\boldsymbol{s}_i'$ to obtain an approximate saddle-point action pair $(\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{v}}_i) := \mathcal{O}_\epsilon\big(Q_k(\boldsymbol{s}_i', \cdot, \cdot)\big)$. Since Q_k is convex-concave from Equation (3), the minimax theorem guarantees that a saddle point of $Q_k(\boldsymbol{s}_i', \cdot, \cdot)$ exists, and $\mathcal{O}_\epsilon$ returns an ϵ-accurate approximation to it. We use $(\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{v}}_i)$ to form the regression target $y_i = r(\boldsymbol{s}_i, \boldsymbol{u}_i, \boldsymbol{v}_i) + \gamma Q_k(\boldsymbol{s}_i', \tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{v}}_i)$, which serves as an empirical estimate of an *approximate* minimax Bellman operator from Equation (1) applied to the sampled tuple $(\boldsymbol{s}_i, \boldsymbol{u}_i, \boldsymbol{v}_i)$. This approximation occurs because in practice, one can solve the $\min\max$ problem up to ϵ-accuracy instead of exactly. Finally, we update the Q-function estimate by solving a least-squares regression problem over $\mathcal{F}$:

$$Q_{k+1} := \arg\min_{f\in\mathcal{F}} \widehat{L}_n(f; Q_k) := \arg\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \left(f(\boldsymbol{s}_i, \boldsymbol{u}_i, \boldsymbol{v}_i) - y_i\right)^2. \tag{4}$$

Note that in Equation (4), the regression target y_i is treated as a constant independent of $\mathcal{F}$. Repeating this procedure for K iterations yields the final estimate Q_K returned by Algorithm 1.

Sources of error in Algorithm 1. We now shift our focus to analyzing how the learned Q_K compares to the fixed point of the minimax Bellman operator Q^*. It is pertinent to note that Algorithm 1 departs from an exact Bellman iteration $Q_{k+1} = \mathcal{T}Q_k$ in three ways, which introduce three corresponding sources of error: (i) an optimization error from only being able to solve the minimax problem up to ϵ-accuracy with $\mathcal{O}_\epsilon$, (ii) an approximation error from restricting to the convex-concave function class $\mathcal{F}$, and (iii) a statistical error from using finitely many samples per iteration. We now describe each in detail. First, because we solve the minimax problem in Equation (1) with ϵ-accuracy and not exactly, Algorithm 1 is better represented by the *approximate* minimax Bellman operator $\tilde{\mathcal{T}} : M(\mathcal{S}\times\mathcal{U}\times\mathcal{V}) \to M(\mathcal{S}\times\mathcal{U}\times\mathcal{V})$ defined by

$$\tilde{\mathcal{T}}Q(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}) = r(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}) + \gamma \mathbb{E}_{\boldsymbol{s}'\sim P(\cdot|\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})}\left[Q\left(\boldsymbol{s}', \mathcal{O}_\epsilon\big(Q\left(\boldsymbol{s}', \cdot, \cdot\right)\big)\right)\right], \tag{5}$$

where use a slight abuse of notation by denoting $Q(\boldsymbol{s}', \tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}})$ where $(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) = \mathcal{O}_\epsilon\big(Q(\boldsymbol{s}', \cdot, \cdot)\big)$ as $Q\big(\boldsymbol{s}', \mathcal{O}_\epsilon(Q(\boldsymbol{s}', \cdot, \cdot))\big)$. Secondly, without additional assumptions on the Markov game, the approximate Bellman operator need not satisfy *Bellman completeness*, i.e., it is possible that for $Q \in \mathcal{F}$, we have $\tilde{\mathcal{T}}Q \notin \mathcal{F}$. This motivates the fundamental approximation error

$$\delta_{\text{approx}} := \sup_{Q\in\mathcal{F}} \inf_{f\in\mathcal{F}} \left\| f - \tilde{\mathcal{T}}Q \right\|_{2,\sigma}. \tag{6}$$

The third error is statistical: because we only have n samples at each iteration, we can only perform an empirical approximate Bellman iteration. We will show how these affect the bound on $\|Q_K - Q^*\|_{2,\rho}$ for a distribution $\rho \in \mathcal{P}(\mathcal{S}\times\mathcal{U}\times\mathcal{V})$, where Q^* is the fixed point of the *exact* Bellman operator $\mathcal{T}$ given in Equation (1). We now list our assumptions that hold throughout our analysis.

Assumption 1 (State-Action Spaces)**.** *The continuous state and action spaces $\mathcal{S}$,$\mathcal{U}$ and $\mathcal{V}$ are compact and convex, with $D_\mathcal{S} := \sup_{\boldsymbol{s}\in\mathcal{S}} \|\boldsymbol{s}\|_2 < \infty$, $D_\mathcal{U} := \sup_{\boldsymbol{u}\in\mathcal{U}} \|\boldsymbol{u}\|_2 < \infty$ and $D_\mathcal{V} := \sup_{\boldsymbol{v}\in\mathcal{V}} \|\boldsymbol{v}\|_2 < \infty$.*

Assumption 2 (Reward Function)**.** *The immediate reward function r is continuous and $\|r\|_\infty \le R_{\max} < \infty$. Further, r is L_r-Lipschitz in its actions, i.e., there exists $L_r > 0$ such that for all $\boldsymbol{s} \in \mathcal{S}$, $\boldsymbol{u}, \boldsymbol{u}' \in \mathcal{U}$, and $\boldsymbol{v}, \boldsymbol{v}' \in \mathcal{V}$,*

$$|r(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}) - r(\boldsymbol{s}, \boldsymbol{u}', \boldsymbol{v}')| \le L_r\left(\|\boldsymbol{u} - \boldsymbol{u}'\|_2 + \|\boldsymbol{v} - \boldsymbol{v}'\|_2\right).$$

Assumption 3 (Transition Dynamics)**.** *For every bounded continuous function $h : \mathcal{S} \to \mathbb{R}$, the map $(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}) \mapsto \mathbb{E}_{s'\sim P(\cdot|\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})}[h(\boldsymbol{s}')]$ is continuous. Further, P is also L_P-Lipschitz in its actions, i.e.,, there exists $L_P > 0$ such that for all $\boldsymbol{s} \in \mathcal{S}$, $\boldsymbol{u}, \boldsymbol{u}' \in \mathcal{U}$, $\boldsymbol{v}, \boldsymbol{v}' \in \mathcal{V}$, and measurable sets S of $\mathcal{S}$,*

$$P(S|\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}) - P(S|\boldsymbol{s}, \boldsymbol{u}', \boldsymbol{v}') \le L_P\left(\|\boldsymbol{u} - \boldsymbol{u}'\|_2 + \|\boldsymbol{v} - \boldsymbol{v}'\|_2\right).$$

Assumptions 1 to 3 are standard in the literature and ensure that the Bellman operator $\mathcal{T}$ preserves continuity, admits a fixed point, and is Lipschitz continuous.

Assumption 4 (Exploratory Distribution). *There exists $c > 0$ such that exploratory sampling distribution σ has a conditional density $p(\boldsymbol{u}, \boldsymbol{v}|\boldsymbol{s})$ satisfying $p(\boldsymbol{u}, \boldsymbol{v}|\boldsymbol{s}) \geq c > 0$ for all $\boldsymbol{s} \in \mathcal{S}$, $\boldsymbol{u} \in \mathcal{U}$, $\boldsymbol{v} \in \mathcal{V}$.*

Assumption 5 (Action Spaces Regularity). *Consider a ball $\mathcal{B}(\boldsymbol{a}, \eta)$ in the joint action space $\mathcal{U} \times \mathcal{V} \subset \mathbb{R}^{d_{\boldsymbol{u}}+d_{\boldsymbol{v}}}$ centered at $\boldsymbol{a} \in \mathcal{U} \times \mathcal{V}$ with radius $\eta > 0$. We assume there exists $\alpha > 0$ such that*

$$\lambda\big(\mathcal{B}(\boldsymbol{a}, \eta) \cap (\mathcal{U} \times \mathcal{V})\big) \geq \min\left(\alpha \cdot \alpha_{d_{\boldsymbol{u}}+d_{\boldsymbol{v}}} \eta^{d_{\boldsymbol{u}}+d_{\boldsymbol{v}}}, \lambda\left(\mathcal{U} \times \mathcal{V}\right)\right).$$

Assumption 4 ensures that the action sampling distribution assigns positive mass to all regions of the players' action spaces, and is satisfied by standard distributions used in reinforcement learning such as Gaussian and uniform distributions. Assumption 5 is a mild geometric condition on the players' joint action spaces ruling out pathological geometries near the boundary, and is satisfied by convex action spaces.

Assumption 6 (Convex-Concave Function Approximation). *For the class of function approximators $\mathcal{F}$, every $Q \in \mathcal{F}$ satisfies:*

- *Boundedness: $\|Q\|_\infty \leq Q_{\max}$, with $Q_{\max} \geq {}^{R_{\max}}/_{1-\gamma}$.*
- *Convex-concavity: $Q(\boldsymbol{s}, \cdot, \cdot)$ is convex in $\boldsymbol{u}$ for fixed $\boldsymbol{v}$ and concave in $\boldsymbol{v}$ for fixed $\boldsymbol{u}$, for every $\boldsymbol{s}$.*
- *Joint Lipschitzness: There exists a positive constant $L_{\mathcal{F}} < \infty$ such that for all $Q \in \mathcal{F}$, $|Q(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}) - Q(\boldsymbol{s}, \boldsymbol{u}', \boldsymbol{v}')| \leq L_{\mathcal{F}}\big(\|\boldsymbol{u} - \boldsymbol{u}'\|_2 + \|\boldsymbol{v} - \boldsymbol{v}'\|_2\big)$ $\forall$ $\boldsymbol{s}$ and all $\boldsymbol{u}, \boldsymbol{u}' \in \mathcal{U}$, $\boldsymbol{v}, \boldsymbol{v}' \in \mathcal{V}$.*

Convexity-concavity in Assumption 6 is satisfied by the neural network architecture we introduce in Section 3 through construction. The Lipschitzness can be ensured by standard machine learning practices such as spectral normalization with bounded inputs (Miyato et al., 2018). Note that the same practices also ensure boundedness. We provide a Lipschitz bound for $\mathcal{F}$ in Theorem 2 of Section C.2.

Assumption 7 (Minimax Oracle). *We have access to an oracle $\mathcal{O}_\epsilon : \mathcal{S} \times \mathcal{F} \to \mathcal{U} \times \mathcal{V}$, such that for any desired $\epsilon > 0$, for $(\pi_Q, \mu_Q) := \mathcal{O}_\epsilon(\boldsymbol{s}, Q)$ we have*

$$\left|\max_{\boldsymbol{v} \in \mathcal{V}} Q\left(\boldsymbol{s}, \pi_Q, \boldsymbol{v}\right) - \min_{\boldsymbol{u} \in \mathcal{U}} Q\left(\boldsymbol{s}, \boldsymbol{u}, \mu_Q\right)\right| < \epsilon \;\forall\; \boldsymbol{s} \in \mathcal{S}, Q \in \mathcal{F}.$$

Assumption 7 is satisfied for convex-concave, Lipschitz Q-functions by a variety of minimax optimization methods introduced in prior literature (Nemirovski, 2004; Lin et al., 2020).

Assumption 8 (Discounted Concentrability Coefficients). *For the sampling distribution $\sigma \in \mathcal{P}(\mathcal{S} \times \mathcal{U} \times \mathcal{V})$ with state marginal $\sigma_{\mathcal{S}}$, let π^j denote any Markov policy with $\pi^j : \mathcal{S} \to \mathcal{U} \times \mathcal{V}$. Define the j^{th}-step concentrability coefficient for a distribution $\nu \ll \sigma_{\mathcal{S}}$*

$$c_j(\nu) := \sup_{\pi^1, \ldots, \pi^j} \left\| \frac{d(\nu P^{\pi^1} \cdots P^{\pi^j})}{d\sigma_{\mathcal{S}}} \right\|_\infty, \quad c_0(\nu) = \left\| \frac{d\nu}{d\sigma_{\mathcal{S}}} \right\|_\infty, \quad C_\nu := (1-\gamma) \sum_{j=0}^{\infty} \gamma^j c_j(\nu),$$

with $P^{\pi^j}(\cdot|\boldsymbol{s}) := P(\cdot|\boldsymbol{s}, \pi^j(\boldsymbol{s}))$. Then, we assume that $C_{\sigma_{\mathcal{S}}} < \infty$.

Assumption 8 quantifies the similarity between the sampling distribution σ and the future state distributions of the Markov game, and analogous versions of Assumption 8 are standard in reinforcement learning literature for both Markov decision processes and zero-sum Markov games (Antos et al., 2007; Munos & Szepesvári, 2008; Scherrer et al., 2015; Lazaric et al., 2016; Fan et al., 2020).

To analyze the convergence of Algorithm 1, we assume that Q^* from Section 2 admits a pure-strategy saddle point, so that Algorithm 1 converges towards a meaningful equilibrium. We formally state this as follows.

Assumption 9 (Saddle-point Existence). *For every $s \in \mathcal{S}$, $\min_{\boldsymbol{u} \in \mathcal{U}} \max_{\boldsymbol{v} \in \mathcal{V}} Q^*(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}) = \max_{\boldsymbol{v} \in \mathcal{V}} \min_{\boldsymbol{u} \in \mathcal{U}} Q^*(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v})$.*

We are now ready to state our main result. For the sampling distribution $\sigma \in \mathcal{P}(\mathcal{S} \times \mathcal{U} \times \mathcal{V})$, $\sigma_{\mathcal{S}}$ denotes the marginal of σ over $\mathcal{S}$.

Theorem 1. *Under Assumptions 1 to 9, let an evaluation distribution $\rho \in \mathcal{P}(\mathcal{S})$ satisfy $c_0(\rho) = \|{}^{d\rho}/_{d\sigma_{\mathcal{S}}}\|_\infty \leq C_{\rho,\sigma} < \infty$. Then for all $\delta > 0$, the output Q_K of Algorithm 1 after K iterations satisfies, with probability at least $1-\delta$*

$$\mathbb{E}_{\boldsymbol{s}\sim\rho}\left[\sup_{\boldsymbol{u}\in\mathcal{U},\boldsymbol{v}\in\mathcal{V}}|Q_K - Q^*|\right] \leq 2Q_{\max}C_{\rho,\sigma}\cdot\gamma^K + \frac{\sqrt{6}C_{\sigma_{\mathcal{S}}}C_{\rho,\sigma}\left(L_{\mathcal{F}} + L_r + 2\gamma Q_{\max}L_P\right)\eta_2}{1-\gamma}$$
$$+ \frac{2C_{\sigma_{\mathcal{S}}}C_{\rho,\sigma}}{(1-\gamma)\sqrt{cV\left(\eta_2/2\right)}}\left(\gamma\epsilon + \sqrt{\delta^2_{\text{approx}} + 5\Delta_n\left(\eta_1, \frac{\delta}{3K}\right) + 4Q^2_{\max}\sqrt{\frac{\log\left(3K/\delta\right)}{2n}}} + 6Q_{\max}\eta_1\right),$$

where

$$\Delta_n\left(\eta_1, {}^{\delta}/_{3K}\right) := 4Q^2_{\max}\sqrt{\frac{2\log\left({}^{6KN(\eta_1)}/_{\delta}\right)}{n}} + \frac{8Q^2_{\max}\log\left({}^{6KN(\eta_1)}/_{\delta}\right)}{3n} + 2Q_{\max}\eta_1,$$

$V(\eta_2/2) = \inf_{\boldsymbol{a}\in\mathcal{U}\times\mathcal{V}}\lambda\Big(\mathcal{B}(\boldsymbol{a},\eta_2/2)\cap(\mathcal{U}\times\mathcal{V})\Big) \geq \min\Big(\alpha\alpha_{d_{\boldsymbol{u}}+d_{\boldsymbol{v}}}\left(\eta_2/2\right)^{d_{\boldsymbol{u}}+d_{\boldsymbol{v}}}, \lambda\left(\mathcal{U}\times\mathcal{V}\right)\Big)$ *from Assumption 5, $N(\eta_1) := N(\mathcal{F},\eta_1,\|\cdot\|_\infty)$ is the covering number for the class of functions $\mathcal{F}$, and $\eta_1,\eta_2 > 0$ are chosen positive constants, and δ_{approx} is defined in Equation* (6)*.*

The proof of Theorem 1 is in Appendix A. Theorem 1 decomposes the error $\sup_{\boldsymbol{u}\in\mathcal{U},\boldsymbol{v}\in\mathcal{V}}\|Q_K - Q^*\|$ for some evaluation distribution ρ into several error sources. The first term is an algorithmic error that decays geometrically. The remaining terms capture the three error sources discussed above, and do not vanish with K, representing the fundamental difficulty of the problem. The ϵ term reflects the cost of using an ϵ-accurate solver as exact minimax solutions cannot be practically computed, and this term linearly vanishes with ϵ at the cost of more costly oracle computation. The presence of δ^2_{approx} in the bound represents the cost of working with a restricted, tractable convex-concave function class $\mathcal{F}$. δ_{approx} will be 0 when the underlying game satisfies Bellman completeness.

The terms $\mathcal{O}\big(\Delta_n(\eta_1, {}^{\delta}/_{3K})\big)$ and $\mathcal{O}\big(\sqrt{{}^{\log(3K/\delta)}/_{2n}}\big)$ correspond to the statistical error and quantify how fast the statistical error decays with the number of samples n and the covering number $N(\eta_1)$ of $\mathcal{F}$. Unlike the finite-action setting, continuous action spaces cannot be covered exactly with finitely many samples. The parameters $\eta_1,\eta_2 > 0$ correspond to this covering tradeoff. Smaller η_1,η_2 lead to better approximation but higher covering number $N(\eta_1)$ and larger inverse volume ${}^1/\sqrt{V(\eta_2/2)}$, which as mentioned in Theorem 1, depends exponentially on the dimension of the action spaces. The covering number $N(\eta_1)$ depends on the particular architecture used to construct $Q \in \mathcal{F}$ and depends polynomially on the action space dimensions. We give a bound on $N(\eta_1)$ for a particular architecture in Theorem 3 in Section C.3.

To the best of our knowledge, Theorem 1 is the first finite-sample guarantee of this form for zero-sum Markov games with continuous state and action spaces beyond the linear quadratic setting. Theorem 1 also allows us to establish a bound on the Nash gap when players employ their security strategies (π_{Q_K}, μ_{Q_K}) induced by the learned Q-function Q_K.

Corollary 1. *Let $\rho \in \mathcal{P}(\mathcal{S})$ be an evaluation distribution. Under Assumptions 1 to 9, if $C_\rho < \infty$, the oracle-returned approximate security strategies $(\pi_{Q_K}(\boldsymbol{s}), \mu_{Q_K}(\boldsymbol{s})) = \mathcal{O}_\epsilon\big(\boldsymbol{s}, Q_K\big)$ corresponding to the output Q_K of Algorithm 1 satisfy*

$$\texttt{NashGap}_\rho(\pi_{Q_K}, \mu_{Q_K}) \leq \frac{4C_\rho}{1-\gamma}\mathbb{E}_{\boldsymbol{s}\sim\sigma_{\mathcal{S}}}\left[\sup_{\boldsymbol{u}\in\mathcal{U},\boldsymbol{v}\in\mathcal{V}}|Q_K - Q^*|\right] + \frac{2\epsilon}{1-\gamma}.$$

The proof of Corollary 1 is in Section B. Importantly, Corollary 1 shows how the bounds developed in Theorem 1 can be used to bound the Nash gap itself.

5 Simulation Experiments

We now complement our result from Theorem 1 with simulation experiments investigating whether the convex-concave Q-function neural network architecture introduced in Equation (3) leads to stable training in practice. The practical issue arises because one cannot implement Algorithm 1 exactly, since each iteration requires solving a non-convex regression problem of optimizing a neural

network. We therefore empirically study a practical version of Algorithm 1 in which at each iteration, we take a gradient step on the empirical loss, mimicking online temporal difference (TD) learning. At iteration k of Algorithm 1, instead of getting Q_{k+1} from Equation (4), we have for a step-size $\alpha > 0$

$$\theta_{k+1} = \theta_k - \alpha \nabla_\theta \hat{L}_n(Q_k; Q_k), \quad Q_{k+1} = Q_{\theta_{k+1}}.$$

We investigate whether this practical modification to Algorithm 1 trains stably with the convex-concave Q-function architecture of Equation (3).

Markov Game Setup. We consider a planar pursuit-evasion game between two players in a 8-dimensional state space $\mathcal{S} = [-2.5, 2.5]^8$. At a state $\boldsymbol{s} \in \mathcal{S}$, the first 4 components $(s_1, s_2), (s_3, s_4)$ denote player 1's planar position and velocity, and the last 4 components $(s_5, s_6), (s_7, s_8)$ denote player 2's planar position and velocity. Both players have 2-dimensional action spaces $\mathcal{U}, \mathcal{V} = [-1, 1]^2$, and actions denote the players' horizontal and vertical planar accelerations. Both players follow noisy double integrator dynamics $\boldsymbol{s}' = A\boldsymbol{s} + B^1\boldsymbol{u} + B^2\boldsymbol{v} + w_n\xi$, where $A = \text{diag}(A_p, A_p)$, $B^1 = [B_p; 0]$, $B^2 = [0; B_p]$, $\xi \sim \mathcal{N}(0, I)$,

$$A_p = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and } B_p = \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}.$$

Player 1 wishes to pursue player 2, who evades player 1, while both want the save their control effort and waste the other's control effort. We encode this behavior with the immediate reward function

$$r(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}) = -w \exp\left(-\frac{\left((s_1 - s_5)^2 + (s_2 - s_6)^2\right)}{2w_d^2}\right) + \frac{1}{2}\boldsymbol{u}^\top R^1 \boldsymbol{u} - \frac{1}{2}\boldsymbol{v}^\top R^2 \boldsymbol{v}.$$

Note that this game lies beyond the linear-quadratic setting studied in prior work (Al-Tamimi et al., 2007a;b; Zhang et al., 2019; 2021b; Bu et al., 2019), and the optimal Q-function cannot be found by using Riccati equations. We choose the sampling distribution $\sigma = \left(\text{Uni}\left([-2.5, 2.5]^8\right) \times \text{Uni}\left([-1, 1]^2\right) \times \text{Uni}\left([-1, 1]^2\right)\right)$, with independent marginals. The discount factor γ is 0.95. We set $w = 0.01, w_d = 0.1, w_n = 0.01, \Delta t = 0.1, R^1 = 0.01I$ and $R^2 = 0.02I$. We use SecOND (Gupta et al., 2024), which is a second-order constraint-aware solver, as our minimax oracle $\mathcal{O}_\epsilon$, as we found it to converge favorably in practice. Instead of setting an ϵ, we terminate the solver once successive candidate solution iterates differ by less than a tolerance of 10^{-8}. For convex-concave, Lipschitz Q-functions, this serves as a practical proxy for ϵ. We use the partially input convex-concave neural network as described in Section 3 for our choice of Q-function, with softplus activations. We empirically bound the value that the Q-function neural network outputs by clipping values to lie between $[r_{\min}/1-\gamma, r_{\max}/1-\gamma]$, and found that while in theory this breaks convexity-concavity, it did not harm performance in practice. Additional implementation details are provided in Section D.

Along the iterations k, Figure 1 plots the normalized mean absolute error

$$\delta_{\text{NMAE}} := \frac{1/n \sum_{i=1}^n |Q_k(\boldsymbol{s}_i, \boldsymbol{u}_i, \boldsymbol{v}_i) - y_i|}{(r_{\max} - r_{\min})/1-\gamma},$$

where y_i is the regression target formed at iteration k of Algorithm 1 for the i^{th} data sample pair $(\boldsymbol{s}_i, \boldsymbol{u}_i, \boldsymbol{v}_i), i \in [n]$, and the denominator $(r_{\max} - r_{\min})/1-\gamma$ is the largest absolute error attainable given the bounded Q-value range derived above.

Figure 1 shows that the TD-style modification of Algorithm 1 trains stably with the partially input convex-concave Q-function architecture. The error δ_{NMAE} decreases and plateaus within the first 100 iterations across all sample sizes we test. Across all 20 seeds, there is no divergence observed across the iterations. This empirically suggests that using the partially input convex-concave architecture from Equation (3) for the Q-function allows stable training. Though Theorem 1 does not directly apply to this practical version, we observe that both the error and its spread shrink as n increases from 100 to 500, which is consistent with the statistical benefit of larger per-iteration sample size n.

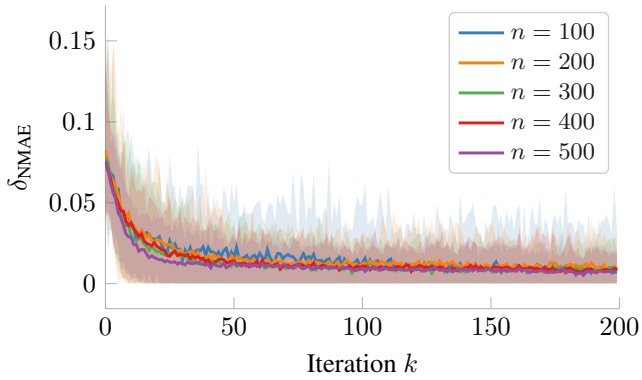


Figure 1: **The TD-style modification of Algorithm 1 trains stably with the partially input convex-concave Q-function architecture.** Each curve shows the normalized mean absolute training error δ_{NMAE} against iteration k, for n samples per iteration. Solid lines are the mean over 20 random seeds and shaded bands show the min–max spread across seeds. An increase in n yields both lower steady-state error and lower variance.

6 Conclusion

We study the learning of Q-functions in infinite-horizon zero-sum Markov games in continuous state and action spaces. Prior finite-sample guarantees of Q-function learning have largely been confined to either games with finite-actions or to continuous linear-quadratic games, where the agents have linear dynamics and quadratic rewards. A key challenge in extending the guarantees of these prior works to more general continuous-action zero-sum Markov games is that the associated Bellman equation involves a minimax problem that is not guaranteed to admit a tractable saddle-point solution in pure strategies. To address this issue, we introduce a class of neural network function approximation for Q which is convex-concave in the players' actions, guaranteeing that the minimax problem admits a pure-strategy saddle point that can be computed efficiently by existing solvers. We study an online variant of the fitted Q-iteration algorithm employing this function class and establish, to the best of our knowledge, the first finite-sample guarantees for continuous state-action zero-sum Markov games beyond the linear-quadratic setting. Finally, we empirically validate that these convex-concave neural networks lead to stable training when used on a practical version of modified fitted Q-iteration. The empirical results naturally point to a future work direction of extending the finite-sample analysis for the practical temporal difference-style variant of Theorem 1 studied in Section 5, in which the exact regression step of Algorithm 1 is replaced by a single gradient step, as in Section 5.

AI use statement

In this work, we used generative AI tools for the following critical tasks:

- Assist in a literature review about zero-sum continuous state-action Markov games.
- Help in understanding the proof machinery of related papers, to see how can existing mathematical techniques be used in our work. These techniques were critical for proving our mathematical claims.
- Help in implementing the practical version of Algorithm 1 for experiments in Section 5.

We have not used generative AI tools for other tasks with required disclosure mentioned in the AI policy for authors like proof writing and developing conceptual frameworks. The rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for checking grammatical correctness of our writing. We have reviewed all AI-assisted work. For the tasks listed above, we took the following measures:

- Any relevant paper suggested by an AI tool was manually screened to verify its relevance to our work.

- The applicability of proof techniques from prior work that were suggested by AI tools were manually verified, and the proof was then manually done using these suggested techniques.
- LLM-generated implementation was manually verified and tested.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

Our work deals with zero-sum game theory. Zero-sum games can be used for robust, safety-oriented applications, but the same techniques may be misused adversarially. Our work is mostly theoretical and does not directly enable any specific application that could be exploited in this manner.

Reproducibility statement

This work is mostly theoretical and all assumptions for the main Theorem 1 are listed before the theorem statement. The proofs of Theorem 1 and Corollary 1 are given in Section A and Section B respectively. Additional bounds and corresponding proofs are presented in Section C.2 and Section C.3. For our experiments, all details are presented in Section 5 and Section D. Code with running instructions can be found in the supplementary material.

References

Lukáš Adam, Rostislav Horčík, Tomáš Kasl, and Tomáš Kroupa. Double oracle algorithm for computing equilibria in continuous games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 5070–5077, 2021.

Asma Al-Tamimi, Murad Abu-Khalaf, and Frank L Lewis. Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(1):240–247, 2007a.

Asma Al-Tamimi, Frank L Lewis, and Murad Abu-Khalaf. Model-free q-learning designs for linear discrete-time zero-sum games with application to h-infinity control. *Automatica*, 43(3):473–481, 2007b.

Ahmet Alacaoglu, Luca Viano, Niao He, and Volkan Cevher. A natural actor-critic framework for zero-sum markov games. In *International Conference on Machine Learning*, pp. 307–366. PMLR, 2022.

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International conference on machine learning*, pp. 146–155. PMLR, 2017.

András Antos, Csaba Szepesvári, and Rémi Munos. Fitted q-iteration in continuous action-space mdps. *Advances in neural information processing systems*, 20, 2007.

Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory*. SIAM, 1998.

Jingjing Bu, Lillian J Ratliff, and Mehran Mesbahi. Global convergence of policy gradient for sequential zero-sum linear quadratic dynamic games. *arXiv preprint arXiv:1911.04672*, 2019.

Zaiwei Chen, Kaiqing Zhang, Eric Mazumdar, Asuman Ozdaglar, and Adam Wierman. Two-timescale q-learning with function approximation in zero-sum stochastic games. In *Proceedings of the 25th ACM Conference on Economics and Computation*, EC '24, pp. 378, 2024.

Jianqing Fan, Zhaoran Wang, Yuchen Xie, and Zhuoran Yang. A theoretical analysis of deep q-learning. In *Learning for dynamics and control*, pp. 486–489. PMLR, 2020.

Songtao Feng, Ming Yin, Yu-Xiang Wang, Jing Yang, and Yingbin Liang. Improving sample efficiency of model-free algorithms for zero-sum markov games. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=ZVmMV3AHjC.

Irving L Glicksberg. A further generalization of the kakutani fixed theorem, with application to nash equilibrium points. *Proceedings of the American Mathematical Society*, 3(1):170–174, 1952.

Kushagra Gupta, Xinjie Liu, Ross Allen, Ufuk Topcu, and David Fridovich-Keil. Second-order algorithms for finding local nash equilibria in zero-sum games. *arXiv preprint arXiv:2406.03565*, 2024.

Zeyu Jia, Lin F Yang, and Mengdi Wang. Feature-based q-learning for two-player stochastic games. *arXiv preprint arXiv:1906.00423*, 2019.

Michail G. Lagoudakis and Ronald Parr. Value function approximation in zero-sum markov games. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, UAI'02, pp. 283–292, 2002.

Alessandro Lazaric, Mohammad Ghavamzadeh, and Rémi Munos. Analysis of classification-based policy iteration algorithms. *Journal of Machine Learning Research*, 17(19):1–30, 2016.

Tianyi Lin, Chi Jin, and Michael I Jordan. Near-optimal algorithms for minimax optimization. In *Conference on learning theory*, pp. 2738–2779. PMLR, 2020.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Jun Morimoto and Kenji Doya. Robust reinforcement learning. In T. Leen, T. Dietterich, and V. Tresp (eds.), *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(5), 2008.

Arkadi Nemirovski. Prox-method with rate of convergence o(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

Julien Perolat, Bruno Scherrer, Bilal Piot, and Olivier Pietquin. Approximate dynamic programming for two-player zero-sum markov games. In *International Conference on Machine Learning*, pp. 1321–1329. PMLR, 2015.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International conference on machine learning*, pp. 2817–2826. PMLR, 2017.

Bruno Scherrer, Mohammad Ghavamzadeh, Victor Gabillon, Boris Lesner, and Matthieu Geist. Approximate modified policy iteration and its application to the game of tetris. *The Journal of Machine Learning Research*, 16(1):1629–1676, 2015.

Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10): 1095–1100, 1953.

Aaron Sidford, Mengdi Wang, Lin Yang, and Yinyu Ye. Solving discounted stochastic two-player games with near-optimal time and sample complexity. In *International Conference on Artificial Intelligence and Statistics*, pp. 2992–3002. PMLR, 2020.

Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171 – 176, 1958.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 2 edition, 2026.

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

Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. Policy optimization provably converges to nash equilibria in zero-sum linear quadratic games. *Advances in Neural Information Processing Systems*, 32, 2019.

Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Başar. Finite-sample analysis for decentralized batch multiagent reinforcement learning with networked agents. *IEEE Transactions on Automatic Control*, 66(12):5925–5940, 2021a.

Yufeng Zhang, Zhuoran Yang, and Zhaoran Wang. Provably efficient actor-critic for risk-sensitive and robust adversarial rl: A linear-quadratic case. In *International Conference on Artificial Intelligence and Statistics*, pp. 2764–2772. PMLR, 2021b.

Yulai Zhao, Yuandong Tian, Jason Lee, and Simon Du. Provably efficient policy optimization for two-player zero-sum markov games. In *International Conference on Artificial Intelligence and Statistics*, pp. 2736–2761. PMLR, 2022.

Kemin Zhou, John Comstock Doyle, Keith Glover, et al. *Robust and optimal control*, volume 40. Prentice hall New Jersey, 1996.

A Proof of Theorem 1

We will first list several classical results. Then, to prove Theorem 1, we establish a per-iteration error bound on $Q_{k+1} - \mathcal{T}Q_k$ in Appendix A.2, and in Appendix A.3 show how this bound propogates across iterations and leads to the final result of Theorem 1.

A.1 Useful Concentration Inequalities

We list two common concentration inequalities here that we use. A detailed background can be found in standard references, for example, Vershynin (2026).

Fact 1 (Bernstein)**.** *If $Z_1, \ldots, Z_n$ are i.i.d., $\mathbb{E}[Z_i] = 0$, $|Z_i| \le b$ a.s., $\mathrm{Var}(Z_i) \le \tau^2$, then for every $t > 0$,*

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} Z_i > t\right) \;\le\; \exp\left(-\frac{nt^2/2}{\tau^2 + bt/3}\right).$$

Equivalently, for every $\delta \in (0,1)$, with probability $\ge 1-\delta$, we have

$$\frac{1}{n}\sum_{i=1}^{n} Z_i \le \sqrt{\frac{2\tau^2 \log(1/\delta)}{n}} + \frac{2b\log(1/\delta)}{3n}.$$

Fact 2 (Hoeffding)**.** *If $Z_i, i \in [n]$ are bounded random variables such that $a_i \le Z_i \le b_i$ almost surely, then $\forall\, t > 0$*

$$P\left(\frac{1}{n}\sum_{i=1}^{n} Z_i - \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[Z_i\right] \right) \le \exp\left(-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

Equivalently, for $\delta > 0$, we have with probability $\ge 1-\delta$

$$\frac{1}{n}\sum_{i=1}^{n} Z_i \le \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[Z_i\right] + \sqrt{\frac{\sum_i (b_i - a_i)^2 \log\left(1/\delta\right)}{2n^2}}.$$

A.2 Per-iteration Error Bound

We will first bound the per-iteration error incurred by Algorithm 1. For convenience we restate the minimax Bellman operator $\mathcal{T}$ from Equation (1) and approximate minimax Bellman operator $\tilde{\mathcal{T}}$ from Equation (5):

$$\mathcal{T}Q\left(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}\right) = r\left(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}\right) + \gamma \mathbb{E}_{\boldsymbol{s}'\sim P(\cdot|\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})}\left[\min_{\tilde{\boldsymbol{u}}\in\mathcal{U}}\max_{\tilde{\boldsymbol{v}}\in\mathcal{V}} Q\left(\boldsymbol{s}',\tilde{\boldsymbol{u}},\tilde{\boldsymbol{v}}\right)\right],$$

$$\tilde{\mathcal{T}}Q(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}) = r(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}) + \gamma \mathbb{E}_{\boldsymbol{s}'\sim P(\cdot|\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})}\Big[Q\big(\boldsymbol{s}', \mathcal{O}_\epsilon(Q(\boldsymbol{s}',\cdot,\cdot))\big)\Big].$$

Let us define the error at iteration k as $\varepsilon_k := Q_{k+1} - \mathcal{T}Q_k$. As mentioned in Section 4, besides the fact that we can only approximate $\mathcal{T}Q_k$, there are additional sources of error: (i) the restriction of $Q_{k+1} \in \mathcal{F}$, while $\mathcal{T}Q_k, \tilde{\mathcal{T}}Q_k$ need not belong to $\mathcal{F}$, and (ii) a statistical error stemming from the use of finite samples. We decompose the error ε_k as

$$\varepsilon_k = \underbrace{\left(Q_{k+1} - \tilde{\mathcal{T}}Q_k\right)}_{\text{statistics + approximation}} + \underbrace{\left(\tilde{\mathcal{T}}Q_k - \mathcal{T}Q_k\right)}_{\text{oracle bias}}$$

$$\implies \|\varepsilon_k\|_{2,\sigma} \leq \left\|Q_{k+1} - \tilde{\mathcal{T}}Q_k\right\|_{2,\sigma} + \left\|\tilde{\mathcal{T}}Q_k - \mathcal{T}Q_k\right\|_{2,\sigma}.$$

From the use of $\mathcal{O}_\epsilon$ and Assumption 7, the oracle bias is uniformly bounded. It is not difficult to see that

$$\left\|\tilde{\mathcal{T}}Q_k - \mathcal{T}Q_k\right\|_{2,\sigma} \leq \gamma\epsilon. \tag{7}$$

We will now analyze the $\|Q_{k+1} - \tilde{\mathcal{T}}Q_k\|_{2,\sigma}$ term. Let $\boldsymbol{x} := (\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}) \in \mathcal{X} := (\mathcal{S} \times \mathcal{U} \times \mathcal{V})$. It is clear from Algorithm 1 that y_i is an unbiased conditional estimate of $\tilde{\mathcal{T}}Q_k(\boldsymbol{s}_i, \boldsymbol{u}_i, \boldsymbol{v}_i) = \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i)$, i.e., $y_i = \tilde{\mathcal{T}}Q_k + \xi_i$, where $\mathbb{E}\left[\xi \mid Q_k, \boldsymbol{x}_i\right] = 0$. We define

$$\bar{f}_k \in \arg\min_{f \in \mathcal{F}} \left\|f - \tilde{\mathcal{T}}Q_k\right\|_{2,\sigma}.$$

It will be useful to quantify how close Q_{k+1} is to $\tilde{\mathcal{T}}Q_k$ empirically for the batch of data samples compared to $\bar{f}_k$. We formalize this in the next lemma.

Lemma 1. *From Algorithm 1,* $Q_{k+1} = \arg\min_{f \in \mathcal{F}} \widehat{L}_n(f; Q_k)$ *yields*

$$\frac{1}{n}\sum_{i=1}^{n}\left(Q_{k+1}(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i)\right)^2 \leq \frac{1}{n}\sum_{i=1}^{n}\left(\bar{f}(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i)\right)^2 + \frac{2}{n}\sum_{i=1}^{n}\xi_i\left(Q_{k+1}(\boldsymbol{x}_i) - \bar{f}(\boldsymbol{x}_i)\right).$$

Proof. As Q_{k+1} is optimal for the regression problem at iteration k, we have

$$\widehat{L}_n(Q_{k+1}; Q_k) \leq \widehat{L}_n(\bar{f}_k; Q_k)$$

$$\implies \frac{1}{n}\sum_i \left(Q_{k+1}(\boldsymbol{x}_i) - y_i\right)^2 \leq \frac{1}{n}\sum_i (\bar{f}(\boldsymbol{x}_i) - y_i)^2$$

$$\implies \frac{1}{n}\sum_i \left(Q_{k+1}(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i) - \xi_i\right)^2 \leq \frac{1}{n}\sum_i (\bar{f}(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i) - \xi_i)^2$$

Opening squares, canceling the ξ_i terms and rearranging yields the claim. □

The $\|Q_{k+1} - \tilde{\mathcal{T}}Q_k\|_{2,\sigma}$ term can be bounded by uniformly bounding the two right hand side terms of Lemma 1. We will first bound the term $\frac{2}{n}\sum_{i=1}^{n}\xi_i\left(Q_{k+1}(\boldsymbol{x}_i) - \bar{f}(\boldsymbol{x}_i)\right)$.

For $f \in \mathcal{F}$, let $Z_i(f) := \xi_i\big(f(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i)\big)$. Note that we can apply Bernstein's inequality on Z_i. To see this, $\mathbb{E}\left[Z_i\right] = \mathbb{E}_{\boldsymbol{x}_i}\left[\mathbb{E}\left[Z_i|\boldsymbol{x}_i\right]\right] = 0$, and from Assumptions 2 and 6, we have $|y_i| \leq Q_{\max}$, $|\tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i)| \leq Q_{\max}$, which implies $|\xi_i| \leq 2Q_{\max}$. Thus,

$$|Z_i(f)| \leq 2Q_{\max} \cdot \|f - \tilde{\mathcal{T}}\|_\infty \leq 4Q_{\max}^2 =: b$$

Further, we have

$$\mathrm{Var}(Z_i(f)) \leq \mathbb{E}\left[\xi_i^2(f(x_i) - \tilde{\mathcal{T}}Q_k(x_i))^2\right] \leq 4Q_{\max}^2\|f - \tilde{\mathcal{T}}Q_k\|_{2,\sigma}^2 \leq 16Q_{\max}^4 =: \tau^2$$

Thus, for $f \in \mathcal{F}$, for $\delta' \in (0,1)$, we have with probability at least $1 - \delta'$

$$\frac{1}{n}\sum_i Z_i(f) \leq 4Q_{\max}^2\sqrt{\frac{2\log(1/\delta')}{n}} + \frac{8Q_{\max}^2\log(1/\delta')}{3n}$$

To extend this statement uniformly to the whole family $\mathcal{F}$, for a fixed $\eta_1 > 0$, let $N := N(\eta_1) := N(\mathcal{F}, \eta_1, \|\cdot\|_\infty)$ be the minimal covering number of $\mathcal{F}$ in the sup norm with associated cover $\{f^{(1)}, \ldots, f^{(N)}\}$. Note that $f \mapsto Z_i(f)$ is $2Q_{\max}$-Lipschitz in the sup norm. To see this, for $f, f' \in \mathcal{F}$, we have $|Z_i(f) - Z_i(f')| = |\xi \cdot (f(\boldsymbol{x}_i) - f'(\boldsymbol{x}_i))| \leq 2Q_{\max}\|f - f'\|_\infty$. Now, for each $f \in \mathcal{F}$, we have from the η_1-cover that $\exists\, j \in [N]$ such that $\|f - f^{(j)}\|_\infty \leq \eta_1$. Thus,

$$\sup_{f\in\mathcal{F}} \left|\frac{1}{n}\sum_i Z_i(f)\right| \leq \max_{1\leq j\leq N} \left|\frac{1}{n}\sum_i Z_i(f^{(j)})\right| + 2Q_{\max}\eta_1.$$

We can use the point wise bounds on the cover points $f^{(j)}$ with $\delta' = \delta/2N(\eta_1)$, where the factor of 2 comes because we bound the absolute value. Using the union bound yields with probability at least $1 - \delta$

$$\begin{aligned}\sup_{f\in\mathcal{F}} \left|\frac{1}{n}\sum_i Z_i(f)\right| &\leq 4Q_{\max}^2\sqrt{\frac{2\log(2N(\eta_1)/\delta)}{n}} + \frac{8Q_{\max}^2\log(2N(\eta_1)/\delta)}{3n} + 2Q_{\max}\eta_1 \\ &=: \Delta_n(\eta_1, \delta). \end{aligned} \tag{8}$$

Thus, we have

$$\frac{2}{n}\sum_i \xi_i\left(Q_{k+1}(\boldsymbol{x}_i) - \bar{f}_k(\boldsymbol{x}_i)\right) = \frac{2}{n}\sum_i Z_i(Q_{k+1}) - \frac{2}{n}\sum_i Z_i(\bar{f}_k) \leq 4\Delta_n(\eta_1, \delta) \tag{9}$$

We need to also connect the empirical LHS term of Lemma 1, $\frac{1}{n}\sum_{i=1}^n \left(Q_{k+1}(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i)\right)^2$ with the bound $\|Q_{k+1} - \tilde{\mathcal{T}}Q_k\|_{2,\sigma}^2$. For $f \in \mathcal{F}$, let us define a centered random variable

$$X_i := \left(f(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i)\right)^2 - \left\|f - \tilde{\mathcal{T}}Q_k\right\|_{2,\sigma}^2.$$

To analyze $\frac{1}{n}\sum_{i=1}^n \left(\bar{f}(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i)\right)^2$, we can again apply Bernstein's inequality on for X_i. To see this, we have $\mathbb{E}[X_i] = 0$, $|X_i| = \left|(f(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i))^2 - \left\|f - \tilde{\mathcal{T}}Q_k\right\|_{2,\sigma}^2\right| \leq 4Q_{\max}^2 = b$, and $\mathrm{Var}(X_i) = \mathrm{Var}\big((f(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i))^2\big) \leq \mathbb{E}\big[(f(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i))^4\big] \leq 16Q_{\max}^4 = \tau^2$. Also, the function $h(f) := (f(\boldsymbol{x}) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}))^2$ is $4Q_{\max}$-Lipschitz in the sup norm as for $f, f' \in \mathcal{F}$, $|h(f) - h(f')| = |f(\boldsymbol{x}) - f'(\boldsymbol{x})| \cdot |f(\boldsymbol{x}) + f'(\boldsymbol{x}) - 2\tilde{\mathcal{T}}Q_k(\boldsymbol{x})| \leq 4Q_{\max}\|f - f'\|_\infty$. Analyzing similarly to Z_i yields

$$\left|\frac{1}{n}\sum_i \left(f(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i)\right)^2 - \|f - \tilde{\mathcal{T}}Q_k\|_{2,\sigma}^2\right| \leq \Delta_n(\eta_1, \delta) + 6Q_{\max}\eta_1. \tag{10}$$

We now analyze the term $\frac{1}{n}\sum_{i=1}^n \left(\bar{f}(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i)\right)^2$ from Lemma 1. Note that for each i, $\left(\bar{f}(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i)\right)^2$ is i.i.d. with mean $\mathbb{E}_{\boldsymbol{x}\sim\sigma}\left[\left(\bar{f}(\boldsymbol{x}) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x})\right)^2\right] = \left\|\left(\bar{f} - \tilde{\mathcal{T}}Q_k\right)\right\|_{2,\sigma}^2 \leq \delta_{\text{approx}}^2$ from the definition of δ_{approx} from Equation (6). Also $\left(\bar{f}(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k(\boldsymbol{x}_i)\right)^2 \in \left[0, 4Q_{\max}^2\right]$. Thus, we can make use of Hoeffding's inequality. For $\delta > 0$, we have with probability at least $1 - \delta$ that

$$\begin{aligned}\frac{1}{n}\sum_i \left(\bar{f}_k(\boldsymbol{x}_i) - \tilde{\mathcal{T}}Q_k\right)^2 &\leq \left\|\left(\bar{f} - \tilde{\mathcal{T}}Q_k\right)\right\|_{2,\sigma}^2 + 4Q_{\max}^2\sqrt{\frac{\log(1/\delta)}{2n}} \\ &\leq \delta_{\text{approx}}^2 + 4Q_{\max}^2\sqrt{\frac{\log(1/\delta)}{2n}}. \end{aligned} \tag{11}$$

We will now combine these inequalities to get a per iteration error bound on ε_k.

Lemma 2 (Per iteration error bound)**.** *Under Assumptions 1 to 4, 6 and 7, for $\delta > 0$, for $\varepsilon_k = \|Q_{k+1} - \mathcal{T}Q_k\|$, we have with probability at least $\geq 1 - \delta$, the error at the k^{th} iteration*

$$\|\varepsilon_k\|_{2,\sigma} \leq \gamma\epsilon + \sqrt{\delta_{\text{approx}}^2 + 5\Delta_n(\eta_1, \frac{\delta}{3}) + 4Q_{\max}^2\sqrt{\frac{\log(3/\delta)}{2n}} + 6Q_{\max}\eta_1}\,.$$

Proof. We have

$$\begin{aligned}
\|Q_{k+1} - \tilde{\mathcal{T}} Q_k\|_{2,\sigma}^2 &\overset{Equation\ (10)}{\leq} \frac{1}{n}\sum_i \Big(Q_{k+1}(\boldsymbol{x}_i) - \tilde{\mathcal{T}} Q_k(\boldsymbol{x}_i)\Big)^2 + \Delta_n(\eta_1,\delta) + 6Q_{\max}\eta_1 \\
&\overset{Lemma\ 1}{\leq} \frac{1}{n}\sum_{i=1}^n \Big(\bar{f}(\boldsymbol{x}_i) - \tilde{\mathcal{T}} Q_k(\boldsymbol{x}_i)\Big)^2 + \frac{2}{n}\sum_{i=1}^n \xi_i\left(Q_{k+1}(\boldsymbol{x}_i) - \bar{f}(\boldsymbol{x}_i)\right) \\
&\qquad + \Delta_n(\eta_1,\delta) + 6Q_{\max}\eta_1 \\
&\overset{Equation\ (11)}{\leq} \frac{2}{n}\sum_{i=1}^n \xi_i\left(Q_{k+1}(\boldsymbol{x}_i) - \bar{f}(\boldsymbol{x}_i)\right) + \delta_{\text{approx}}^2 + 4Q_{\max}^2\sqrt{\frac{\log(1/\delta)}{2n}} \\
&\qquad + \Delta_n(\eta_1,\delta) + 6Q_{\max}\eta_1 \\
&\overset{Equation\ (9)}{\leq} \delta_{\text{approx}}^2 + 5\Delta_n(\eta_1,\delta) + 4Q_{\max}^2\sqrt{\frac{\log(1/\delta)}{2n}} + 6Q_{\max}\eta_1.
\end{aligned}$$

□

A.3 Error Propogation Across Iterations

We have so far a bound on $\|\varepsilon_k(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})\|_{2,\sigma}$. For error propogation across iterations, we shall need a bound on $\bar{\varepsilon}_k := \|\sup_{\boldsymbol{u},\boldsymbol{v}} \varepsilon_k(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})\|_{2,\sigma}$ To that end, we have the following result for our setting, analogous to single-agent versions used in prior work (Antos et al., 2007).

Lemma 3. *Let $f : \mathcal{S}\times\mathcal{U}\times\mathcal{V}\to\mathbb{R}$ be such that $\forall\ \boldsymbol{s}\in\mathcal{S}, f(\boldsymbol{s},\cdot,\cdot)$ is jointly L_f-Lipschitz and non-negative. Let $\mathcal{B}(\boldsymbol{a},\eta)$ denote an ℓ^2 ball in $\mathcal{U}\times\mathcal{V}$ centered at $\boldsymbol{a}\in\mathcal{U}\times\mathcal{V}$ with radius η. Then for $\eta_2>0$, we have*

$$\mathbb{E}_{\boldsymbol{s}\sim\sigma_{\mathcal{S}}}\left[\sup_{\boldsymbol{u},\boldsymbol{v}} f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})^2\right] \leq \frac{4}{cV(\eta_2/2)}\|f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})\|_{2,\sigma}^2 + 3L_f^2\eta_2^2,$$

where $V(\eta)$ is the lebesgue measure $V(\eta) = \inf_{\boldsymbol{a}\in\mathcal{U}\times\mathcal{V}} \lambda\Big(\mathcal{B}(\boldsymbol{a},\eta)\cap(\mathcal{U}\times\mathcal{V})\Big)$.

Proof. Let $\{(\boldsymbol{u}^j,\boldsymbol{v}^j)\}_{j\in[N(\eta_2)]}$ be the minimal cover of $\mathcal{U}\times\mathcal{V}$ associated with $\eta_2>0$ in the $\|\cdot\|_2$ norm. Consider $(\boldsymbol{u},\boldsymbol{v})$ and let its closest cover point be $(\boldsymbol{u}^j,\boldsymbol{v}^j)$. Then we have

$$\begin{aligned}
&\left|f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}) - f(\boldsymbol{s},\boldsymbol{u}^j,\boldsymbol{v}^j)\right| \leq L_f\eta_2 \\
\implies &|f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})| \leq \left|f(\boldsymbol{s},\boldsymbol{u}^j,\boldsymbol{v}^j)\right| + L_f\eta_2 \\
\implies &\sup_{\boldsymbol{u},\boldsymbol{v}} f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})^2 \leq 2\max_j f(\boldsymbol{s},\boldsymbol{u}^j,\boldsymbol{v}^j)^2 + 2L_f^2\eta_2^2 \qquad (12)
\end{aligned}$$

Now we bound $\max_j f(\boldsymbol{s},\boldsymbol{u}^j,\boldsymbol{v}^j)^2$. We have for $(\boldsymbol{u},\boldsymbol{v})$ in a radius $\eta_2/2$ around $(\boldsymbol{u}^j,\boldsymbol{v}^j)$, i.e., $(\boldsymbol{u},\boldsymbol{v})\in\mathcal{B}\left((\boldsymbol{u}^j,\boldsymbol{v}^j),\eta_2/2\right)$, that

$$\begin{aligned}
f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}) - f(\boldsymbol{s},\boldsymbol{u}^j,\boldsymbol{v}^j) &\geq -\frac{L_f\eta_2}{2} \\
\implies f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})^2 &\geq \left(\max\left\{f(\boldsymbol{s},\boldsymbol{u}^j,\boldsymbol{v}^j) - \frac{L_f\eta_2}{2}, 0\right\}\right)^2 \\
&\geq \frac{f(\boldsymbol{s},\boldsymbol{u}^j,\boldsymbol{v}^j)^2}{2} - \frac{L_f^2\eta_2^2}{4}.
\end{aligned}$$

Here we used that $(\max\{a-b,0\})^2 \geq a^2/2 - b^2$ for positive a,b. Now, $\|f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})\|_{2,p(\cdot,\cdot|s)}^2 \geq \int_{\mathcal{B}((\boldsymbol{u}^j,\boldsymbol{v}^j),\eta_2/2)} f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})^2 p(\boldsymbol{u},\boldsymbol{v}|\boldsymbol{s})d(\boldsymbol{u},\boldsymbol{v})$, meaning that $\exists\ c>0$ such that

$$\begin{aligned}
&\|f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})\|_{2,p(\cdot,\cdot|s)}^2 \\
&\geq \int_{\mathcal{B}((\boldsymbol{u}^j,\boldsymbol{v}^j),\eta_2/2)} f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})^2 p(\boldsymbol{u},\boldsymbol{v}|\boldsymbol{s})d(\boldsymbol{u},\boldsymbol{v}) \geq cV\left(\frac{\eta_2}{2}\right)\left(\frac{f(\boldsymbol{s},\boldsymbol{u}^j,\boldsymbol{v}^j)^2}{2} - \frac{L_f^2\eta_2^2}{4}\right) \qquad (13) \\
&\implies f(\boldsymbol{s},\boldsymbol{u}^j,\boldsymbol{v}^j)^2 \leq \frac{2}{cV(\eta_2/2)}\|f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})\|_{2,p(\cdot,\cdot|s)}^2 + \frac{L_f^2\eta_2^2}{2}. \qquad (14)
\end{aligned}$$

Combining Equations (12) and (13), we get that

$$\sup_{\boldsymbol{u},\boldsymbol{v}} f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})^2 \leq \frac{4}{cV\left(\eta_2/2\right)}\|f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})\|^2_{2,p(\cdot,\cdot|s)} + 3L_f^2\eta_2^2$$

$$\implies \mathbb{E}_{\boldsymbol{s}\sim\sigma_{\mathcal{S}}}\left[\sup_{\boldsymbol{u},\boldsymbol{v}} f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})^2\right] \leq \frac{4}{cV\left(\eta_2/2\right)}\|f(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})\|^2_{2,\sigma} + 3L_f^2\eta_2^2.$$

This completes the proof. □

We shall use Lemma 3 to analyze error propagation across iterations. We define $D_k := Q^* - Q_k$. Note that $D_{k+1} = Q^* - Q_{k+1} = \mathcal{T}Q^* - \mathcal{T}Q_{k+1} - \varepsilon_k$. Thus

$$D_{k+1}(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}) = \left(\mathcal{T}Q^* - \mathcal{T}Q_k\right)(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}) - \varepsilon_k(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}).$$

Let $e_k(\boldsymbol{s}) := \sup_{\boldsymbol{u},\boldsymbol{v}} |D_k(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})|$. Then, we have

$$\forall\ s \in \mathcal{S},\ e_{k+1}(\boldsymbol{s}) \leq \gamma \sup_{\boldsymbol{u},\boldsymbol{v}} \mathbb{E}_{\boldsymbol{s}'\sim P(\cdot|\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})}\left[e_k(\boldsymbol{s}')\right] + \sup_{\boldsymbol{u},\boldsymbol{v}} \left|\varepsilon_k(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})\right|.$$

At step K, for $\boldsymbol{s}_{(0)} \sim \sigma_{\mathcal{S}}$, we have

$$\begin{aligned}
e_K(\boldsymbol{s}_{(0)}) &\leq \gamma \sup_{\boldsymbol{u}_{(0)},\boldsymbol{v}_{(0)}} \mathbb{E}_{\boldsymbol{s}_{(1)}\sim P(\cdot|\boldsymbol{s},\boldsymbol{u}_{(0)},\boldsymbol{v}_{(0)})}\left[e_{K-1}(\boldsymbol{s}_{(1)})\right] + \sup_{\boldsymbol{u},\boldsymbol{v}} \left|\varepsilon_{K-1}(\boldsymbol{s}_{(0)},\boldsymbol{u},\boldsymbol{v})\right| \\
&\leq \gamma \sup_{\boldsymbol{u}_{(0)},\boldsymbol{v}_{(0)}} \mathbb{E}_{\boldsymbol{s}_{(1)}}\left[\gamma \sup_{\boldsymbol{u}_{(1)},\boldsymbol{v}_{(1)}} \mathbb{E}_{\boldsymbol{s}_{(2)}\sim P(\cdot|\boldsymbol{s}_{(1)},\boldsymbol{u}_{(1)},\boldsymbol{v}_{(1)})}\left[e_{K-2}(\boldsymbol{s}_{(2)})\right] + \sup_{\boldsymbol{u},\boldsymbol{v}} \left|\varepsilon_{K-2}(\boldsymbol{s}_{(1)},\boldsymbol{u},\boldsymbol{v})\right|\right] \\
&\qquad + \sup_{\boldsymbol{u},\boldsymbol{v}} \left|\varepsilon_{K-1}(\boldsymbol{s}_{(0)},\boldsymbol{u},\boldsymbol{v})\right| \\
&\leq \gamma^2 \sup_{\boldsymbol{u}_{(0)},\boldsymbol{v}_{(0)}} \mathbb{E}_{\boldsymbol{s}_{(1)}}\left[\sup_{\boldsymbol{u}_{(1)},\boldsymbol{v}_{(1)}} \mathbb{E}_{\boldsymbol{s}_{(2)}}\left[e_{K-2}(\boldsymbol{s}_{(2)})\right]\right] + \gamma \sup_{\boldsymbol{u}_{(0)},\boldsymbol{v}_{(0)}} \mathbb{E}_{\boldsymbol{s}_{(1)}}\left[\sup_{\boldsymbol{u},\boldsymbol{v}} \left|\varepsilon_{K-2}(\boldsymbol{s}_{(1)},\boldsymbol{u},\boldsymbol{v})\right|\right] \\
&\qquad + \sup_{\boldsymbol{u},\boldsymbol{v}} \left|\varepsilon_{K-1}(\boldsymbol{s}_{(0)},\boldsymbol{u},\boldsymbol{v})\right| \\
&\qquad\qquad \vdots \\
&\leq \gamma^K \sup_{\substack{\boldsymbol{u}_{(0)},\\ \boldsymbol{v}_{(0)}}} \mathbb{E}_{\boldsymbol{s}_{(1)}}\left[\sup_{\substack{\boldsymbol{u}_{(1)},\\ \boldsymbol{v}_{(1)}}} \mathbb{E}_{\boldsymbol{s}_{(2)}}\left[\cdots \sup_{\substack{\boldsymbol{u}_{(K-1)},\\ \boldsymbol{v}_{(K-1)}}} \mathbb{E}_{\boldsymbol{s}_{(K)}}\left[e_0(\boldsymbol{s}_{(K)})\right]\right]\right] \\
&\qquad + \sum_{k=0}^{K-2} \gamma^{K-1-k} \sup_{\substack{\boldsymbol{u}_{(0)},\\ \boldsymbol{v}_{(0)}}} \mathbb{E}_{\boldsymbol{s}_{(1)}}\left[\cdots \sup_{\substack{\boldsymbol{u}_{(K-2-k)},\\ \boldsymbol{v}_{(K-2-k)}}} \mathbb{E}_{\boldsymbol{s}_{(K-1-k)}}\left[\sup_{\boldsymbol{u},\boldsymbol{v}} \left|\varepsilon_k(\boldsymbol{s}_{(K-1-k)},\boldsymbol{u},\boldsymbol{v})\right|\right]\right] \\
&\qquad + \sup_{\boldsymbol{u},\boldsymbol{v}} \left|\varepsilon_{K-1}\left(\boldsymbol{s}_{(0)},\boldsymbol{u},\boldsymbol{v}\right)\right|
\end{aligned}$$

Consider the term e_0. We have $\forall\ \boldsymbol{s} \in \mathcal{S}, e_0(\boldsymbol{s}) \leq \|D_k\|_\infty \leq 2Q_{\max}$. For the second term, we have from Assumption 8 that

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{s}_{(0)}\sim\sigma_{\mathcal{S}}}\left[\sup_{\substack{\boldsymbol{u}_{(0)},\\ \boldsymbol{v}_{(0)}}} \mathbb{E}_{\boldsymbol{s}_{(1)}}\left[\cdots \sup_{\substack{\boldsymbol{u}_{(K-2-k)},\\ \boldsymbol{v}_{(K-2-k)}}} \mathbb{E}_{\boldsymbol{s}_{(K-1-k)}}\left[\sup_{\boldsymbol{u},\boldsymbol{v}} \left|\varepsilon_k(\boldsymbol{s}_{(K-1-k)},\boldsymbol{u},\boldsymbol{v})\right|\right]\right]\right] \\
&\leq c_{K-1-k}\mathbb{E}_{\boldsymbol{s}\sim\sigma_{\mathcal{S}}}\left[\sup_{\boldsymbol{u},\boldsymbol{v}} \left|\varepsilon_k(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})\right|\right] \\
&\leq c_{K-1-k}\sqrt{\mathbb{E}_{\boldsymbol{s}\sim\sigma_{\mathcal{S}}}\left[\sup_{\boldsymbol{u},\boldsymbol{v}} \varepsilon_k(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})^2\right]}, = c_{K-1-k}\bar{\varepsilon}_k \quad \text{(Jensen's inequality } (\mathbb{E}[X])^2 \leq \mathbb{E}[X^2]\text{)}
\end{aligned}$$

where we write $c_j := c_j(\sigma_{\mathcal{S}})$. Thus, we have

$$\mathbb{E}_{\boldsymbol{s}\sim\sigma}\left[e_K(\boldsymbol{s})\right] \leq \gamma^K \cdot 2Q_{\max} + \sum_{k=0}^{K-1} \gamma^{K-1-k} c_{K-1-k}\bar{\varepsilon}_k$$

Now we have for $\boldsymbol{s} \in \mathcal{S}$, $\boldsymbol{u}, \boldsymbol{u}' \in \mathcal{U}$ and $\boldsymbol{v}, \boldsymbol{v}' \in \mathcal{V}$

$$\begin{aligned}
|\varepsilon_k(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}) - \varepsilon_k(\boldsymbol{s},\boldsymbol{u}',\boldsymbol{v}')| &\leq |Q_k(\boldsymbol{s},\boldsymbol{u}',\boldsymbol{v}') - Q_k(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})| + |\mathcal{T}Q_k(\boldsymbol{s},\boldsymbol{u}',\boldsymbol{v}') - \mathcal{T}Q_k(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})| \\
&\leq L_{\mathcal{F}}\|(\boldsymbol{u},\boldsymbol{v}) - (\boldsymbol{u}',\boldsymbol{v}')\|_2 + |r(\boldsymbol{s},\boldsymbol{u}',\boldsymbol{v}') - r(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})| \\
&\quad + \gamma \left| \int \min_{\tilde{\boldsymbol{u}} \in \mathcal{U}} \max_{\tilde{\boldsymbol{v}} \in \mathcal{V}} Q_k(\boldsymbol{s}', \tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) \Big(P(d\boldsymbol{s}'|\boldsymbol{u},\boldsymbol{v}) - P(d\boldsymbol{s}'|\boldsymbol{u}',\boldsymbol{v}') \Big) \right| \\
&\leq \underbrace{\sqrt{2}\left(L_{\mathcal{F}} + L_r + 2\gamma Q_{\max} L_P\right)}_{:=L_\varepsilon} \|(\boldsymbol{u},\boldsymbol{v}) - (\boldsymbol{u}',\boldsymbol{v}')\|_2,
\end{aligned}$$

implying that ε_k is $L_\varepsilon := \sqrt{2}\left(L_{\mathcal{F}} + L_r + 2\gamma Q_{\max} L_P\right)$-Lipschitz in the joint actions. Thus from Lemma 3, we have for $\bar{\varepsilon}_k = \|\sup_{\boldsymbol{u},\boldsymbol{v}} |\varepsilon_k(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})|\|_{2,\sigma}$ for $\eta_2 > 0$

$$\begin{aligned}
\bar{\varepsilon}_k^2 &\leq \frac{4}{cV\left(\eta_2/2\right)} \|\varepsilon_k\|_{2,\sigma}^2 + 3L_\varepsilon^2 \eta_2^2 \\
\implies \bar{\varepsilon}_k &\leq \frac{2}{\sqrt{cV\left(\eta_2/2\right)}} \|\varepsilon_k\|_{2\sigma} + \sqrt{3} L_\varepsilon \eta_2.
\end{aligned}$$

Thus, for $\delta > 0$, from Lemma 2, using the union bound for k iterations, we have with probability at least $1 - \delta$

$\forall\, k \in [K], \bar{\varepsilon}_k \leq$

$$\underbrace{\sqrt{3} L_\varepsilon \eta_2 + \frac{2}{\sqrt{cV\left(\eta_2/2\right)}} \left(\gamma\epsilon + \sqrt{\delta_{\text{approx}}^2 + 5\Delta_n\left(\eta_1, \frac{\delta}{3K}\right) + 4Q_{\max}^2 \sqrt{\frac{\log\left(3K/\delta\right)}{2n}}} + 6Q_{\max}\eta_1 \right)}_{:=b}$$

This yields that with probability at least $1 - \delta$

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{s}\sim\sigma}\left[e_K(\boldsymbol{s})\right] &\leq \gamma^K \cdot 2Q_{\max} + \sum_{k=0}^{K-1} \gamma^{K-1-k} c_{K-1-k} \cdot b \\
&\leq \gamma^K \cdot 2Q_{\max} + \frac{C_{\sigma_{\mathcal{S}}}}{1-\gamma} b \\
\implies \mathbb{E}_{\boldsymbol{s}\sim\rho}\left[e_K(\boldsymbol{s})\right] &\leq \gamma^K \cdot 2Q_{\max} C_{\rho,\sigma} + C_{\sigma_{\mathcal{S}}} \frac{C_{\rho,\sigma} C_{\sigma_{\mathcal{S}}}}{1-\gamma} b, \qquad \left(d\rho/d\sigma \leq C_{\rho,\sigma}\right)
\end{aligned}$$

where in the last step we used $\mathbb{E}_\rho[g] \leq C_{\rho,\sigma}\mathbb{E}_\sigma[g]$ for non-negative g by the definition of $C_{\rho,\sigma}$ from the statement of Theorem 1. This completes the proof for Theorem 1.

B Proof of Corollary 1

Let us denote the true security strategies corresponding to $Q \in \mathcal{F}$ as $\pi_Q^*(\boldsymbol{s}) \in \arg\min_{\boldsymbol{u}\in\mathcal{U}} \max_{\boldsymbol{v}\in\mathcal{V}} Q(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})$ and $\mu_Q^*(\boldsymbol{s}) \in \arg\max_{\boldsymbol{v}\in\mathcal{V}} \min_{\boldsymbol{u}\in\mathcal{U}} Q(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})$. We will first bound the entity $\max_{\mu'} V^{\pi_{Q_K},\mu'} - V^*$. Consider any state $\boldsymbol{s}$ and Markov policy μ'. Let $e_K(\boldsymbol{s}) := \sup_{\boldsymbol{u}\in\mathcal{U},\boldsymbol{v}\in\mathcal{V}} |Q_K(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}) - Q^*(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v})|$. Now consider the difference $\tilde{\Delta}(\boldsymbol{s}) := V^{\pi_{Q_K},\mu'}(\boldsymbol{s}) - V^*(\boldsymbol{s}) = V^{\pi_{Q_K},\mu'}(\boldsymbol{s}) - Q^*(\boldsymbol{s},\pi_{Q_K},\mu'(\boldsymbol{s})) + Q^*(\boldsymbol{s},\pi_{Q_K},\mu'(\boldsymbol{s})) - V^*(\boldsymbol{s})$. From the definition of V and Q^* we have

$$\begin{aligned}
V^{\pi_{Q_K},\mu'}(\boldsymbol{s}) - Q^*(\boldsymbol{s},\pi_{Q_K},\mu'(\boldsymbol{s})) &= \gamma\mathbb{E}_{\boldsymbol{s}_1\sim P(\cdot|\boldsymbol{s},\pi_{Q_K}(\boldsymbol{s}),\mu'(\boldsymbol{s}))}\left[V^{\pi_{Q_K},\mu'}(\boldsymbol{s}_1) - V^*(\boldsymbol{s}_1)\right] \\
&= \gamma\mathbb{E}_{\boldsymbol{s}_1\sim P(\cdot|\boldsymbol{s},\pi_{Q_K}(\boldsymbol{s}),\mu'(\boldsymbol{s}))}\left[\tilde{\Delta}(\boldsymbol{s}_1)\right]
\end{aligned}$$

Thus we have the recursion

$$\tilde{\Delta}(\boldsymbol{s}) = Q^*(\boldsymbol{s},\pi_{Q_K},\mu'(\boldsymbol{s})) - V^*(\boldsymbol{s}) + \gamma\mathbb{E}_{\boldsymbol{s}_1\sim P(\cdot|\boldsymbol{s},\pi_{Q_K}(\boldsymbol{s}),\mu'(\boldsymbol{s}))}\left[\tilde{\Delta}(\boldsymbol{s}_1)\right].$$

This yields

$$V^{\pi_{Q_K},\mu'}(\boldsymbol{s}) - V^*(\boldsymbol{s}) = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\boldsymbol{s}_t|(\boldsymbol{s}_0=\boldsymbol{s},\pi_{Q_K},\mu')} \left[Q^*(\boldsymbol{s}_t, \pi_{Q_K}(\boldsymbol{s}_t), \mu'(\boldsymbol{s}_t)) - V^*(\boldsymbol{s}_t) \right], \tag{15}$$

where we use $\mathbb{E}_{\boldsymbol{s}_t|(\boldsymbol{s}_0=\boldsymbol{s},\pi_{Q_K},\mu')}$ as shorthand to denote the iterated expectation over the states from $\boldsymbol{s}_0$ to $\boldsymbol{s}_t$ under player policies π_{Q_K} and μ'. To bound $Q^*(\boldsymbol{s}, \pi_{Q_K}(\boldsymbol{s}), \mu'(\boldsymbol{s})) - V^*(\boldsymbol{s})$. We have that

$$\begin{aligned}
Q^*(\boldsymbol{s}, \pi_{Q_K}(\boldsymbol{s}), \mu'(\boldsymbol{s})) &\leq Q^*(\boldsymbol{s}, \pi_{Q_K}, \tilde{\boldsymbol{v}}),\ \tilde{\boldsymbol{v}} \in \arg\max_{\boldsymbol{v}\in\mathcal{V}} Q^*(\boldsymbol{s}, \pi_{Q_K}, \boldsymbol{v}) \\
&\leq Q_K(\boldsymbol{s}, \pi_{Q_K}, \tilde{\boldsymbol{v}}) + e_K(\boldsymbol{s}) \\
&\leq \max_{\boldsymbol{v}\in\mathcal{V}} Q_K(\boldsymbol{s}, \pi_{Q_K}, \boldsymbol{v}) + e_K(\boldsymbol{s}) \\
&\leq \min_{\boldsymbol{u}\in\mathcal{U}} Q_K(\boldsymbol{s}, \boldsymbol{u}, \mu_{Q_K}) + e_K(\boldsymbol{s}) + \epsilon \\
&\leq Q_K(\boldsymbol{s}, \tilde{\boldsymbol{u}}, \mu_{Q_K}) + e_K(\boldsymbol{s}) + \epsilon, \tilde{\boldsymbol{u}} \in \arg\min_{\boldsymbol{u}\in\mathcal{U}} Q^*(\boldsymbol{s}, \boldsymbol{u}, \mu_{Q_K}) \\
&\leq Q^*(\boldsymbol{s}, \tilde{\boldsymbol{u}}, \mu_{Q_K}) + 2e_K(\boldsymbol{s}) + \epsilon \\
&= \min_{\boldsymbol{u}\in\mathcal{U}} Q^*(\boldsymbol{s}, \boldsymbol{u}, \mu_{Q_K}) + 2e_K(\boldsymbol{s}) + \epsilon \\
&\leq \underbrace{\max_{\boldsymbol{v}\in\mathcal{V}} \min_{\boldsymbol{u}\in\mathcal{U}} Q^*(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v})}_{=V^*(\boldsymbol{s})} + 2e_K(\boldsymbol{s}) + \epsilon
\end{aligned}$$

Using this in Equation (15), we get

$$V^{\pi_{Q_K},\mu'}(\boldsymbol{s}) - V^*(\boldsymbol{s}) \leq \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\boldsymbol{s}_t|(\boldsymbol{s}_0=\boldsymbol{s},\pi_{Q_K},\mu')} \left[2e_K(\boldsymbol{s}_t) + \epsilon \right],$$

From Assumption 8, we have $\mathbb{E}_{\boldsymbol{s}\sim\rho}\left[e_K(\boldsymbol{s}_t)\right] \leq c_t(\rho)\mathbb{E}_{\boldsymbol{s}\sim\sigma_{\mathcal{S}}}\left[e_K(\boldsymbol{s}_t)\right]$. Thus

$$\begin{aligned}
\sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\boldsymbol{s}\sim\rho}\left[2e_K(\boldsymbol{s}_t) + \epsilon\right] &\leq \sum_{t=0}^{\infty} \gamma^t \left(2c_t(\rho)\mathbb{E}_{\boldsymbol{s}\sim\sigma_{\mathcal{S}}}\left[e_K(\boldsymbol{s})\right] + \epsilon\right) \\
&\leq \frac{2C_\rho}{1-\gamma}\mathbb{E}_{\boldsymbol{s}\sim\sigma_{\mathcal{S}}}\left[e_K(\boldsymbol{s})\right] + \frac{\epsilon}{1-\gamma}.
\end{aligned}$$

Thus we have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{s}\sim\rho}\left[V^{\pi_{Q_K},\mu'}(\boldsymbol{s}) - V^*(\boldsymbol{s})\right] &\leq \frac{2C_\rho}{1-\gamma}\mathbb{E}_{\boldsymbol{s}\sim\sigma_{\mathcal{S}}}\left[e_K(\boldsymbol{s})\right] + \frac{\epsilon}{1-\gamma} \\
\implies \mathbb{E}_{\boldsymbol{s}\sim\rho}\left[\max_{\mu'} V^{\pi_{Q_K},\mu'}(\boldsymbol{s}) - V^*(\boldsymbol{s})\right] &\leq \frac{2C_\rho}{1-\gamma}\mathbb{E}_{\boldsymbol{s}\sim\sigma_{\mathcal{S}}}\left[e_K(\boldsymbol{s})\right] + \frac{\epsilon}{1-\gamma}.
\end{aligned} \tag{16}$$

Similarly, for a policy π', let us define $\tilde{\Delta}'(\boldsymbol{s}) := V^*(\boldsymbol{s}) - V^{\pi',\mu_{Q_K}}(\boldsymbol{s}) = V^*(\boldsymbol{s}) - Q^*(\boldsymbol{s}, \pi', \mu_{Q_K}) + Q^*(\boldsymbol{s}, \pi', \mu_{Q_K}) - V^{\pi',\mu_{Q_K}}(\boldsymbol{s})$. From the definition of V^*, Q^*,

$$\begin{aligned}
Q^*(\boldsymbol{s}, \pi', \mu_{Q_K}) - V^{\pi',\mu_{Q_K}}(\boldsymbol{s}) &= \gamma\mathbb{E}_{\boldsymbol{s}_1\sim P(\cdot|\boldsymbol{s},\pi_{Q_K},\mu')}\left[V^*(\boldsymbol{s}_1) - V^{\pi',\mu_{Q_K}}(\boldsymbol{s}_1)\right] \\
&= \gamma\mathbb{E}_{\boldsymbol{s}_1\sim P(\cdot|\boldsymbol{s},\pi_{Q_K},\mu')}\left[\tilde{\Delta}'(\boldsymbol{s}_1)\right].
\end{aligned}$$

This gives the recursion

$$\tilde{\Delta}'(\boldsymbol{s}) = V^*(\boldsymbol{s}) - Q^*(\boldsymbol{s}, \pi', \mu_{Q_K}) + \gamma\mathbb{E}_{\boldsymbol{s}_1\sim P(\cdot|\boldsymbol{s},\pi_{Q_K},\mu')}\left[\tilde{\Delta}'(\boldsymbol{s}_1)\right],$$

which yields

$$V^*(\boldsymbol{s}) - V^{\pi',\mu_{Q_K}}(\boldsymbol{s}) = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\boldsymbol{s}_t|(\boldsymbol{s}_0=\boldsymbol{s},\pi_{Q_K},\mu')}\left[V^*(\boldsymbol{s}_t) - Q^*(\boldsymbol{s}_t, \pi'(\boldsymbol{s}_t), \mu_{Q_K}(\boldsymbol{s}_t))\right] \tag{17}$$

To bound $V^*(\boldsymbol{s}) - Q^*(\boldsymbol{s}, \pi'(\boldsymbol{s}), \mu_{Q_K}(\boldsymbol{s}))$, similar to before we have

$$\begin{aligned}
Q^*(\boldsymbol{s}, \pi'(\boldsymbol{s}), \mu_{Q_K}(\boldsymbol{s})) &\geq Q^*(\boldsymbol{s}, \tilde{\boldsymbol{u}}, \mu_{Q_K}(\boldsymbol{s})), \tilde{\boldsymbol{u}} \in \arg\min_{\boldsymbol{u}} Q^*(\boldsymbol{s}, \boldsymbol{u}, \mu_{Q_K}(\boldsymbol{s})) \\
&\geq Q_K(\boldsymbol{s}, \tilde{\boldsymbol{u}}, \mu_{Q_K}(\boldsymbol{s})) - e_K(\boldsymbol{s}) \\
&\geq \min_{\boldsymbol{u} \in \mathcal{U}} Q_K(\boldsymbol{s}, \boldsymbol{u}, \mu_{Q_K}(\boldsymbol{s})) - e_K(\boldsymbol{s}) \\
&\geq \max_{\boldsymbol{v} \in \mathcal{V}} Q_K(\boldsymbol{s}, \pi_{Q_K}, \boldsymbol{v}) - e_K(\boldsymbol{s}) - \epsilon \\
&\geq Q_K(\boldsymbol{s}, \pi_{Q_K}, \tilde{\boldsymbol{v}}) - e_K(\boldsymbol{s}) - \epsilon,\ \tilde{\boldsymbol{v}} \in \arg\max_{\boldsymbol{v} \in \mathcal{V}} Q^*(\boldsymbol{s}, \pi_{Q_K}(\boldsymbol{s}), \boldsymbol{v}) \\
&\geq Q^*(\boldsymbol{s}, \pi_{Q_K}, \tilde{\boldsymbol{v}}) - 2e_K(\boldsymbol{s}) - \epsilon \\
&= \max_{\boldsymbol{v} \in \mathcal{V}} Q^*(\boldsymbol{s}, \pi_{Q_K}, \boldsymbol{v}) - 2e_K(\boldsymbol{s}) - \epsilon \\
&\geq \underbrace{\min_{\boldsymbol{u} \in \mathcal{U}} \max_{\boldsymbol{v} \in \mathcal{V}} Q^*(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v})}_{:= V^*(\boldsymbol{s})} - 2e_K(\boldsymbol{s}) - \epsilon \\
\implies V^*(\boldsymbol{s}) &\leq Q^*(\boldsymbol{s}, \pi'(\boldsymbol{s}), \mu_{Q_K}(\boldsymbol{s})) + 2e_K(\boldsymbol{s}) + \epsilon.
\end{aligned}$$

Using in Equation (17), we get

$$V^*(\boldsymbol{s}) - V^{\pi', \mu_{Q_K}}(\boldsymbol{s}) \leq \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\boldsymbol{s}_t | (\boldsymbol{s}_0 = \boldsymbol{s}, \pi_{Q_K}, \mu')} \left[2e_K(\boldsymbol{s}_t) + \epsilon \right].$$

From similar analysis as previously, this yields for $\boldsymbol{s} \sim \rho$

$$\mathbb{E}_{\boldsymbol{s} \sim \rho} \left[V^*(\boldsymbol{s}) - \min_{\pi'} V^{\pi', \mu_{Q_K}}(\boldsymbol{s}) \right] \leq \frac{2C_\rho}{1-\gamma} \mathbb{E}_{\boldsymbol{s} \sim \sigma_{\mathcal{S}}} \left[e_K(\boldsymbol{s}) \right] + \frac{\epsilon}{1-\gamma} \tag{18}$$

Combining Equations (16) and (18), we get for $\boldsymbol{s} \sim \rho$

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{s} \sim \rho} \left[\max_{\mu'} V^{\pi_{Q_K}, \mu'}(\boldsymbol{s}) - \min_{\pi'} V^{\pi', \mu_{Q_K}}(\boldsymbol{s}) \right] &\leq \frac{4C_\rho}{1-\gamma} \mathbb{E}_{\boldsymbol{s} \sim \sigma_{\mathcal{S}}} \left[e_K(\boldsymbol{s}) \right] + \frac{2\epsilon}{1-\gamma} \\
\implies \texttt{NashGap}_\rho(\pi_{Q_K}, \mu_{Q_K}) &\leq \frac{4C_\rho}{1-\gamma} \mathbb{E}_{\boldsymbol{s} \sim \sigma_{\mathcal{S}}} \left[\sup_{\boldsymbol{u} \in \mathcal{U}, \boldsymbol{v} \in \mathcal{U}} |Q_K - Q^*| \right] + \frac{2\epsilon}{1-\gamma}
\end{aligned}$$

C On Partially Input Convex-Concave Neural Network Architecture

In this section, we describe the architecture underlying the function class $\mathcal{F}$ of Section 3. Throughout, we use the softplus action function $\varphi(x) := \log(1 + e^x)$ applied entrywise to vectors and matrices. Softplus satisfies the PICNN requirement of being convex and non-decreasing in the input required for guaranteeing desirable convexity in the partial input (Amos et al., 2017). We first review the PICNN architecture.

Partially Input Convex Neural Networks (Amos et al., 2017) Fix $i \in \{1, 2\}$ and write $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{s}, \boldsymbol{u})$ for $i = 1$ and $(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{s}, \boldsymbol{v})$ for $i = 2$. For convenience, we assume both blocks $N^i_{\theta^i}$, $i \in \{1, 2\}$ have hidden width m for the convex path corresponding to $\boldsymbol{y}$ and p for the state path corresponding to $\boldsymbol{x}$, and depth L. All results presented here follow for different, network specific choices of m, p as well. The architecture involves two recursions, one related with the *state* path, denoted by h^i, and the other related with the *convex* path, denoted by z^i, give by

$$\begin{aligned}
h^i_0 := \boldsymbol{x}, \qquad h^i_{\ell+1} &= \varphi\Big(\widetilde{W}^i_\ell \, h^i_\ell + \widetilde{b}^i_\ell \Big), \\
z^i_0 = \mathbf{0}, \quad z^i_{\ell+1} &= \varphi\Bigg(W^{(z),i}_\ell \Big(z^i_\ell \odot \varphi\big(W^{(zh),i}_\ell h^i_\ell + b^{(z),i}_\ell \big) \Big) + \\
&\qquad W^{(y),i}_\ell \Big(\boldsymbol{y} \odot \big(W^{(yh),i}_\ell \, h^i_\ell + b^{(y),i}_\ell \big) \Big) + W^{(h),i}_\ell \, h^i_\ell + b^i_\ell \Bigg), \\
N^i(\boldsymbol{x}, \boldsymbol{y}) &= (w^i)^\top z^i_L, \quad \text{where } \ell = 0, \ldots, L-1.
\end{aligned} \tag{19}$$

Convexity of $y \mapsto N^i_{\theta_i}(\boldsymbol{x}, \boldsymbol{y})$ requires $W^{(z),i}_\ell$ and w^i to have non-negative entries (Amos et al., 2017). This can be done in multiple ways, for example, by projecting the W^i_ℓ and w^i parameters to the non-negative set while training. In our work, to ensure smoothness, we enforce non-negativity through unconstrained reparameterization where one can optimize over free parameters $\tilde{W}^{(z),i}_\ell \in \mathbb{R}^{m\times m}$ and $\widetilde{w}^i \in \mathbb{R}^m$ and set

$$W^{(z),i}_\ell = \varphi\big(\tilde{W}^{(zh),i}_\ell\big), \qquad w^i = \varphi\big(\widetilde{w}^i\big) \tag{20}$$

Equation (19) implies that the network N^i has

$$\begin{aligned} |\theta^i| :=& L(m^2 + m d_{\boldsymbol{y}} + 2m + d_{\boldsymbol{y}}) + m + p d_{\boldsymbol{s}} + (p^2 + 2mp + p d_{\boldsymbol{y}})(L-1) + pL \\ &+ 2m d_{\boldsymbol{s}} + d_{\boldsymbol{y}} d_{\boldsymbol{s}} \end{aligned} \tag{21}$$

parameters, with $d_{\boldsymbol{y}} = d_{\boldsymbol{u}}$ for $i = 1$ and $d_{\boldsymbol{y}} = d_{\boldsymbol{v}}$ for $i = 2$. We use this partially input convex N^i to construct our Q-function, as described in Equation (3), which we restate here

$$Q_\theta(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}) = N^1_{\theta^1}(\boldsymbol{s}, \boldsymbol{u}) - N^2_{\theta^2}(\boldsymbol{s}, \boldsymbol{v}) + \boldsymbol{u}^\top M_{\theta^3}(\boldsymbol{s})\, \boldsymbol{v}, \qquad \theta = (\theta^1, \theta^2, \theta^3) \in \Theta. \tag{22}$$

Here, Θ is a convex set with $\|\theta\|_2 \le B_\Theta \; \forall\, \theta \in \Theta$. We take $M(\boldsymbol{s})$ as a standard feedforward network given by

$$h^M_0 = \boldsymbol{s}, \quad h^M_{\ell+1} = \varphi(W^M_\ell h^M_\ell + b^M_\ell),\ \ell = 0, \cdots, L_M - 1, \quad \mathrm{vec}(M) := W_{\text{out}} h^M_{L_M}, \tag{23}$$

where we take the hidden width as p_M. For convenience, we take the activation functions for M to be the same as N^i, though M can have any 1-Lipschitz activation alternatives such as ReLU. Our construction exploits two elementary properties of the softplus function φ. Equation (23) implies that M has

$$|\theta^3| = p_M L_M + (L_M - 1) p_M^2 + p_M d_{\boldsymbol{s}} + p_M d_{\boldsymbol{u}} d_{\boldsymbol{v}} \tag{24}$$

parameters. Thus, Q_θ has $|\theta^1| + |\theta^2| + |\theta^3|$ total parameters.

Lemma 4. *Let $\varphi(t) = \log(1 + e^t)$. Then*

(i) φ is 1-Lipschitz, strictly increasing and convex;

(ii) for every $\boldsymbol{a} \in \mathbb{R}^m$, with φ applied entrywise, $\|\varphi(\boldsymbol{a})\|_2 \le \|\boldsymbol{a}\|_2 + \sqrt{m}\log 2$.

Proof. (i) is immediate from $\varphi'(t) = e^t/(1 + e^t) \in (0, 1)$, and convexity follows from $\varphi'' > 0$. For (ii), let $\mathbf{1} \in \mathbb{R}^m$ be the all-ones vector and note $\varphi(0) = \log 2$. By (i) each coordinate obeys $|\varphi(a_j) - \log 2| \le |a_j|$, so $\|\varphi(\boldsymbol{a}) - (\log 2)\mathbf{1}\|_2^2 = \sum_j (\varphi(a_j) - \varphi(0))^2 \le \sum_j a_j^2 = \|\boldsymbol{a}\|_2^2$. The triangle inequality gives $\|\varphi(\boldsymbol{a})\|_2 \le \|\varphi(\boldsymbol{a}) - (\log 2)\mathbf{1}\|_2 + \|(\log 2)\mathbf{1}\|_2 \le \|\boldsymbol{a}\|_2 + \sqrt{m}\log 2$. □

The following assumption will be required to establish output and Lipschitz bounds, and is standard in neural network literature.

Assumption 10 (Bounded inputs and weights)**.** *There exist finite positive constants B_W, B_b, B_w, B_M and $D_{\mathcal{S}}$ such that:*

1. *For any weight matrix W and bias vector b in Equation* (19)*, the spectral norm $\|W\|_2 \le B_W$ and ℓ^2-norm $\|b\|_2 \le B_b$.*
2. *The output weights w^i satisfy $\|w^i\|_2 \le B_w$.*

Assumption 10 allows us to establish the Lipschitzness (with respect to the convex input) of a single PICNN N^i as described in Equation (19).

C.1 Notation for Lipschitz and covering number bounds.

Besides the constants in Assumption 10, we use several other notations throught the rest of this Section. For a scalar $t \in \mathbb{R}$, let the function $G(t, j)$ denote the sum $G(t, j) = \sum_{k=0}^{j} t^k$. From eq. (19), for N^i, we refer to the pre-activation layer of the convex path as $a^i_{\ell+1}$, i.e., $z^i_{\ell+1} = \varphi(a^i_{\ell+1})$, and we denote $D^i_{\ell+1} := \mathrm{diag}\left(\varphi'(a^i_{\ell+1})\right)$ for all $\ell \in \{0, \ldots, L-1\}$. Similarly, for

all $\ell \in \{0, \dots, L-1\}$, we refer to the pre-activation layer of the state path as $\tilde{a}^i_{\ell+1} := \tilde{W}^i_\ell h^i_\ell + \tilde{b}^i_\ell$, and define $\tilde{D}^i_{\ell+1} := \operatorname{diag}\left(\varphi'(\tilde{a}^i_{\ell+1})\right)$. For each layer ℓ of the PICNN N^i, we denote the vectors dependent only on $\boldsymbol{x}$ as

$$c^{(z)}_\ell := \varphi\big(W^{(zh),i}_\ell h^i_\ell + b^{(z),i}_\ell\big) \in \mathbb{R}^m_{\geq 0}, \qquad c^{(y)}_\ell := W^{(yh),i}_\ell h^i_\ell + b^{(y),i}_\ell \in \mathbb{R}^{d_{\boldsymbol{y}}},$$

with $\|c^{(z)}_\ell\|_\infty, \|c^{(y)}_\ell\|_\infty \leq B_W H + B_b + \log 2 =: C$, where H is defined in Lemma 5, and B_W, B_b defined in Assumption 10. We also denote $\rho := B_W(C)$, and $C_Z = B_W H + B_b + \sqrt{m}\log 2$. Finally, B_Θ is a positive constant such that $\|\theta\|_2 \leq B_\Theta \; \forall\, \theta \in \Theta$.

C.2 Lipschitz Bounds on Partially Input Convex-Conave Neural Network

Lemma 5 (Lipschitzness of PICNN)**.** *Under Assumptions 1 and 10, a single PICNN $N^i_\theta(\boldsymbol{x}, \boldsymbol{y})$ as described in Equation* (19) *is L_N-Lipschitz in the input $\boldsymbol{y}$, where*

$$L_N = B_w \cdot \rho \cdot G(\rho, L-1)$$

where $\rho = B_W(B_W H + B_b + \log 2)$ from Section C.1 and H is defined as

$$H = \max\left(D_{\mathcal{S}}, (B_W)^L D_{\mathcal{S}} + (B_b + \sqrt{p}\log 2)\cdot G(B_W, L)\right),$$

and $D_{\mathcal{S}} = \sup_{\boldsymbol{s}\in\mathcal{S}} \|\boldsymbol{s}\|_2 < \infty$ from Assumption 1.

Proof. We will first show that $\|h^i_\ell\|_2 \leq H$ for all $\ell \in [L]$ and $\boldsymbol{s} \in \mathcal{S}$. From Equation (19), we have $\|h^i_0\|_2 = \|\boldsymbol{s}\| \leq D_{\mathcal{S}}$, and

$$\|h^i_{\ell+1}\|_2 = \left\|\varphi\Big(\widetilde{W}^i_\ell\, h^i_\ell + \widetilde{b}^i_\ell\Big)\right\|_2 \leq B_W\|h^i_\ell\|_2 + B_b + \sqrt{p}\log 2.$$

Unrolling this yields H. For each layer ℓ, considering the vectors dependent only on $\boldsymbol{x}$, we have $\|c^{(z)}_\ell\|_\infty \leq \|W^{(zh),i}_\ell h^i_\ell + b^{(z),i}_\ell\|_2 \leq B_W H + B_b + \log(2)$. Similarly, $\|c^{(y)}_\ell\|_\infty \leq B_W H + B_b$. For fixed $\boldsymbol{x}$, the vectors $c^{(z)}_\ell$ and $c^{(y)}_\ell$ can be treated as constants independent of y. Then, the forward pass reduces to:

$$z^i_{\ell+1} = \varphi\Big(W^{(z),i}_\ell \operatorname{diag}\left(c^{(z)}_\ell\right) z^i_\ell + W^{(y),i}_\ell \operatorname{diag}\left(c^{(y)}_\ell\right)\boldsymbol{y} + \text{constant in } \boldsymbol{y}\Big).$$

Let $J_\ell := \|\partial z^i_\ell/\partial \boldsymbol{y}\|_2$. Differentiating through φ, since $\varphi' \in (0,1)$:

$$J_{\ell+1} \leq \|W^{(z),i}_\ell\|_2 \|c^{(z)}_\ell\|_\infty J_\ell + \|W^{(y),i}_\ell\|_2\|c^{(y)}_\ell\|_\infty, \qquad J_0 = 0.$$

Note that both $\|W^{(z),i}_\ell\|_2\|c^{(z)}_\ell\|_\infty$ and $\|W^{(y),i}_\ell\|_2\|c^{(y)}_\ell\|_\infty$ are less than $\rho := B_W(B_W H + B_b + \log 2)$. Then, unrolling $J_{\ell+1} \leq \rho J_\ell + \rho$ from $J_0 = 0$ yields $J_L \leq \sum_{k=1}^L \rho^k = \rho \cdot G(\rho, L-1)$. Since $N^i(\boldsymbol{x}, \boldsymbol{y}) = (w^i)^\top z^i_L$ and $\|\partial N^i/\partial y\|_2 \leq \|w^i\|_2 J_L \leq B_w J_L$, the mean value theorem gives $|N^i(\boldsymbol{x}, \boldsymbol{y}) - N^i(\boldsymbol{x}, \boldsymbol{y}')| \leq B_w J_L \|\boldsymbol{y} - \boldsymbol{y}'\| = L_N\|\boldsymbol{y} - \boldsymbol{y}'\|$. □

We are now ready to show that the convex-concave neural network function approximation class introduced in Section 3 is Lipschitz in the players' actions.

Theorem 2 (Lipschitzness of $Q \in \mathcal{F}$)**.** *Under Assumptions 1 and 10, any function belonging to the convex-concave neural network function class introduced in Section 3 with architecture specified in eqs.* (19) *and* (23) *is $L_{\mathcal{F}}$-Lipschitz in the players' joint actions, with*

$$L_{\mathcal{F}} = 2L_N + B_M\left(D_{\mathcal{U}} + D_{\mathcal{V}}\right),$$

where L_N is defined in Lemma 5, $D_{\mathcal{U}}, D_{\mathcal{V}}$ are defined in Assumption 1, and $\|M_\theta(s)\|_2 \leq B_M$ for all $s \in \mathcal{S}$. Note that a bound for B_M will be established later on in Theorem 3.

Proof. For fixed $\boldsymbol{s} \in \mathcal{S},\ \theta \in \Theta$, we have from Lemma 5 that

$$\begin{aligned}
\big|Q_\theta(\boldsymbol{s},\boldsymbol{u},\boldsymbol{v}) - Q_\theta(\boldsymbol{s},\boldsymbol{u}',\boldsymbol{v}')\big| &\leq \underbrace{|N^1(\boldsymbol{s},\boldsymbol{u}) - N^1(\boldsymbol{s},\boldsymbol{u}')|}_{\leq L_N\|\boldsymbol{u}-\boldsymbol{u}'\|_2} + \underbrace{|N^2(\boldsymbol{s},\boldsymbol{v}) - N^2(\boldsymbol{s},\boldsymbol{v}')|}_{\leq L_N\|\boldsymbol{v}-\boldsymbol{v}'\|_2} \\
&\quad + |\boldsymbol{u}^\top M(\boldsymbol{s})\boldsymbol{v} - \boldsymbol{u}'^\top M(\boldsymbol{s})\boldsymbol{v}'| \\
&\leq L_N\|\boldsymbol{u}-\boldsymbol{u}'\|_2 + L_N\|\boldsymbol{v}-\boldsymbol{v}'\|_2 + |\boldsymbol{u}^\top M(\boldsymbol{s})\boldsymbol{v} - \boldsymbol{u}^\top M(\boldsymbol{s})\boldsymbol{v}' + \boldsymbol{u}^\top M(\boldsymbol{s})\boldsymbol{v}' - \boldsymbol{u}'^\top M(\boldsymbol{s})\boldsymbol{v}'| \\
&\leq L_N\|\boldsymbol{u}-\boldsymbol{u}'\|_2 + L_N\|\boldsymbol{v}-\boldsymbol{v}'\|_2 + D_{\mathcal{V}}B_M\|\boldsymbol{u}-\boldsymbol{u}'\|_2 + D_{\mathcal{U}}B_M\|\boldsymbol{v}-\boldsymbol{v}'\|_2.
\end{aligned}$$

This established the Lipschitz constant $L_{\mathcal{F}}$ of $Q \in \mathcal{F}$ for the architecture specified in eqs. (19) and (23). □

C.3 Upper Bound on the covering number of convex-concave Q-function

Theorem 1 depends on the covering number of the convex-concave neural network function class $\mathcal{F}$ introduced in Section 3 in the sup norm, $N(\eta) := N(\mathcal{F}, \eta, \|\cdot\|_\infty)$ for the case when $Q_\theta \in \mathcal{F}$ is constructed according to eqs. (19) and (23). In this section, we provide a bound on this covering number.

Theorem 3 (Covering number bound)**.** *Under Assumptions 1 and 10, for convex-concave networks $Q_\theta \in \mathcal{F}$ constructed according to eqs.* (19) *and* (23) *with $\|\theta\|_2 \le B_\Theta \; \forall\, \theta \in \Theta$, we have*

$$N(\mathcal{F}, \eta, \|\cdot\|_\infty) \le \left(\frac{2B_\Theta \Gamma}{\eta} + 1\right)^{|\theta^1|+|\theta^2|+|\theta^3|}$$

with $\Gamma = \sqrt{2L_\theta^{\text{PICNN}} + L_\theta^M}$, where

$$L_\theta^{\text{PICNN}} = \left(C^2Z^2 + C^2D_{\boldsymbol{y}}^2 + H^2 + \left(B_W^2Z^2 + B_W^2D_{\boldsymbol{y}}^2\right)\left(H^2+1\right)+1\right)B_\delta + \left(H^2+1\right)B_{\tilde{\delta}} + Z^2,$$

$$L_\theta^M = D_{\mathcal{U}}^2 D_{\mathcal{V}}^2 \left(B_W\right)^{2L_M}\left(\left(H^M\right)^2+1\right) G\left(\frac{1}{B_W^2}, L_M - 1\right) + D_{\mathcal{U}}^2 D_{\mathcal{V}}^2 H^{M^2}$$

$$B_\delta = B_w^2 G\left(\rho^2, L-1\right),$$

$$B_{\tilde{\delta}} = \sum_{\ell=0}^{L-1}\left(\left(B_W^2(Z+D_{\boldsymbol{y}}) + B_W\right)B_w\rho^{L-\ell-1}G\left({}^{B_W}/_{\rho}, L-1-\ell\right)\right)^2$$

$$Z = \left(D_{\boldsymbol{y}}\rho + C_Z\right)G(\rho, L-1), \quad D_{\boldsymbol{y}} = \max D_{\mathcal{U}}, D_{\mathcal{V}},$$

C, C_Z, ρ are defined in Section C.1, H is the constant defined in Lemma 5, H^M is defined in eq. (28) *and $|\theta^1|, |\theta^2|$, and $|\theta^3|$ are defined in eqs.* (21) *and* (24)

Further, for $\sup_{\boldsymbol{s}} \|M(\boldsymbol{s})\|_2 \le B_M$ used in Theorem 2, we have $B_M = B_W H^M$.

Proof. To bound the covering number, we need a bound on the Lipschitz constant of $Q_\theta \in \mathcal{F}$ with respect to the network parameters θ, which can in turn be found by bounding $\sup_{\boldsymbol{su},\boldsymbol{v}} \|\nabla_\theta Q_\theta\|_2$ uniformly over all $\theta \in \Theta$. Let $\delta_\ell^i := {}^{\partial N^i}/_{\partial z_\ell^i}$, a bound on $\|\delta_\ell^i\|$ will help us bound the gradients with respect to all network parameters. By definition, $\delta_L^i = w^i$, and for $\ell = 0, \ldots, L-1$, by the chain rule

$$\delta_\ell^i = \left(\frac{\partial z_{\ell+1}^i}{\partial z_\ell^i}\right)^\top \delta_{\ell+1}^i = \text{diag}(c_\ell^{(z)})\left(W_\ell^{(z),i}\right)^\top \text{diag}(\varphi'(a_{\ell+1}^i))\delta_{\ell+1}^i$$

$$\implies \|\delta_\ell^i\|_2 \le \|c_\ell^{(z)}\|_\infty B_W \|\delta_{\ell+1}^i\|_2 = CB_W\|\delta_{\ell+1}^i\|_2$$

Starting from $\|\delta_L^i\|_2 = \|w^i\|_2 \le B_w$, this recursion implies $\|\delta_\ell^i\|_2 \le B_w\rho^{L-\ell}$. Let us define

$$\sum_{\ell=0}^{L-1} B_w^2\rho^{2(L-\ell-1)} = B_w^2 G\left(\rho^2, L-1\right) := B_\delta$$

We will also use a bound on $\|z_\ell^i\|_2$. We have $z_\ell^0 = \mathbf{0}$, and from Equation (19), $\|z_{\ell+1}^i\|_2 \le \rho\|z_\ell^i\|_2 + D_{\boldsymbol{y}}\rho + C_Z \implies \|z_\ell^i\|_2 \le \left(D_{\boldsymbol{y}}\rho + C_Z\right)G(\rho, \ell-1)$ implying that for all $\ell \in \{0, \ldots, L\}$, $\|z_\ell^i\|_2 \le \|z_L^i\|_2 \le \left(D_{\boldsymbol{y}}\rho + C_Z\right)G(\rho, L-1) =: Z$. We will use this result to bound the gradients with respect to all parameters. Starting with the parameters on the convex path, we have for $r_\ell^{(z),i} :=$

$W_\ell^{(zh),i} h_\ell^i + b_\ell^{(z),i}$:

$$
\begin{aligned}
&\nabla_{W_\ell^{(z),i}} N^i = D_{\ell+1}^i \delta_{\ell+1}^i (z_\ell^i \odot c_\ell^{(z)})^\top \implies \|\nabla_{W_\ell^{(z),i}} N^i\|_F \le \|\delta_{\ell+1}^i\|_2 \cdot CZ, \\
&\nabla_{W_\ell^{(y),i}} N^i = D_{\ell+1}^i \delta_{\ell+1}^i (\boldsymbol{y} \odot c_\ell^{(y)})^\top \implies \|\nabla_{W_\ell^{(y),i}} N^i\|_F \le \|\delta_{\ell+1}^i\|_2 \cdot CD_{\boldsymbol{y}}, \\
&\nabla_{W_\ell^{(h),i}} N^i = D_{\ell+1}^i \delta_{\ell+1}^i (h_\ell^i)^\top \implies \|\nabla_{W_\ell^{(h),i}} N^i\|_F \le \|\delta_{\ell+1}^i\|_2 \cdot H, \\
&\nabla_{W_\ell^{(zh),i}} N^i = \left(\varphi'\left(r_\ell^{(z),i}\right) \odot z_\ell^i \odot \left(\left(W_\ell^{(z),i}\right)^\top D_{\ell+1}^i \delta_{\ell+1}^i \right) \right) (h_\ell^i)^\top \\
&\qquad\qquad \implies \|\nabla_{W_\ell^{(zh),i}} N^i\|_F \le \|\delta_{\ell+1}^i\|_2 \cdot B_W Z H, \\
&\nabla_{W_\ell^{(yh),i}} N^i = \left(\left(\left(W_\ell^{(y),i}\right)^\top D_{\ell+1}^i \delta_{\ell+1}^i \right) \odot \boldsymbol{y} \right) (h_\ell^i)^\top \\
&\qquad\qquad \implies \|\nabla_{W_\ell^{(yh),i}} N^i\|_F \le \|\delta_{\ell+1}^i\|_2 \cdot B_W D_{\boldsymbol{y}} H \\
&\nabla_{b_\ell^{(z),i}} N^i = \varphi'\left(r_\ell^{(z),i}\right) \odot z_\ell^i \odot \left(\left(W_\ell^{(z),i}\right)^\top D_{\ell+1}^i \delta_{\ell+1}^i \right) \implies \|\nabla_{b_\ell^{(z),i}} N^i\|_2 \le \|\delta_{\ell+1}^i\|_2 \cdot B_W Z, \\
&\nabla_{b_\ell^{(y),i}} N^i = \left(\left(W_\ell^{(y),i}\right)^\top D_{\ell+1}^i \delta_{\ell+1}^i \right) \odot \boldsymbol{y} \implies \|\nabla_{b_\ell^{(y),i}} N^i\|_2 \le \|\delta_{\ell+1}^i\|_2 \cdot B_W D_{\boldsymbol{y}}, \text{ and} \\
&\nabla_{b_\ell^i} N^i = D_{\ell+1}^i \delta_{\ell+1}^i \implies \|\nabla_{b_\ell^i} N^i\|_2 \le \|\delta_{\ell+1}^i\|_2,
\end{aligned}
\tag{25}
$$

Similarly, for parameters on the state path, we will need a bound on $\tilde{\delta}_\ell^i := \nabla_{h_\ell^i} N^i$. We have $\tilde{\delta}_\ell^i = \left(\partial h_{\ell+1}^i / \partial h_\ell^i\right)^\top \tilde{\delta}_{\ell+1}^i + \left(\partial a_{\ell+1}^i / \partial h_\ell^i\right)^\top \left(\partial N^i / \partial a_{\ell+1}^i\right)$. From Section C.1, $\tilde{a}_{\ell+1}^i = \tilde{W}_\ell^i h_\ell^i + \tilde{b}_\ell^i$, and $\tilde{D}_{\ell+1}^i = \operatorname{diag}\left(\varphi'(\tilde{a}_{\ell+1}^i)\right)$. Then the first derivative above $\partial h_{\ell+1}^i / \partial h_\ell^i = \tilde{D}_{\ell+1}^i \tilde{W}_\ell^i$, and the second derivative $\partial a_{\ell+1}^i / \partial h_\ell^i = W_\ell^{(z),i} \operatorname{diag}\left(z_\ell^i \odot \varphi'\left(r_\ell^{(z),i}\right)\right) W_\ell^{(zh),i} + W_\ell^{(y),i} \operatorname{diag}(\boldsymbol{y}) W_\ell^{(yh),i} + W_\ell^{(h),i}$, yielding

$$
\begin{aligned}
\tilde{\delta}_\ell^i &= \left(W_\ell^{(z),i} \operatorname{diag}\left(z_\ell^i \odot \varphi'\left(r_\ell^{(z),i}\right)\right) W_\ell^{(zh),i} + W_\ell^{(y),i} \operatorname{diag}(\boldsymbol{y}) W_\ell^{(yh),i} + W_\ell^{(h),i} \right)^\top D_{\ell+1} \delta_{\ell+1}^i \\
&\quad + \left(\tilde{W}_\ell^i\right)^\top \tilde{D}_{\ell+1}^i \tilde{\delta}_{\ell+1}^i \\
&\implies \|\tilde{\delta}_\ell^i\|_2 \le B_W \|\tilde{\delta}_{\ell+1}^i\|_2 + \left(B_W^2 (Z + D_{\boldsymbol{y}}) + B_W\right) B_w \rho^{L-\ell-1}
\end{aligned}
$$

Unrolling this recursion with $\|\tilde{\delta}_{L-1}^i\|_2 \le \left(B_W^2 (Z + D_{\boldsymbol{y}}) + B_W\right) B_w$ yields $\|\tilde{\delta}_\ell^i\|_2 \le \left(B_W^2 (Z + D_{\boldsymbol{y}}) + B_W\right) B_w \rho^{L-\ell-1} G\left(B_W/\rho, L-1-\ell\right)$. We define

$$
\sum_{\ell=0}^{L-1} \left(\left(B_W^2 (Z + D_{\boldsymbol{y}}) + B_W\right) B_w \rho^{L-\ell-1} G\left(B_W/\rho, L-1-\ell\right) \right)^2 =: B_{\tilde{\delta}}
$$

Using this, we have

$$
\begin{aligned}
&\nabla_{\tilde{W}_\ell^i} N^i = (\tilde{D}_{\ell+1}^i \tilde{\delta}_{\ell+1}^i)(h_\ell^i)^\top \implies \|\nabla_{\tilde{W}_\ell^i} N^i\|_F \le \|\tilde{\delta}_{\ell+1}^i\|_2 \cdot H \\
&\nabla_{\tilde{b}_\ell^i} N^i = (\tilde{D}_{\ell+1}^i \tilde{\delta}_{\ell+1}^i) \implies \|\nabla_{\tilde{W}_\ell^i} N^i\|_F \le \|\tilde{\delta}_{\ell+1}^i\|_2
\end{aligned}
$$

Now for each PICNN N^i, we have

$$
\begin{aligned}
\|\nabla_{\theta^i} N^i\|_2^2 = \sum_{\ell=0}^{L-1} \Big(&\|\nabla_{W_\ell^{(z),i}} N^i\|_F^2 + \|\nabla_{W_\ell^{(y),i}} N^i\|_F^2 + \|\nabla_{W_\ell^{(h),i}} N^i\|_F^2 + \|\nabla_{W_\ell^{(zh),i}} N^i\|_F^2 \\
&+ \|\nabla_{W_\ell^{(yh),i}} N^i\|_F^2 + \|\nabla_{b_\ell^{(z),i}} N^i\|_2^2 + \|\nabla_{b_\ell^{(y),i}} N^i\|_2^2 + \|\nabla_{b_\ell^i} N^i\|_2^2 \\
&+ \|\nabla_{\tilde{W}_\ell^i} N^i\|_F^2 + \|\nabla_{\tilde{b}_\ell^i} N^i\|_2^2 + \|\nabla_{w^i} N^i\|_2^2 \Big).
\end{aligned}
\tag{26}
$$

Using the bounds for each set of derivatives from Equations (25) and (26), we have

$$\begin{aligned}\sup_{\theta^i}\|\nabla_{\theta^i}N^i\|_2^2 &\le \left(C^2Z^2+C^2D_{\boldsymbol{y}}^2+H^2+\left(B_W^2Z^2+B_W^2D_{\boldsymbol{y}}^2\right)\left(H^2+1\right)+1\right)\sum_{\ell=0}^{L-1}\|\delta_{\ell+1}^i\|_2^2\\ &\quad+\left(H^2+1\right)\sum_{\ell=0}^{L-1}\|\tilde{\delta}_{\ell+1}^i\|_2^2+Z^2\\ &\le \underbrace{\left(C^2Z^2+C^2D_{\boldsymbol{y}}^2+H^2+\left(B_W^2Z^2+B_W^2D_{\boldsymbol{y}}^2\right)\left(H^2+1\right)+1\right)B_\delta+\left(H^2+1\right)B_{\tilde{\delta}}+Z^2}_{=:L_\theta^{\text{PICNN}}} \quad (27)\end{aligned}$$

For the standard feedforward network $M_\theta^3(\boldsymbol{s})$ defined in eq. (23), we will first bound $\|h_\ell^M\|_2$. We have $h_{k+1}^M=\varphi(W_kh_k^M+b_k) \implies \|h_{\ell+1}^M\|_2\le B_W\|h_\ell^M\|_2+B_b+\sqrt{p_M}\log 2$. Unrolling this recursion with $\|h_0^M\|_2=\|\boldsymbol{s}\|_2\le D_{\boldsymbol{s}}$ yields $\|h_\ell^M\|_2\le D_{\mathcal{S}}B_W^\ell+(B_b+\sqrt{p_M}\log 2)\cdot G(B_W,\ell-1)$ for $\ell>0$. This gives that

$$\|h_\ell^M\|\le\max\left(D_{\mathcal{S}},D_{\mathcal{S}}B_W^{L_M}+(B_b+\sqrt{p_M}\log 2)\cdot G(B_W,L_M-1)\right)=:H^M \quad (28)$$

Note that Equation (28) also gives $\sup_{\boldsymbol{s}}\|M\|_2\le\|W_{\text{out}}h_{L_M}^M\|_F\le B_WH^M=:B_M$ Let $w=\text{vec}(\boldsymbol{u}\boldsymbol{v}^\top)$, then note that $\|w\|_2\le D_{\mathcal{U}}D_{\mathcal{V}}$. Let $f=\boldsymbol{u}^\top M\boldsymbol{v}$, then $f=\boldsymbol{u}^\top M\boldsymbol{v}=\left(\text{vec}(\boldsymbol{u}\boldsymbol{v}^\top)\right)^\top\text{vec}(M)=w^\top W_{\text{out}}h_{L_M}^M$. We have

$$\nabla_{W_{\text{out}}}f=w(h_{L_M}^M)^\top\implies\|\nabla_{W_{\text{out}}}f\|_F\le D_{\mathcal{U}}D_{\mathcal{V}}\cdot\|h_{L_M}^M\|$$

Let $\delta_\ell^M:=\nabla_{a_\ell^M}f$, then $\delta_\ell^M=\text{diag}\left(\varphi'(a_\ell^M)\right)(W_\ell^M)^\top\delta_{\ell+1}^M$. Unrolling this recursion with $\|\delta_{L_M}^M\|_2\le B_WD_{\mathcal{U}}D_{\mathcal{V}}$ yields $\|\delta_\ell^M\|_2\le B_W^{L_M-\ell+1}D_{\mathcal{U}}D_{\mathcal{V}}$. Now we have

$$\begin{aligned}\nabla_{W_\ell^M}f=\delta_{\ell+1}^M(h_\ell^M)^\top&\implies\|\nabla_{W_\ell^M}f\|_F\le B_W^{L_M-\ell}D_{\mathcal{U}}D_{\mathcal{V}}\|h_\ell^M\|_2\\ \nabla_{b_\ell^M}f=\delta_{\ell+1}^M&\implies\|\nabla_{W_\ell^M}f\|_F\le B_W^{L_M-\ell}D_{\mathcal{U}}D_{\mathcal{V}}\end{aligned}$$

Thus

$$\begin{aligned}&\|\nabla_{\theta^3}\boldsymbol{u}^\top M_{\theta^3}(\boldsymbol{s})\boldsymbol{v}\|_2^2=\|\nabla_{W_{\text{out}}}f\|_F^2+\sum_{\ell=0}^{L_M-1}\|\nabla_{W_\ell^M}f\|_F^2+\|\nabla_{b_\ell^M}f\|_2^2\\ \implies&\sup_{\theta^3}\|\nabla_{\theta^3}\boldsymbol{u}^\top M_{\theta^3}(\boldsymbol{s})\boldsymbol{v}\|_2^2\\ &\le\underbrace{D_{\mathcal{U}}^2D_{\mathcal{V}}^2\left(B_W\right)^{2L_M}\left(\left(H^M\right)^2+1\right)G\left(\frac{1}{B_W^2},L_M-1\right)+D_{\mathcal{U}}^2D_{\mathcal{V}}^2{H^M}^2}_{:=L_\theta^M}. \quad (29)\end{aligned}$$

Thus, using eqs. (27) and (29), any $Q_\theta\in\mathcal{F}$ constructed according to the architecture specified in eqs. (19) and (23) satisfies

$$\|\sup_\theta\nabla_\theta Q_\theta\|_2\le\sqrt{2L_\theta^{\text{PICNN}}+L_\theta^M}=\Gamma.$$

Thus, the Q_θ network in the function class $\mathcal{F}$ with the architecture specified in eqs. (19) and (23) is Γ-Lipschitz in the parameters θ. Correspondingly, using the facts that the required covering number $N(\mathcal{F},\eta,\|\cdot\|_\infty)\le N(\Theta,\eta/\Gamma,\|\cdot\|_2)$, and that for positive B_Θ such that $\|\theta\|_2\le B_\Theta\ \forall\ \theta\in\Theta$, $N(\Theta,\tilde{\eta},\|\cdot\|_2)\le(2B_\Theta/\tilde{\eta}+1)^{|\theta|}$ when Θ is any convex set within the B_Θ-bounded ℓ^2 ball in $R^{|\theta|}$ (c.f., Chapter 4 of Vershynin (2026)), we have $N(\mathcal{F},\eta,\|\cdot\|_\infty)\le(2B_\Theta\Gamma/\eta+1)^{|\theta^1|+|\theta^2|+|\theta^3|}$, with $|\theta^1|$, $|\theta^2|$, and $|\theta^3|$ from eqs. (21) and (24). □

D Experimental Details

In this section, we provide additional implementation details for the experiments in Section 5.

Convex-Concave Q-function. We used the architecture described in eqs. (19) and (23). Each partially convex input network has 2 hidden layers with hidden depth 32 for both state and convex paths. We used softplus activation for the convex path, and ReLU activation for the state path. For the bilinear network M, we used 1 hidden layer with width 32, and ReLU activation.

Hardware. All experiments were carried out on a desktop with an AMD Ryzen 9 7950X 16-Core Processor.